Understanding legacy information systems and abandonment decision making

Towards methodological support

Arnold Commandeur

Publisher: University of Groningen
Groningen, The Netherlands

Printed by: Ipskamp Printing
Enschede, The Netherlands

ISBN: 978-94-034-1587-1 (printed version)
978-94-034-1586-4 (electronic version)

rijksuniversiteit
groningen

Understanding legacy information systems and abandonment decision making

Towards methodological support

Proefschrift

ter verkrijging van de graad van doctor aan de
Rijksuniversiteit Groningen
op gezag van de
rector magnificus prof. dr. E. Sterken
en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op

donderdag 25 april 2019 om 12:45 uur

door

Arnold Lucas Commandeur

geboren op 14 mei 1965
te Wormerveer

Promotor

Prof. dr. E.W. Berghout

Beoordelingscommissie

Prof. dr. A. Boonstra
Prof. dr. J. Peppard
Prof. dr. P. Powell

ISBN: 978-94-034-1587-1 (printed version)
978-94-034-1586-4 (electronic version)

To Jacqueline and Anna Luna

ACKNOWLEDGEMENTS

"Sometimes it's a little better to travel than to arrive"
--- Pirsig, R. M. (1974)---

This PhD research was a journey of many years. For me, it was also a journey into the unknown. During this journey I met many interesting people, who opened their door, offered me their time, hospitality, knowledge and indicated many new doors to open. This dissertation is therefore a report of a humble traveler, who used scientific methods to report and organize the wisdom of many others. Thank you for sharing your knowledge with me.

I would like to thank Prof. dr. Egon Berghout for being my supervisor. He invited me for this journey and saw the opportunity for me to finish this quest. He guided me on my journey of this research and through the process of writing this dissertation. Without his help I would never have accomplished this quest successfully. I would also like to thank Prof. dr. Philip Powell as a honorary professor at the University of Groningen for his reviews and suggestions. Furthermore, I would like to thank Durkje van Lingen and Iris Neef-Huizinga for making my life on the university much easier and the University of Groningen for providing me the opportunity to work on my dissertation.

I would also like to thank dr. Johan Kerling. Before the millennium we were IBM colleagues and worked together in the large account sales team of the Personal Computing division of IBM. Fifteen years later Johan spent a lot of time on editing this dissertation. Furthermore, I would like to thank all my colleagues and staff at the University of Groningen, especially my roommate on Zernike in Groningen dr. Peter Schuurman and my colleague dr. Jan Braaksma. I would also like to thank my fellow PhD students and guest speakers at the most valuable course I took in 2010-2011, the Information Management PhD course, headed by prof. dr. Michel Avital.

My gratitude goes to Jacqueline Schlikker for being my partner and mother of our daughter Anna Luna. Without the support of Jacqueline and Anna Luna this research would not have been possible, they always supported me and never complained. To family (Cor, Roelina, Chris, Ingrid, Paul, Zeliha, Lara, Pascalle, David) and friends (Marco, Gerda, William, Michel, Peter) I would like to say thanks for supporting me. You often invited me to parties but I did not have time, and when I joined I had to leave early because there was always some work to do. Thanks for keeping inviting me to parties and holidays. I will definitely spend more time with you.

Arnold Commandeur

Wormer, The Netherlands
March 2019

TABLE OF CONTENTS

1 INTRODUCTION

Similar to other systems, information systems (IS) are subject to deterioration and obsolescence through aging (Lehman 1980a; Parnas, 1994; Swanson & Dans, 2000). These aging IS typically increasingly resist meeting organizational requirements and are, at some point in time, referred to as "legacy IS". Eventually an organization may decide to abandon these legacy IS. This research attempts to deepen the understanding of legacy IS and the discontinuation of IS. What does aging of IS imply? Why do organizations decide to abandon IS? What good practices regarding the management of IS, or decision support methodologies for abandonment decision making could be identified?

The literature and theory on how legacy IS are aging, how organizations decide to abandon legacy IS and how organizations practically abandon legacy IS remains scant (see Chapter 3) and this impedes deliberate decision making around legacy IS (Furneaux & Wade, 2011; Sakthivel, 1994; Sellars, 2004; Swanson & Dans, 2000). Furthermore, methods to support the abandoning of legacy IS, or good practices for abandoning, could hardly be found in the literature. The social relevance of abandoning legacy IS seems, however, high, as legacy IS consume large amounts of resources and the continuity of the IS, or even entire organizations, could be at stake. According to the United States Government Accountability Office, GAO (2015, p. 47), federal agencies plan to spend about $58 billion on legacy systems. This research focuses on the aging, abandonment decision making and practical abandoning of legacy IS. Besides literature research, five case studies have been researched. Furthermore, a method is proposed to support decision making regarding the abandoning of legacy IS.

IS typically go through a life cycle of inception, maintenance and abandoning (Berghout & Nijland, 2002; Brussaard & Tas, 1980). Concerning inception decision making, ample research is available (Andresen, 2001; Braaksma, Commandeur, & Berghout, 2006; Berghout, 1997; Remenyi, 1999; Renkema & Berghout, 1999). This also holds for decision making during the development of IS (Jacobson, Booch, & Rumbaugh, 1999; Martin, 1991; Royce, 1970; Sassenburg, 2005). Concerning the subsequent life cycle phase, being maintenance of IS (including re-engineering or modernization), there is also a considerable amount of literature available. For instance, the best practices of ITIL (OCG, 2007), ASL (Pols & Backer, 2006) and BISL (Pols & Backer, 2007). Regarding re-engineering or modernization of IS, methodologies have been developed by Brodie and Stonebraker (1995), Seacord, Plakosh and Lewis (2003), Sellars (2004), Ulrich (2002), Ulrich and Newcomb (2010), Van den Heuvel (2007) and Warren (1999). The final management activity, being the "abandon" activity of IS, remains relatively unexplored (Furneaux, 2009; Furneaux & Wade, 2010, 2011; Swanson & Dans, 2000).

Abandoning a legacy IS concerns the process of abandoning current constellations and combinations of software, hardware, data sets, people and procedures. Often individual components of the abandoned legacy IS can be re-used in a new IS, for instance, people and data sets. However, components carrying the data can also be malfunctioning, e.g. hardware and software (Grance, Hash, & Stevens, 2004). Many organizations continue to operate IS designed and developed during the 1960's and 1970's (Daga, Cesare, Lycett, & Partridge, 2005). These systems sometimes run on obsolete hardware and software and are often designed in stovepipe fashion, include rigid work processes and their maintenance budgets frequently consume 60-80% of the software related budgets of the organization (Brodie & Stonebraker, 1995). Large organizations nowadays have hundreds of operational IS. However, there are few indications of the actual number of legacy IS and their impact on society. It is estimated there are 200 billion lines of COBOL software, accounting for roughly 60% of the total software deployed worldwide (Ulrich, 2002) and more than 30-billion COBOL-based transactions are processed daily (Lawrence, 2007). Gartner estimates that legacy IS are 30-50% more expensive to run than comparable packaged systems or other newer technologies (Hunter & Aron, 2006).

Running legacy IS may cause various problems. For example, systems are more difficult to modify with the confidence that a given change will not cause other problems (Ulrich, 2002). Legacy IS often run on obsolete hardware that is slow and expensive to maintain (Bisbal, Lawles, Wu, & Crimson, 1999). For a business to be agile, IS should be adaptable and malleable and most legacy systems are not (Ulrich, 2002). "Legacy" according

to Lawrence (2007) refers to existing information technology (IT) assets that have been deployed in the past. These assets could have been installed anywhere from yesterday to twenty years ago and in many cases, the legacy IS is typically supporting critical business processes.

In the following Section 1.1 the research definitions, followed by the primary research question (see Section 1.2, p. 8) and finally the dissertation outline (see Section 1.3) are elaborated on and described.

1.1 Research definitions

Many definitions and interpretations of IS exist and the word or terminology "IS" is often used without explanation or definition (Orlikowski & Iacono, 2001). Therefore this section starts with an elaboration of the research definitions. First, based on literature, the concept of IS including their constituents are elaborated and the IS artifact of research is defined (Section 1.1.1). Then, the concept of a legacy IS including the legacy constituents is worked out (Section 1.1.2). Finally, the concept of decision making is discussed (Section 1.1.3).

1.1.1 Information systems

The etymology of the words "information" and "systems" provides us with anecdotes that are tangential to the concept of "information systems". Therefore, first the etymology of "information" is enlarged upon and then the etymology of "systems" is described.

The term information has a Latin origin (Capurro, 2009; Capurro & Hjorand, 2003). The Thesaurus Linguae Latinae (1900) refers to *informatio* and *informo* in Latin since Vergil (70-19 B.C.) (Capurro & Hjorand, 2003). *Informatio* has two fundamental meanings, an objective meaning namely the action of giving a form to something practical as well as a subjective meaning namely the act of communicating knowledge to another person (Capurro, 2009). From the middle ages to modernity (due to the loss in everyday language) there is a transition from the objective meaning of information: *"giving a (substantial) form to matter",* retaining the subjective one: *"communicating something (new) to someone"* (Capurro, 2009). Today, the Oxford Dictionary of English (2010) defines "information" as: "*facts provided or learned about something or someone*". The word "system" also has a long history which can be traced back to Plato (Lerner, 2005). It meant *"whole compounded of several parts or members"* (Liddell et al., 1940) and it was called *systema* in late Latin (System, n.d.). From the 17th century it means: *"an organized or connected group of objects"* (Liddell et al., 1940).

Systems are as pervasive as the universe in which we live (Blanchard & Fabrycky, 1990). At one extreme, they are as grand as the universe itself. At the other extreme, they are as infinitesimal as the atom (Blanchard & Fabrycky, 1990). Every system is made up of components and any component can be broken down into smaller components (Blanchard & Fabrycky, 1990). This characteristic of a system is referred to as recursivity, if two hierarchical levels are involved in a given system, the lower hierarchy is referred to as sub-system, and a higher hierarchy is called a supra-system (Blanchard & Fabrycky, 1990). In any particular situation it is important to define the system under consideration by specifying its limits or boundaries. Everything that remains outside the boundaries of the system is considered to be the environment (Blanchard & Fabrycky, 1990). Sommerville (2007) defines a system as a purposeful collection of interrelated components that work together to achieve some objective.

"Information system" as a combination of words was first documented in "The Times" in 1904 and in relation to computing environment in "Moore school lectures" in 1946 (Information, n.d.). IS come in different forms, computerized or not, from a manual card box to a highly sophisticated computerized system. An IS is defined by Chaffey and Wood (2005) as a computerized system or manual system to capture data and transform them into information or knowledge. "Data" is a representation of facts whilst "information" is the data processed in an order that is meaningful to an interpreter. Davis and Olson (1985) define an IS as an integrated, user-machine system for providing information to support operations, management and decision making functions in an organization. The system utilizes computer hardware and software, manual procedures, models for analysis,

planning control and decision making and a database (Davis & Olson, 1985). In defining an IS, Avgerou and Cornford (1998) use a more social and organizational focus. They refer to information and data handling activities in human organizations. “Information handling” in this sense is a purposeful activity sustained over time and includes the activities of collecting information, storing it, directing it to appropriate places and utilizing it in various tasks within the organization (Avgerou & Cornford, 1998). IS are social systems, heavily influenced by goals, values and beliefs of individuals and groups, as well as the performance of the technology (Angell & Smithson, 1991). As such the behavior of IS is not deterministic and does not fit into any formal algorithmic representation (Angell & Smithson, 1991). Angell and Smithson (1991) and Somerville (2007) define a socio-technical system as a system including hardware and software components, that has defined operational processes followed by human operators and that operates within an organization. It is therefore influenced by organizational policies, procedures and structures.

Land and Kennedy-McGregor (1987) define five distinct components of an IS, being (1) the informal human system, which has to do with culture and political aspects, (2) the formal human system, such as, regulations, roles and departmental boundaries, (3) the formal computer system, which is automation, (4) the informal computer system, which is unstructured information available by using personal computing and (5) the external system, which refers to links with the external world. Land and Kennedy-McGregor (1987) conclude that these five components are interlinked and that an organization can only operate at an effective level if all five components interact and if those who design systems are aware that the richness and diversity of the five components provide the strength which makes effective operation possible.

Brussaard and Tas (1980) apply a system theoretical approach and define “the information paradigm”. The information paradigm states that each dynamic system (such as an organization) can be represented in a “real system” (RS) and an IS. From a functional perspective the IS can best be defined in juxtaposition to real life systems such as public or private organizational units (Brussaard & Tas, 1980). RS are those parts or aspects of reality to be investigated as a whole in order to know or eventually control them. RS sends data to the IS, the IS processes the data in order to manage or control the RS by sending data back to the RS (Brussaard & Tas, 1980). This is illustrated in Figure 1.

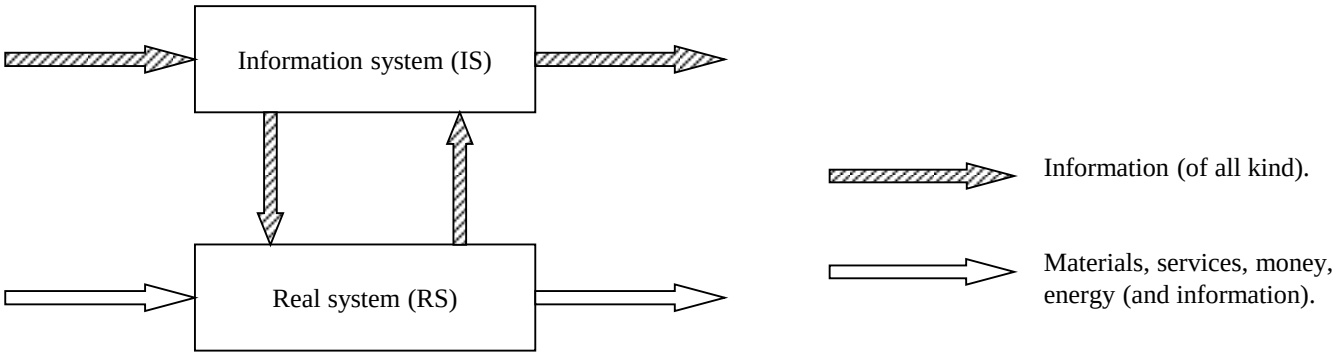


Figure 1. Information paradigm (Brussaard & Tas, 1980)

Due to its recursivity, every IS and RS can be divided into sub-systems and also remain part of a supra-system. On a higher level complete organizations or even chains of organizations can be labeled IS and RS (Brussaard & Tas, 1980). From a system theory perspective an IS can be defined from three perspectives (Berghout, 1997; Brussaard & Tas, 1980):

1. Functional perspective: an IS is used to identify states of a controlled system. The states that are identified can refer to the past (reports), the present (registrations) and the future (plans). IS can be used to transfer data, to other places (communication), in time (storage), in contents (processing).
2. Analytical perspective: an IS is comprised of a particular physical component which is used to provide the required data, these are: associated software, hardware, data sets, people and procedures.
3. Temporal perspective: IS are built, maintained and finally abandoned in a system life cycle.

Next the components of the IS are elaborated. According to the Oxford Dictionary of Business and Management (2016) software is defined as: *"the programs used with a computer, together with their documentation, as opposed to the physical parts of the computer system (hardware)".* Software or computer software is often divided into application software (also known as applications) and system software (O'Brien & Marakas, 2009) or support software (Sommerville, 2007). Application software requires system support software to operate (Sommerville, 2007). Support software includes programs, such as operating systems and utilities (Sommerville, 2007). Application software is a subclass of software that employs the capabilities of a computer directly and thoroughly to a task that the user wishes to perform (Sommerville, 2007). This should be contrasted with system software which is involved in integrating a computer's various capabilities, but typically does not directly apply them in the performance of tasks that benefit the user (Sommerville, 2007). Application software is defined as the application system that provides the business services which is usually composed of a number of separate programs that have been developed at different times (Sommerville, 2007). *"Each organization tends to define its applications differently, so it is important to have a single common definition to ensure consistency in the way application teams discuss the subject of the analysis"* (Scardino, Parameswaran, Young, & Buttorff Sikes, 2005). Based on O'Brien and Marakas (2009) and Scardino et al. (2005) and Sommerville (2007) it is concluded that software or computer software is divided into application software (also abbreviated by application) and system software or support software.

Hardware as a component of an IS is regarded as "*the physical equipment used for input, processing and output activities in an IS"* (Laudon & Laudon, 2001). "*It contains the computer processing unit; various input-, output- and storage devices; and physical media to link these devices together"* (Laudon & Laudon, 2001). According to Valacich and Schneider (2010), hardware is physical computer equipment, such as the computer, monitor, central processing unit or keyboard. A data set is defined as "*a collection of related information made up of separate elements that can be treated as a unit in data handling"* (Bangia, 2010). The people component in an IS are all the persons involved in the IS. Procedures are defined by the: "International Organization for Standardization" (ISO) as a specified way to carry out an activity or a process (ISO 9000:2005, Clause 3.4.5). Procedures are part of a process. Since each process requires a set of procedures, there is a number of procedures within a process. Examples of generic procedures are described in best practices guides, such as: ITIL (OCG, 2007), ASL (Pols & Backer, 2006) or BISL (Pols & Backer, 2007).

All components of an IS are interlinked with each other and an organization can only operate at an effective level if all components interact and if those who design systems are aware that the richness and diversity of the components provide the strength which makes effective operation possible (Land & Kennedy-McGregor, 1987). Figure 2 illustrates the five components of an IS which are interrelated with each other and defines the IS artifact of this research. Within the illustration the support software or system software (e.g. operating system) is considered to be part of the hardware (Sommerville, 2007). It should be emphasized that the components of IS are dynamic, the relative share of the components (e.g. hardware, software and people) of the IS will change in time.

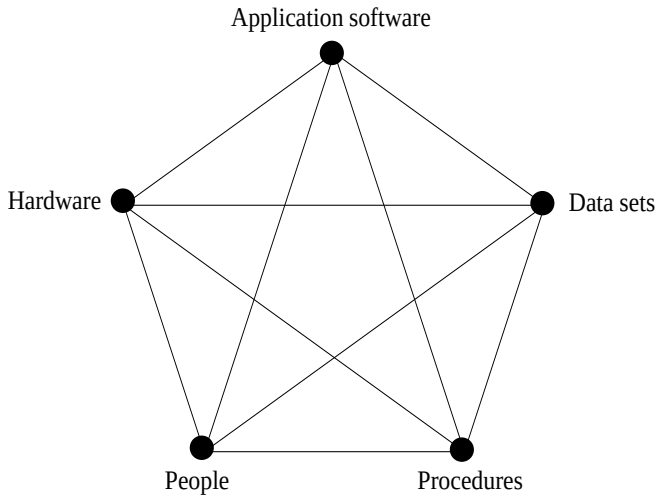


Figure 2. The IS Pentagram (based on the components of an IS as defined by Brussaard and Tas, 1980 and the interlinkage of the components as defined by Land and Kennedy-McGregor, 1987)

Due to the fact that legacy research on sub-IS-level will differ from research on supra-IS-level, it is necessary to define different levels of IS within this dissertation. The lowest IS level is defined as application software level, which includes a single application. The IS concerns the intermediate level, which refers to the logical entity supporting a particular business process and usually contains 1-10 applications. The "information function" concerns the highest level and typically supports a business unit and consists of more than 10 applications. All the information functions within an organization are referred to as the information provisioning of the organization. In Table 1 the three hierarchy levels are illustrated.

Hierarchy level	**Supra-IS**	**IS**	**Sub-IS**
Definition within this dissertation	Information function	Information system (IS)	Application software
Number of applications	>10 applications.	1-10 applications	Single application (can be composed of separate programs)

Table 1. Different IS hierarchies defined and named as used in this dissertation

1.1.2 Legacy information systems

The former section concerns the definition of IS in general. In this section, legacy IS are elaborated upon. Several definitions can be found regarding legacy IS. For instance, Brodie and Stonebraker (1995) define a legacy IS as "*any IS that significantly resists modification and evolution to meet new and constantly changing business requirements"*. Brooke (2002) defines a legacy system as "*a computer system, situated within a particular organizational environment, which no longer meets the needs of that environment"*. Furthermore, Brooke and Ramage (2001) state that the legacy IS is made up of technical components and social factors (such as software, people, skills, business processes), which no longer meet the needs of the business environment. Brooke (1994) also refers to the gap between the capabilities of the system and the needs of the business in which it is used. Ulrich (2002) presents an architectural view on legacy systems and defines a legacy architecture as a collective set of application software, data structures and operational platforms currently running in enterprise computing environments. Aging hardware, software and data architectures prevent organizations from fully exploiting computers and the value they bring in organizations, its customers and its partners (Ulrich, 2002). Aging legacy architectures can stymie critical business initiatives while preventing an enterprise from responding to competitive pressures in a timely fashion (Ulrich, 2002). Sommerville (2007) defines a legacy system as a socio-technical computer-based system that has been developed in the past, often using older or obsolete technology. These systems include not only hardware and software but also legacy processes and procedures, old ways of doing things that are difficult to change because they rely on legacy software (Sommerville, 2007).

Every system has the potential to become a legacy system, this is not a matter of time or age, all that is required is the occurrence of an appropriate event (Alderson & Shah, 1998). Brooke and Ramage (2001) state that the word legacy also relates to what is left after a particular event occurs (typically, someone's death). Legacy events will also vary from company to company. The key point for all of these examples is that IS become legacy IS when somebody recognizes an important legacy characteristic (Brooke & Ramage, 2001). In addition, the event may not be a one-off and it could also result from internal organizational circumstances as well as external ones (Brooke & Ramage, 2001). Long (2006) states that the connotation "legacy application" is, itself, becoming a legacy term because length of time is no longer the metric by which an application becomes a legacy application. With the speed at which business requirements and technology change in today's environment, the application placed into production just yesterday may be tomorrow's legacy application (Long, 2006). According to Lawrence (2007) legacy IS refers to existing IT assets that have been deployed in the past. These assets could have been installed anywhere from yesterday to twenty years ago and in many cases the legacy investment is running critical business processes (Lawrence, 2007).

Legacy characteristics may apply to all individual components of the IS (see Figure 2, p. 5). For example *application software* could suffer from structural deterioration or increasing complexity (Lehman, 1980a; Lehman, 1980b). Inadequate, limited or missing documentation of the application software is also mentioned by several researchers (Adolph, 1996; Kelly, Gibson, Holland, & Light, 1999; Parnas, 1994; Seacord et al., 2003). Legacy *hardware* are characterized by slow, or inadequate performance (Adolph, 1996; Bisbal et al., 1999; Daga et al., 2005; Kelly et al., 1999; Parnas, 1994). Some old hardware may have a poor reliability (Parnas, 1994) or the hardware supplier could discontinue the product line (Adolph, 1996). Legacy *data* may refer to data inconsistencies, duplicated files and aging databases. Legacy data tends to be stored redundantly across multiple stovepipe business units and applications; the same or similar data is inconsistently defined across multiple systems; the same data terminology may be used to define different data across multiple applications and business units; the integrity of the data may be poor and contain inappropriate information; data may not be easily accessible by modern systems or through user-based inquiries; data cannot be readily shared across systems, business units and organizational boundaries (Ulrich, 2002). Examples of *people-related* legacy characteristics are, skills that are increasingly hard to find or retiring staff (Adolph, 1996; Daga et al., 2005; Warren, 1999). There could also be a growing dislike of the tedious work on legacy IS, *"if something is difficult, tedious and slow, people will try to avoid doing it"* (Adolph, 1996). Engineers prefer working on new system development, instead of maintaining old software (Bennet, 1995; Ulrich, 2002). Legacy *procedures* typically do not take advantage of streamlined facilities and standard routines; use programming and design practices that current management would not permit (Zvegintzov, 1984). Business knowledge is always locked into a specific technology at any given time (Daga et al., 2005) and business processes may be designed around legacy systems and constrained by functionality that it provides (Sommerville, 2007).

Besides looking through the viewpoint of the components from a system theory perspective (Berghout, 1997; Brussaard & Tas, 1980), also other perspectives are possible: a technical perspective, a functional perspective and an economical perspective are discerned.

From a *technical* perspective IS get legacy characteristics when there is a lack of technical maintenance or there are technical opportunities (Adolph, 1996; Furneaux & Wade, 2011; Swanson & Dans, 2000; Warren, 1999; Zvegintzov, 1984), e.g. through sustained maintenance over many years, the structure of the program code has weakened.

From a *functional* perspective IS get legacy characteristics when an IS is not or only partly capable of delivering the required functionality (Brooke, 1994; 2002; Brooke & Ramage, 2001; Comella-Dorda, Wallnau, Seacord, & Robert, 2000; Seacord et al., 2003; Van den Heuvel, 2007). Comella-Dorda et al. (2000) describe a situation in which the business needs concerning functionality of an IS will increase over time and that by maintaining the IS, the IS will follow these business needs, until a moment in time where this is no longer possible.

From an *economical* perspective IS get legacy characteristics when an IS is expensive or there are cheaper alternatives available (Bisbal et al*.,* 1999; Brodie & Stonebraker, 1995; Hunter & Aron, 2006; Lehman, Kahen, & Ramil*,* 2000; Sakthivel, 1994). There are indications that legacy portfolios are 30-50% more expensive to run than their packaged systems or newer technology (Hunter & Aron, 2006). Of the IT budgets, 60-80% is consumed by maintaining current IS (Brodie & Stonebraker, 1995).

Legacy IS primarily have a negative connotation. There are, however, also authors who emphasize their positive aspects. Legacy IS often run critical business processes and are often considered the cash cows of the enterprise, generating unusually high profit Lawrence (2007). Bakehouse and Wakefield (2005) write that staff are familiar with all aspects of the system, reliability of systems, critical for the business, holding all business information (Bakehouse & Wakefield, 2005). Light (2003) states that it is clear that legacy IS (mellow and antediluvian), may sometimes offer immense value to organizations. Legacy IS are a valuable source (in some cases the only source) of business knowledge, which serves as a precious resource for future improvements to the IS and the organization as a whole (Daga et al., 2005). Bennet (1995) states that, *"we must recognize that, historically, the information requirements of an organization may have been achieved through the continual maintenance and fine tuning of legacy systems, leading to the perception that it represents a best fit for management".* Additionally, the legacy software may be seen as *"very reliable and responsive to customers' needs*" (Bennet, 1995). "*Software does not suffer from wear, tear, corrosion or pollution"* (Lehman, 1980b). Unlike mechanical machine parts, software does not wear out from use (Long, 2006). Where friction and other environmental factors may, over time and use, cause a gear part within an assembly such as a transmission to wear out, software does not exhibit this characteristic (Long, 2006). As long as a software application continues to deliver business value and curiously sometimes even when it does not, it can remain in operation for a very long time (Long, 2006).

In summary, it can be stated that there are many perspectives to describe legacy IS. Besides its individual components, software, hardware, data, people and procedures (Berghout, 1997; Brussaard & Tas, 1980), there is also the technical, functional, and economical perspective. Legacy IS are not only associated with negative connotations and often run the most critical business processes (Lawrence, 2007). Occasionally, replacement of the legacy IS seems unavoidable, simply because one of the vendors announces to terminate support of particular hardware or software components. Most of the time, the situation will be much more complex. Technically IS will hardly wear out (Lehman, 1980b; Long, 2006). The interaction of all components, functionalities and the environment of the IS and the environment of the organization, make the system aged. Brodie and Stonebraker (1995) note that a legacy IS is "*any IS that significantly resists modification and evolution to meet new and constantly changing business requirements"*, where the Oxford Dictionary of English (2010) defines "significantly" as "*in a sufficiently great or important way as to be worthy of attention"*. This also implies that many stakeholders could pertain many perspectives leading to other thresholds regarding worthy of attention or resistance to change. This research will cohere to Brodie and Stonebraker's (1995) definition of legacy IS, being:

"A legacy IS is any IS that significantly resists meeting organizations' requirements".

1.1.3 Decision making

Decision making concerning legacy IS within organizations is done by managers, and Harrison (1999) provides the following characteristics of managerial decision making in the decision making process. The decision making process starts with setting the managerial objectives or requirements (Harrison, 1999). Subsequently there is a search for alternatives to gain these managerial objectives. This process of searching for alternatives will draw new insights in the managerial objectives, and based on these new insights the managerial objectives might be revised. The alternatives are compared and evaluated. Based on this evaluation the next phase is the act of choice. This is a separate stage because it is possible to keep comparing and evaluating without actually making a choice. The next stage is implementing the decision. Based on the implementation new insights can be gained which will result in the search of possible new alternatives. In addition, there will be the phase of follow up and

control. The difference between implementing and the managerial objectives can lead to corrective actions in implementation, which again may lead to additional search for alternatives and might even lead to setting new objectives. This process is illustrated in Figure 3.

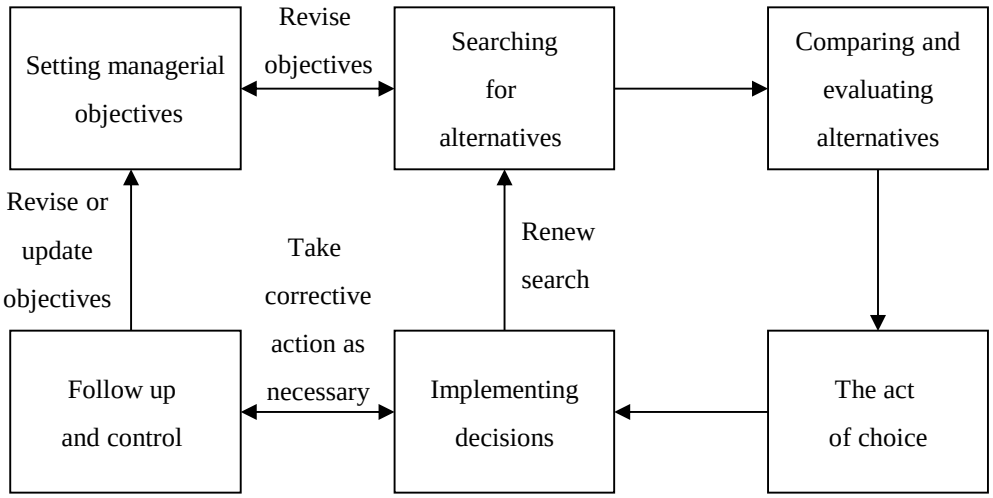


Figure 3. The managerial decision making process of Harrison (1999)

Harrison (1999) describes an interdisciplinary framework of decision making. The interdisciplinary framework of decision making contains the following disciplines: philosophy, psychology, mathematics, law/anthropology/political science, sociology/social psychology and economics/statistics. These disciplines may also apply to the abandonment decision making of legacy IS. Furthermore, Harrison (1999) suggests an eclectic approach concerning decisions, where eclectic implies, "*selecting what appears to be best in various doctrines or styles*" (Harrison, 1999). Concerning legacy decision making several authors prefer a technical perspective (Adolph, 1996; Furneaux & Wade, 2011; Swanson & Dans, 2000; Warren, 1999; Zvegintzov, 1984). Others suggest a business functionality perspective (Brooke, 1994; 2002; Brooke & Ramage, 2001; Comella-Dorda et al*.,* 2000; Seacord et al*.,* 2003; Van den Heuvel, 2007). These authors emphasize the gap between the supplied business functionality of the application and the required business functionality. Different interdisciplinary approaches to decision making may be viewed as decision making models, because they represent a particular segment of the real world at a given time and place under varying conditions (Harrison, 1999). Another decision making perspective "muddling through" is advocated by Lindblom (1959). Instead of taking large steps to solve a problem, he advocates a theory of incrementalism (small steps) in policy and decision-making. For the definition of abandonment decision making, this dissertation will cohere to Harrison's (1999) definition of managerial decision making, being, a moment of choice in an ongoing process of evaluating alternatives for meeting an objective, at which expectations about a particular course of action impel the decision maker to select that course of action most likely to result in attaining the objective.

1.2 Primary research question

Earlier studies of Furneaux (2009) and Swanson and Dans (2000) emphasize to take a more interpretative stance and study legacy decision making cases earlier in their life cycle and to study the actual abandonment decision from the perspective of multiple stakeholders. In their studies it turned out to be difficult to identify the true arguments underlying legacy decision making. Furneaux (2009) suspects that decision makers sometimes identify (the same) arguments after the abandonment decision has been made. This research should therefore be in-depth, because particularly in legacy IS, the *context* of the arguments, constraints and opportunities will determine the way stakeholders enact (Avgerou, 2002). This research attempts to deepen the understanding of legacy IS and abandonment decision making. Furthermore, a method is proposed to support decision making regarding the abandoning of legacy IS. How are legacy IS identified and how are they managed? What does aging of legacy IS imply? Which components of IS age, and how? Why do organizations subsequently decide to

abandon IS? What good practices regarding the management of IS, or decision support methods for abandonment decision making could be identified?

The primary research question is formulated as:

How do organizations identify legacy IS and how do they manage abandonment?

The preferred research approach will be elaborated upon in Chapter 2.

1.3 Dissertation outline

The research approach used and presented in this dissertation is as follows:

- In Chapter 2 the research design is presented, this includes the chosen research philosophy, research approach, research strategy, research choice, time horizon, techniques and procedures.
- In Part 1 (Chapter 3, 4 and 5) the results of the "*exploration phase"* are described. In Chapter 3 existing theory is discussed. This includes existing theory concerning IS management, a systematic literature review on legacy IS abandonment decision making and practical abandoning of legacy IS. In Chapter 4 the complex background of Case A is described. This includes a description of how Case A abandoned 471 IS. In Chapter 5 the longitudinal exploratory study at Case A is described.
- In Part 2 (Chapter 6) the results of the "*explanation phase"* are described. This includes aging, abandonment triggers, decision making processes and the practical abandoning processes of Legacy IS. Furthermore, design dilemmas for a method to abandon legacy IS are conceptualized.
- In Part 3 (Chapter 7) the results of the "*testing phase"* are described. The concepts of Chapter 6 are confirmed, further enhanced and validated in four other case organizations.
- In Part 4 (Chapter 8) the results of the "d*esign phase"* are described. A method to abandon legacy IS is proposed, based on the results obtained in the previous phase.
- In Chapter 9 the conclusions of this research are described and recommendations for further research are provided. This final chapter (Chapter 9) is followed by the appendices and the references.

The outline of this dissertation is illustrated in Figure 4.

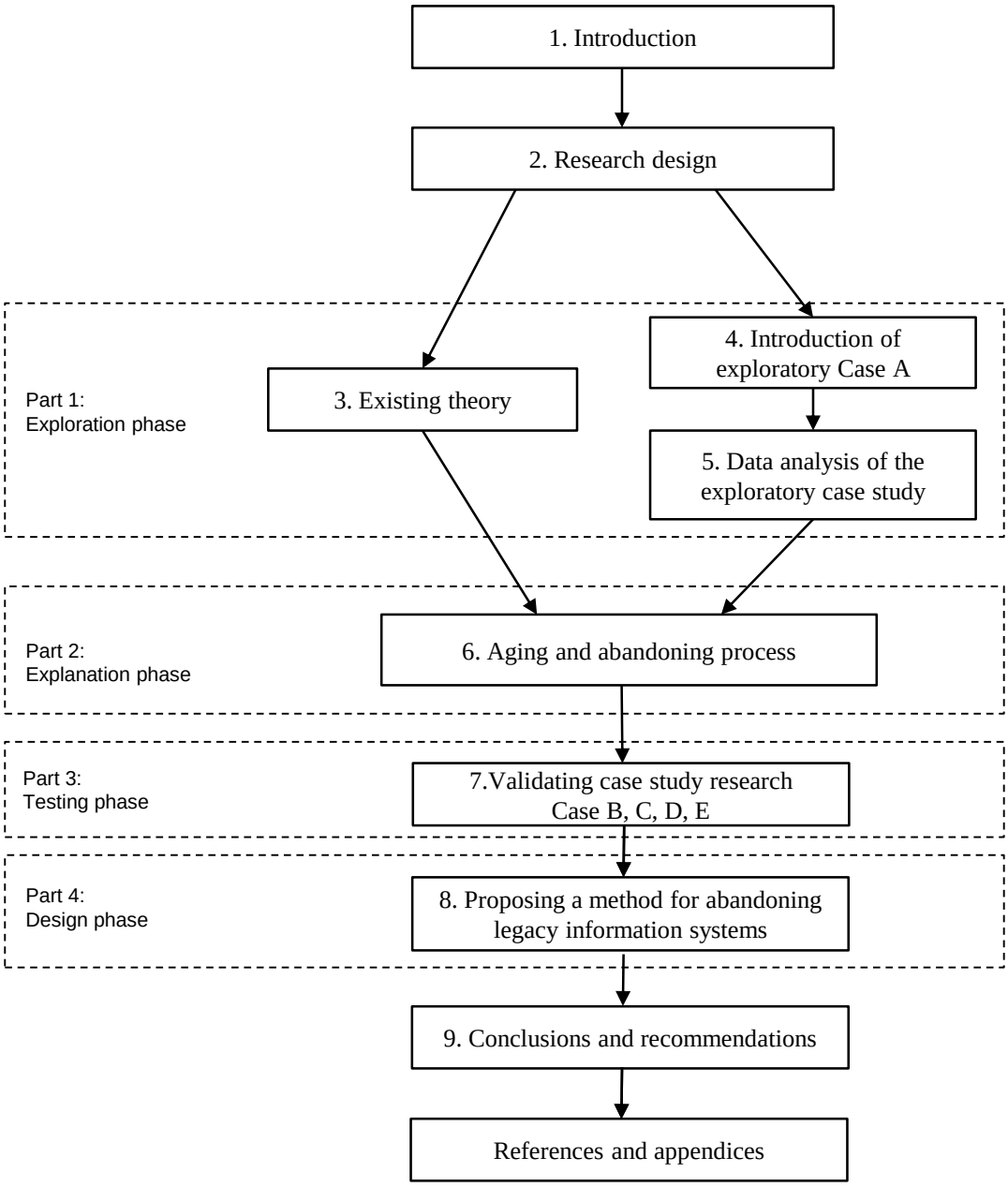


Figure 4. Dissertation outline

2 RESEARCH DESIGN

"There is, however, no universal recipe for scientific advance. It is a matter of groping forward into terra incognita of the outer world by means of methods which should be adapted to the circumstances".
---Bemmelen, R. W. (1961)---

2.1 Introduction

In this chapter the research design of this dissertation is presented. The research design is defined as the logical sequence that connects the empirical data to a study's initial research questions and ultimately its conclusions (Yin, 2003). Design of this research is based on the research "onion" of Saunders, Lewis and Thornhill (2007). This research "onion" includes the following layers, research philosophy, research approach, research strategy, research choice, time horizon, technique and procedure (Saunders et al., 2007). In designing a research to answer the research questions, choices have to be made within these layers. The research "onion" (Saunders et al., 2007) is illustrated in Figure 5.

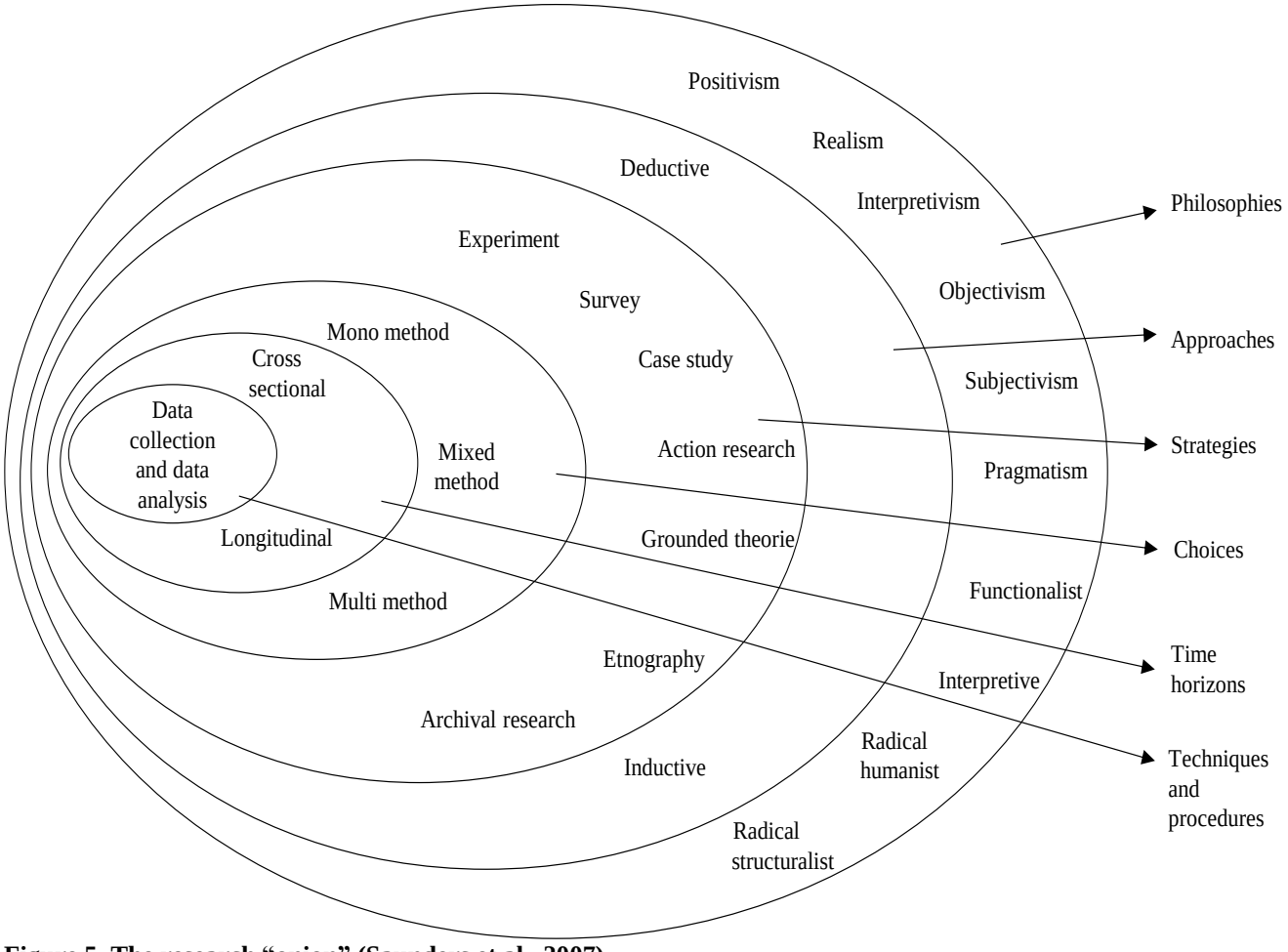


Figure 5. The research "onion" (Saunders et al., 2007)

In accordance with the research "onion" the research design of this study is described (Saunders et al., 2007). First the research philosophy (Section 2.2) is elaborated on, subsequently the research approach (Section 2.3) and the research strategy (Section 2.4) are detailed, followed by research choices (Section 2.5), time horizons (Section 2.6) and research techniques and procedures (Section 2.7). Then the overall research design is presented (Section 2.8.), followed by the role of the researcher (Section 2.9). Finally in the summary (Section 2.10) the various choices regarding the research design are summarized.

2.2 Research philosophy

In this section the first layer of the research onion of Saunders et al. (2007), the research philosophy, is discussed. A discussion of the research philosophy is essential before embarking on a research project, due to the fact that every research tool or procedure is inextricably embedded in commitments to particular visions of the world (Hughes, 1990). Also different assumptions adopted towards reality and to the obtainment of knowledge might lead to different outcomes (Hirschheim & Klein, 1989). Guba and Lincoln (1994) define three basic philosophical questions:

1. *What is there that can be known?* This is usually called the ontological question. Ontology is that branch of philosophy (specifically, of metaphysics) that is concerned with issues of existence or being as such. Another way to phrase the question is: "*what is the nature of reality?"* (Guba & Lincoln, 1994).
2. *What is the relationship of the knower to the known (or the knowable*)? Epistemology is that branch of philosophy that deals with the origin, nature and limits of human knowledge. Another way to phrase the question is this: "*how can we be sure that we know what we know?"* (Guba & Lincoln, 1994).
3. *What are the ways of finding out knowledge*? This is usually called the methodological question. Methodology is a more practical branch of philosophy that deals with methods, systems and rules for the conduct of inquiry. Another way to phrase the question is: "*how can we go about finding out things?"* (Guba & Lincoln, 1994).

Ontology is a branch of philosophy or metaphysics concerned with the nature and the relations of being (Remenyi, Williams, Money, & Swartz, 2005). Whether the object of investigation is the product of consciousness (nominalism) or whether it exists independently (Realism) (Remenyi et al., 2005). Saunders et al. (2007) distinguish subjectivism (phenomena are created from the perceptions and consequent actions of social actors) and objectivism (social entities exist in reality) (Saunders et al., 2007). This dissertation attempts to deepen the understanding of legacy IS and abandonment decision making. Furthermore, a method is proposed to support decision making regarding the abandoning of legacy IS. The primary research question is formulated as: *how do organizations identify legacy IS and how do they manage abandonment?* Applying the ontology philosophy to this research, it should be emphasized that the identification and abandonment management is related to IS within organizations. In this research organizations are constructed and comprised of humans and are considered to be social constructed realities. Also, as defined earlier, people are part of an IS; decision making concerning the abandoning of legacy IS is a human activity. This should suit a nominalistic or subjectivist ontology.

Epistemology is the theory of knowledge, it is what our grounds of knowledge are (Remenyi et al., 2005). Epistemology refers to the type of knowledge that can be obtained about a phenomenon under study (a continuum for epistemology runs from positivism to anti-positivism) (Cornford & Smithson, 2006). Three distinct epistemological stances can be identified: the positivism, the realism and the interpretivism stances (Saunders et al., 2007):

1. Positivism - the researcher will be concerned with facts rather than impressions, there is an observable social reality. The assumption is that the researcher is independent of and neither affects nor is affected by the subject of the research (Remenyi et al., 2005). The research is deemed value free.
2. Realism - the essence of realism is that objects have an existence independent of the human mind (Saunders et al., 2007).
3. Interpretivism is an epistemology that advocates that it is necessary for the researcher to understand differences between humans in our role as social actors. This emphasizes the difference between conducting research among people rather than objects such as trucks and computers (Saunders et al., 2007). Crucial to the interpretivist epistemology is that the researcher has to adopt an empathetic stance (Saunders et al., 2007). The challenge here is to enter the social world of our research subjects and understand their world from their point of view (Saunders et al., 2007). Positivism has trouble explaining why so many people hate their jobs (Remenyi et al., 2005). The more traditional natural life scientist regards such approaches as being inferior, it is however increasingly accepted that phenomenology is better suited for this type of research where the central issues concern people and their behavior (Remenyi et al., 2005). Critics to the positivists

in the world of business and management argue that rich insights into this complex world are lost if such complexity is reduced entirely to a series of law like generalizations (Saunders et al., 2007).

It can be concluded that applying the epistemology to the primary research question: "How do organizations identify legacy IS and how do they manage abandonment?" results in the following possible research epistemology. For the research on decision making within organizations done by humans an interpretivism epistemology is a good fit. Concerning the IS which has the component people, this people component also fits an interpretive stance. Regarding other parts of the IS, which are software, hardware, data sets and procedures, a positivistic stance according the epistemology can also be applied.

Different possible ontological and epistemological viewpoints of this research are described above. By choosing one philosophical stance another valuable philosophical stance could be discarded. According to Kanellis and Papadopoulos (2009), *"any research activity seeks valid knowledge, this validity stems from community acceptance that is an agreement on a set of values which have produced knowledge claims".* Chua (1986) suggests that a community of scientist share "a constellation of beliefs, values and techniques" and these beliefs circumscribe definitions of worthwhile problems and acceptable scientific evidence. This set of values is referred to as a "research paradigm" (Kanellis & Papadopoulos, 2009).

A paradigm is a fundamental set of assumptions adopted by a professional community that allows its members to share similar perceptions and engage in commonly shared practices (Hirschheim & Klein, 1989). Mingers (2001) suggests that IS research has to draw upon a wide range of disciplines; technology, psychology, economics, sociology, mathematics, linguistics and semiotics, which encompass different research traditions. This puts IS in a position similar to other management areas such as organizational studies, which are also characterized by a plurality of research paradigms, each with particular research methods. Therefore Mingers (2001) advocates a plurity of research paradigms.

A framework to integrate the positivist and interpretivist approaches to organizational research is proposed by Lee (1991). The proposed framework provides a demonstration of the feasibility of integrating two approaches often believed to be opposed and incompatible when performing organizational research (Lee, 1991). The combined framework fully accounts for an additional, critical feature of social reality that distinguishes it from the physical subject matter of the natural sciences – namely, the subjective understanding while retaining all the rigors of the natural science model of traditional positivism (Lee, 1991). Based on Robey (1996) who argues that a diversity of research methods and paradigms within the discipline is a source of strength, Mingers (2001) suggests a pluralist approach to IS research, or even a plurality of paradigms. Results will be richer and more reliable if different research methods, preferably from different paradigms, are routinely combined together (Mingers, 2001).

The difference between various philosophies is a continuum, rather than opposite positions (Tashakkori & Teddlie, 1998). They argue that the most important determinant of the research philosophy adopted is the research question. One approach may be better than the other for answering particular questions (Saunders et al., 2007). It is perfectly possible to work with both philosophies (positivist and interpretive) (Saunders et al., 2007). Mixed methods, both qualitative and quantitative are possible and possibly highly appropriate within one study (Saunders et al., 2007). Based on the discussion of research philosophy as described in this section, this research will adopt this pragmatic paradigm (Tashakkori & Teddlie, 1998).

2.3 Research approach

Two research approaches are distinguished, the deductive and inductive research approach (Saunders et al., 2007). Mason (2002) describes deductive reasoning as the "hypothetico-deductive method", whereby theoretical propositions or hypotheses are generated in advance of the research process and then modified usually through a process of falsification by the empirical research. Deduction is defined by Remenyi et al. (2005) as the process of deriving conclusions by logical reasoning in which the conclusion about particular issues follows

necessarily from general or universal premises. Within inductive reasoning the researcher will develop theoretical propositions or explanations out of the data in a process which is commonly seen as moving from the particular to more general (Mason, 2002). Induction is defined by Mason (2002) as the inference of a generic or generalized conclusion from the observation of particular instances. The major differences between deductive research approaches and inductive research approaches are listed in Table 2.

Deduction emphasis	**Induction emphasis**
• Scientific principles • Moving from theory to data • The need to explain causal relationships between variables • The collection of quantitative data • The application of controls to ensure validity of data • The operationalization of concepts to ensure clarity of definition • A highly structured approach • Researcher independence of what is being researched • The necessity to select samples of sufficient size in order to generalize conclusions	• Gaining and understanding of the meanings humans attach to events • A close understanding of the research context • The collection of qualitative data • A more flexible structure to permit changes of research emphasis as the research progresses • Less concern with the need to generalise

Table 2. Major differences between deductive and inductive approaches to research (Saunders et al., 2007)

This research had the opportunity to first explore a longitudinal case study in which 471 IS had been abandoned and where unlimited access was provided to all documents and decision makers. Therefore this research first follows an inductive approach within the explorative case organization A. By means of inductive reasoning the researcher will subsequently develop theoretical explanations concerning *"aging", "abandonment decision making" and "the practical abandoning of legacy IS"*. Furthermore, dilemmas for a method to abandon legacy IS will be provided. These findings will subsequently be confirmed, further enhanced and validated in four additional case studies (Chapter 7). Saunders et al. (2007) confirm that it is perfectly possible to combine deduction and induction within the same piece of research.

2.4 Research strategy

There are seven research strategies for collecting data, being: experiment, survey, case study, action research, grounded theory, ethnography and archival research (Saunders et al., 2007). The research strategy depends on the type of the research question, the extent of control an investigator has over actual behavioral events and the degree of focus on contemporary as opposed to historical events (Yin, 2003), as depicted in Table 3.

Strategy	**Type of research question**	**Requires controls of behavioral events**	**Focuses on contemporary events**
Experiments	How, why	Yes	Yes
Survey	Who, what, where, how many, how much	No	Yes
Archival analyses	Who, what, where, how many, how much	No	Yes/No
History	How, why	No	No
Case study	How, why	No	Yes

Table 3. Different research strategies (Yin, 2003)

The primary research question is: "*How do organizations identify legacy IS and how do they manage abandonment?"* (see Section 1.2). Supporting research questions in this research typically focus on "how" and "why" questions, such as, (1) *"how do legacy IS age?",* (2) "*why do organizations decide to abandon legacy IS?",* (3) *"how do organizations practically abandon legacy IS?"* Answering these research questions will deepen the understanding of legacy IS and abandonment decision making and will provide input towards the design of methodological support, which is defined as (4) *"propose a method to abandon legacy IS".* Yin (2003) states that in general case studies are the preferred strategy when "*how*" and "*why*" questions are posed and the

investigator has limited control over the event and focusses on a contemporary phenomenon within some real life content (Yin, 2003). The case study method allows investigators to retain the holistic and meaningful characteristics of real-life events such as organizational and managerial processes (Yin, 2003).

In the first case organization and also in the subsequent validating case studies, experiments turned out to be hardly possible, because the organization would not allow interferences in their decision making processes. The legacy decision making process simply included too many delicate decisions. Furthermore, archival studies would not have been possible, because of the limited availability of archival data. Therefore, case study research was selected as preferred research strategy.

2.5 Research choice

Saunders et al. (2007) refer to two main research choices, being mono methods or multiple methods. Mono methods use a single data collection technique and a single corresponding data analyze procedure. Multiple methods are split into multi-methods and mixed-methods (Saunders et al., 2007). Multi-methods use multiple data collection techniques and analysis procedures in either qualitative or quantitative domain, however, quantitative and qualitative research is not mixed. Mixed-methods also use more data collection techniques and analysis procedures in the quantitative data and qualitative data (Saunders et al., 2007). Mixed-methods can be categorized in "mixed-method research", which uses qualitative analysis techniques for qualitative research and quantitative analysis techniques for quantitative data and "mixed model research" in which quantitative data is transformed and analyzed as qualitative data. This way, mixed-methods can be used for triangulation (Saunders et al., 2007). The most important advantage of using multiple sources of evidence is the development of converging lines of inquiry, a process of triangulation (Yin, 2003). Triangulation refers to obtaining evidence from multiple sources to avoid single and possibly biased views (Remenyi et al., 2005). The research choices defined by Sanders et al. (2007) are illustrated in Figure 6.

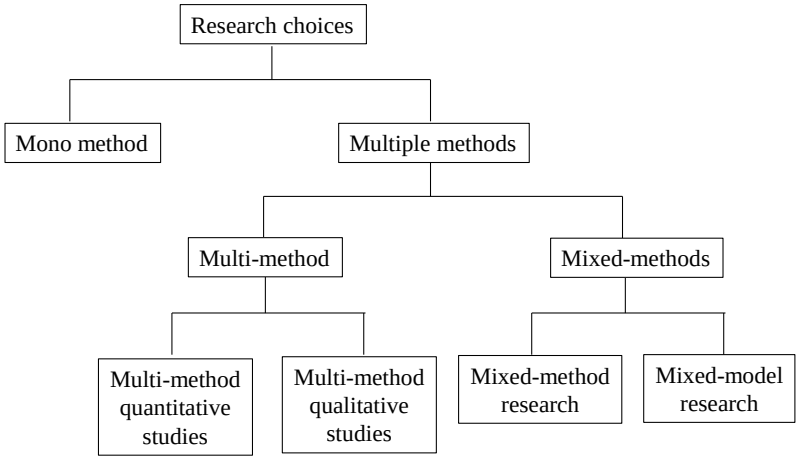


Figure 6. Research choices (Saunders et al., 2007)

Presumably, the multiple methods and multiple disciplines involved in this research (e.g. economics, sociology, technology, psychology, and politicology) will provide richer and more reliable results in this study (Lee, 1991; Mingers, 2001). To assure this research provides richer and more reliable results, this research is using multiple methods and multiple disciplines.

2.6 Time horizon

Longitudinal research is used to describe a study that extends over a substantial period of time and involves studying changes over time (e.g. 5 or 10 years). Longitudinal studies have considerable potential for yielding rich data that can trace changes over time, and with great accuracy (Gorard, 2001). Strengths of longitudinal studies are (Cohen, Manion, & Morisson, 2007), (1) *separates real trends from chance occurrence,* (2) *enables change to be analyzed at the individual level,* (3) *enables the dynamics of chance to be caught,* (4) *gathers data*

contemporaneously rather than retrospectively and thereby avoids the problems of selective or false memory. Weaknesses of longitudinal studies are (Cohen et al*.,* 2007), (1) *time consuming and expensive, it takes a long time for the studies to be conducted and the results to emerge, and* (2) *the data, being rich at an individual level, are typically complex to analyze.* Furthermore, cross-sectional studies take a snapshot of a situation (Remenyi et al., 2005). Strengths of cross-sectional studies are that these studies are "*comparatively quick to conduct and cheap to administer"* (Cohen et al*.,* 2007). Weaknesses are that these studies only "*catch a particular moment in time and may be ineffective to study change"* (Cohen et al., 2007). In business and management studies longitudinal research usually offers the best opportunity to obtain useful insights into practices and policies (Remenyi et al., 2005).

The first section of this dissertation, which is the exploration phase, concerns longitudinal research. The legacy abandoning team in the first case organization (Case A) is followed over a five year period along with semi-structured interviews and desk-research (e.g. monthly progress reports, presentations and communication of the abandoning of legacy IS). Although longitudinal research is time consuming and expensive, it will generate rich data and disclose dynamics of change, providing rich insights in the process of abandoning legacy IS. Based on the longitudinal research of Case A, conceptualizations concerning the aging, the abandonment decision making and the practical abandoning of legacy IS are provided (Chapter 6), these conceptualizations are subsequently confirmed, further enhanced and validated in a cross-sectional study of four other organizations (Chapter 7).

2.7 Technique and procedure data collection and data analysis

This research decided to opt for case study research, some researchers nevertheless disdain case studies because Yin (2003):

1. Lack of rigor of the case study research.
2. Case study research provides little basis for scientific generalization.
3. Take too long and result in massive, unreadable documents.

In handling these concerns with regard to case study research it is therefore important to follow systematic procedures and to rigorously report all evidence (Yin, 2003). This can be performed by having a case study protocol (CSP) and recording and transcribing all interviews. Further, if more cases can be investigated, multiple case designs are likely to be stronger than single case designs (Yin, 2003). Finally the case study design needs to maximize four conditions related to design quality: a) *construct validity*, b) *internal validity*, c) *external validity*, d) *reliability* (Yin, 2003).

Construct validity is defined by Saunders et al. (2007) as the extent to which your measurement questions actually measure the presence of those constructs you intended to measure. Internal validity is the degree of validity of statements made about whether X causes Y (Remenyi et al., 2005). External validity is concerned with knowing whether the researcher's findings are generalizable to a wider universe beyond the immediate research environment (Remenyi et al., 2005). External validity is also referred to as generalizability (Saunders et al., 2007). With reliability (Yin, 2003) the objective is to be sure that if a later investigator follows the same procedures as described by an earlier investigator and conducted the same case study all over again, investigator should arrive at the same findings and conclusions.

Concerning techniques and procedures, for the exploration and the testing phase of this research, case study protocols (CSP) are designed. A CSP is recommendable as it increases the reliability of case study research and guides the researchers in carrying out their data collection (Yin, 2003). The CSP includes the instrument (the questionnaire) as well as the procedures and general rules. The CSP (including questionnaire) is subject to review prior to the conduct of the first interview. Subsequent interviews had been recorded, transcribed and are inserted in a research database. To avoid seeking conformance, the CSP described that interviewer had to organize critical reflection and further probing of findings. Data analyses is supported by ATLAS.ti software for qualitative data analysis.

2.8 Overall research design

This research attempts to deepen the understanding of legacy IS and abandonment decision making. Furthermore, a method is proposed to abandon legacy IS. The primary research question is: *"how do organizations identify legacy IS and how do they manage abandonment?"* This study includes two empirical cycles and two design cycles (Van Engelen & Van der Zwaan, 1994). The first empirical cycle, which is defined *"exploration phase"* consists of a literature review and an in-depth longitudinal case study to obtain evidence, how IS age and how organization abandon legacy IS (first empirical cycle) (Van Engelen & Van der Zwaan, 1994). In the subsequent design cycle, defined *"explanation phase"*, the aging and abandoning process of legacy IS is conceptualized (first design cycle). The subsequent empirical cycle concerns the *"testing phase"* (Van Engelen & Van der Zwaan, 1994) in which the findings of the design cycle are subsequently validated in four validating cases (second empirical cycle). Subsequently, in the final design cycle (which is defined "*design phase*"), a method to abandon legacy IS will be proposed (second design cycle). The overall research design of this study is illustrated in Figure 7.

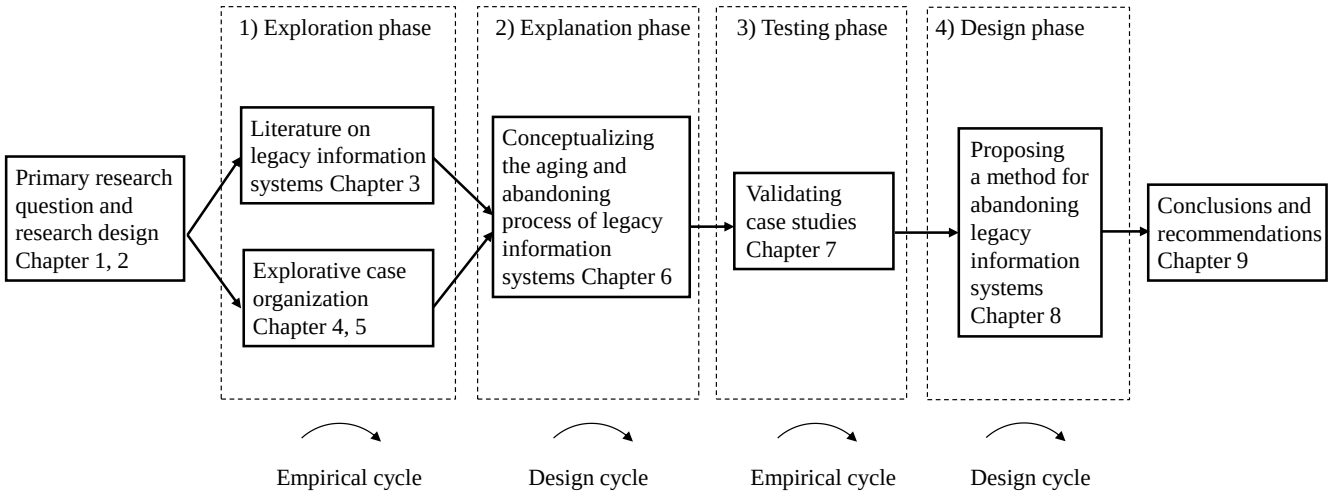


Figure 7. Overall research design

2.9 Role of the researcher

This research mainly uses an interpretive approach to its subject matter. This implies that researchers study artefacts in their natural settings, attempting to make sense of it, or interpret, phenomena in terms of the meanings people bring to them (Pettigrew, 1985). To avoid seeking conformance, the CSP described that the interviewer required to have a critical reflection and required to do further probing of issues. The researcher prepared and coordinated the case studies and performed the later analysis. During the "e*xploration phase*" the role of the researcher is best characterized as observer and interpreter, whilst during "*testing phase"* the role of the researcher is best characterized as observer and evaluator (Sassenburg, 2006). In both phases, researcher did not become involved as a consultant, avoiding action research. The researcher limits his activities to observing the study phenomenon and conducted case studies without intervention. Crucial to the interpretivist epistemology is that the researcher has to adopt an empathetic stance (Saunders et al., 2007). The challenge here is to enter the social world of our research subjects and understand their world from their point of view (Saunders et al., 2007).

2.10 Summary and conclusions

This research is initiated to improve the understanding of legacy IS and abandonment decision making. Furthermore this study will develop methodological support. The study has characteristics of both a positivistic and an interpretivistic research philosophy, the preferred research philosophy is pragmatism (Tashakkori & Teddlie, 1998). The most important determinant of the research philosophy adopted is the research question

(Tashakkori & Teddlie, 1998). This implies that different research methods are combined, resulting in richer and more reliable results (Mingers, 2001). The appropriate strategy for studying aging, abandonment decision making and practical abandoning of legacy IS in a practical context is found to be case studies, both for the "*exploration phase*" and the "*testing phase*". This research starts with a literature research on legacy IS. Subsequently, a longitudinal and explorative case study is researched. In this first case study, the aging process of legacy IS and subsequent abandonment decision making and practical abandoning are studied over a five year period. Such a relative long period is probably necessary to capture the dynamics of change associated with legacy IS and abandonment decision making (Furneaux, 2011). In the subsequent "*explanation phase*" the notions of aging of legacy IS and abandonment decision making are elaborated upon and conceptualized. In the testing phase the research findings are subsequently validated in four other case organizations by means of a cross-sectional research. Subsequently, a method to abandon legacy IS is proposed in the "*design phase*". Finally, the conclusions and recommendations for further research are given, an overview is provided in Table 4.

Layer	**Exploration phase**	**Explanation phase**	**Testing phase**	**Design phase**
Philosophy	Pragmatism (Positivism/Interpretivism)			
Approach	Inductive	Inductive	Inductive/Deductive	Inductive/Deductive
Strategy	Case study	-	Case studies	-
Choice	Multiple methods	-	Multiple methods	-
Time horizon	Longitudinal	-	Cross-sectional	-
Technique and procedure	Interviews, questionnaire, documentation (including CSP). Coding with ATLAS.ti		Interviews, questionnaire, documentation (including CSP). Coding with ATLAS.ti	-

Table 4. Research philosophy, approach, strategy, choice, time horizon, technique and procedure (Saunders et al., 2007)

PART 1: EXPLORATION PHASE

3 THE LITERATURE ON LEGACY INFORMATION SYSTEMS

"The greatest part of a writer's time is spent in reading, in order to write: a man will turn over half a library to make one book".

---Johnson, S. as quoted by Boswell, J. (1791)---

3.1 Introduction

In this chapter the theory related to legacy IS is elaborated upon. First an overview is provided of existing relevant IS management literature, this includes the concepts of IS life cycle management and IS portfolio management (Section 3.2). Subsequently, a systematic literature review on the topic of this research, legacy IS and legacy IS decision making, is provided (Section 3.3). This will be used to position this research in the current literature. Based on this literature, legacy IS characteristics (Section 3.4) and legacy IS abandonment triggers (Section 3.5) are described. Then, decision making options of legacy IS are elaborated upon (Section 3.6). Subsequently, legacy IS decision making based on eclectic decision making approaches are elaborated (Section 3.7). Furthermore, methods to identify legacy IS are described (Section 3.8). Subsequently, relevant literature concerning the practical abandoning of legacy IS is presented (Section 3.9). This chapter closes with a summary and conclusions (Section 3.10).

3.2 Information system management

IS represent considerable assets in many organizations today; they can be very important and expensive assets for an organization. Like all assets, IS assets should be managed and an asset IS management plan should be available (Kyte, 2008). One of the major problems with the literature on "IS management" is the lack of common definitions of terms like: "IS management", "information management", "technology management", "strategic information management" and "IT management" (Booth & Philip, 2005). They are all used for similar concepts. IT focuses on the technical issues (application software, hardware and data sets), while IS takes a holistic viewpoint and also includes the people and procedures component (see Figure 2, p. 5). Every system consists of sub-systems and is part of a supra-system. This principle is called recursivity. In this dissertation, three "system" hierarchies are distinguished. From the highest to the lowest level, these system levels are: (1) "information function", (2) the IS and (3) the application software level (see Table 1**,** p. 5). IS can be managed on an individual basis (the life cycle is managed) and can be managed individually, as part of a portfolio.

Several authors define the concept of IS life cycle management (Berghout & Nijland, 2002; Swinkels, 1997) besides IS portfolio management (Benson, Bugnitz, & Walton, 2004; Maizlish & Handler, 2005). IS life cycle management suggests managing the life cycle of individual IS and finds its origin in product life cycle research (Lehner, 1989). IS portfolio management refers to managing a set of individual IS arranged into a portfolio of IS and finds its origin in the work of financial portfolios (Markowitz, 1952). Within life cycle management, the focus is on managing the IS components within the boundary of the IS. In analogy with Boehm's (1981) communication paths, the complexity of IS life cycle management is illustrated by calculating the number of relations of the components. The number of relations within each IS can be calculated as N*(N-1)/2, in which N is the number of components of the IS. Each individual component will have particular characteristics and each relation between components will also have particular characteristics. According to this formula, the number of relations will grow quadratically with the number of components. This implies growing complexity, which is important in understanding legacy IS abandonment decision making issues. The IS artifact includes five component types: application software, hardware, data sets, people and procedures (Brussaard & Tas, 1980). Their 5*(5-1)/2 = 10 relations are illustrated in the Figure 8.

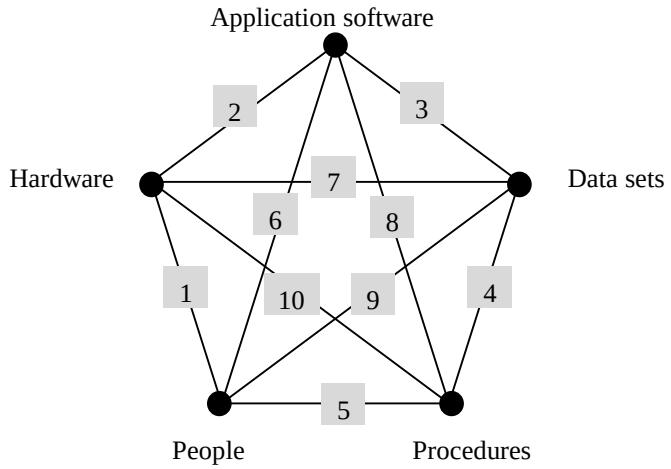


Figure 8. Ten relations between the components of the IS Pentagon

In the above Figure 8 relation number two refers to the relation between application software and the hardware. An example of this relationship is, for instance, that high quality software is running on inadequately performing hardware and as a result the entire IS might be labeled a legacy IS. In practice it will often be difficult to identify the most critical components affecting the performance of the overall system. The number of relations will grow quadratically with the number of components, according to the formula N*(N-1)/2 = ½ N^2-1/2 N. For example, the number of relations of two IS with in total N=10 components (each IS has N=5 components) and assuming that all the components have relations with each other, will already result in 45 relations (N*[N-1]/2 into 10*[10-1]/2= 45). Ten IS with five components each, will result in 50*(50-1)/2 = 1225 relations. In contemporary organizations the information functions will include hundreds or more of IS (Berghout, Nijland, & Powell, 2011). Many organizations therefore use “IS life cycle management” (Berghout & Nijland, 2002) and “IS portfolio management” (Maizlish & Handler, 2005; McFarlan, 1981; Sommerville, 2007; Ward, 1988) to control IS and their relationships. The relevance of techniques for managing legacy IS is elaborated in the subsequent two sections.

3.2.1 Life cycle management

The life cycle theory has been known for a long time as a valuable instrument in the analyses of the dynamic developments of products in the market (Lehner, 1989). IS are incepted, maintained and finally abandoned (Berghout & Nijland, 2002; Swinkels, 1997). This research follows the life cycle as described by Berghout and Nijland (2002). Berghout and Nijland (2002) describe full IS life cycle management and distinguish three major life cycle activities: planning, development and exploiting. In the exploitation stage operation and maintenance activities are being performed to ensure that the IS continues to meet business requirements. These activities are often based on best practices such as: “Application Service Library” (ASL) (Pols & Backer, 2006); “Business Information Services Library” (BISL) (Pols & Backer, 2007) and “Information Technology Infrastructure Library” (ITIL) (Office of Government Commerce, 2007) and on other research (Berghout et al., 2011; Klompé, 2003).

Looijen (1995) defines his “state model”. This model starts with information policy and information planning, followed by development, acceptation and implementation of an IS (Looijen, 1995). Then three parallel states can be distinguished: these three states are: “utilization of the IS”, “maintenance of the IS” and “exploitation of the IS” (Looijen, 1995). Although Looijen (1995) mentions the end of the life cycle of an IS as the moment when the IS is abandoned, he does not describe the abandoning state in his “state model”. This research adopts the IS life cycle terminology of Berghout and Nijland (2002) which also includes the abandoning of the IS (Figure 9).

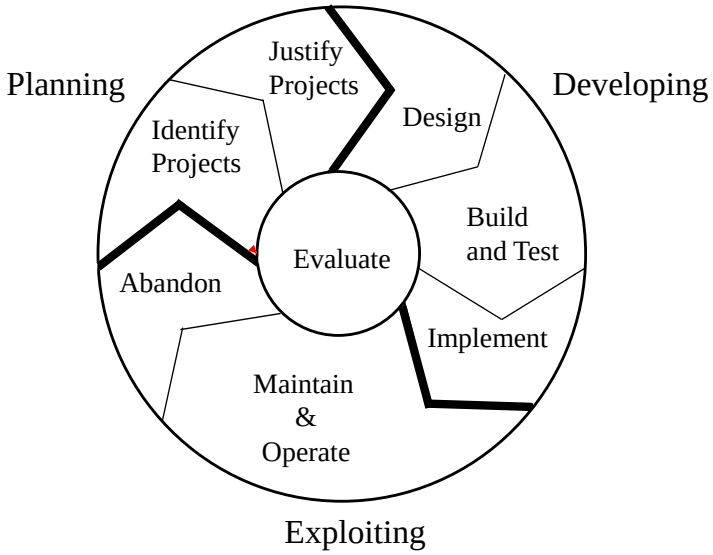


Figure 9. IS life cycle (based on Berghout and Nijland, 2002)

Another perspective of life cycle management is provided by Lehner (1989), who includes time as a component on a horizontal axis and use benefit, cost and usage on the vertical axis. Lehner (1989) distinguishes a life cycle model with four phases: "system development and introduction, growth, saturation/maturity and decline". This is illustrated in Figure 10.

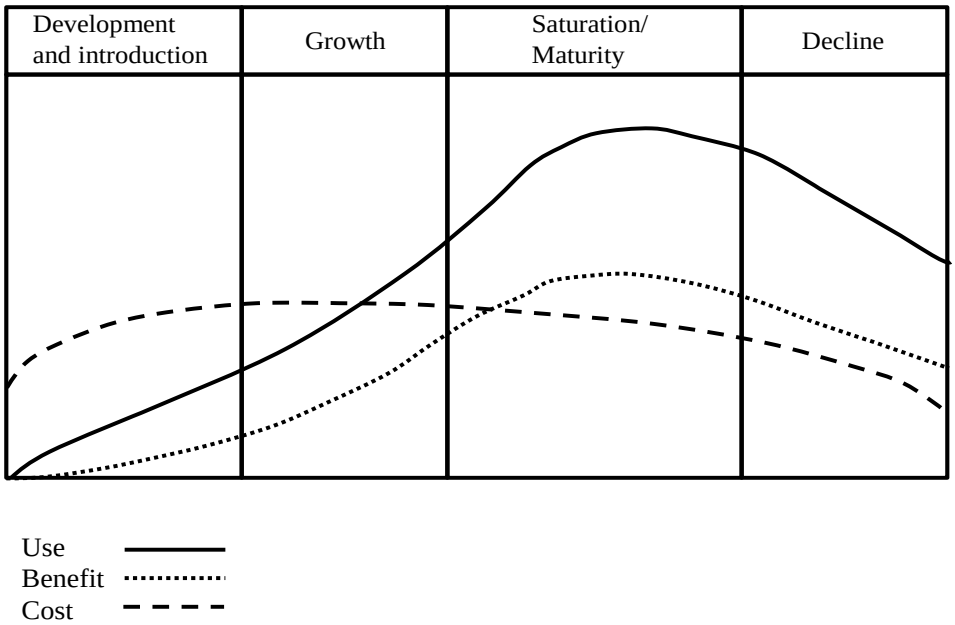


Figure 10. Software life cycle (Lehner, 1989)

Lehner (1989) does not include the abandonment of the asset in his figure. It can be argued that based on common economic principles, an asset should be abandoned if the costs are higher than the benefits minus exit costs and this phase can be added. A further enhancement on the benefits and burden of the software life cycle is provided by Maizlish and Handler (2005), who describe the value and cost components of software.

During the maintenance phase IS are subject to structural deterioration and obsolescence with age (Lehman, 1980a; Swanson & Dans, 2000). Obsolescence is defined by the Oxford Dictionary of English (2010) as: "*the process of becoming obsolete or outdated and no longer used*". When the maintenance effort activities are very expensive and time consuming due to the deterioration of an IS or when the opportunity costs are growing (a

better alternative is available) an IS is often referred to as: “legacy IS”. A legacy IS is therefore a subset of an IS with specific characteristics and will ultimately be subject to retirement studies. Swanson and Dans (2000) suggest that management should equilibrate its allocation of system maintenance effort with its estimate of a systems remaining life. An increase in the maintenance effort may extend the system’s estimated remaining life. However, the estimate of a shortened remaining life may also call for a reduction on maintenance effort. Because the two variables are related, management should consider each with respect to its decision or estimate of the other (Swanson & Dans, 2000). When a decision to retire a legacy IS has been made, the system will move to the final activity in the life cycle, being the practical abandoning activity (also referred to as disposal phase, Rittinghouse, 2004; White, 2007). The decision to abandon a legacy IS and the practical abandoning of a legacy IS, is the final management activity in the life cycle of an IS and should ensure the orderly termination of the system (Rittinghouse, 2004).

Decision making as a process is described in Section 1.1.3 (p. 7). Decision making applies to all phases within the life cycle of an IS. Research on organizational decision making related to the abandonment of legacy IS is scant (Furneaux & Wade, 2011; Sakthivel, 1994; Sellars, 2004; Swanson & Dans, 2000). Suggestions as to when to abandon legacy IS are made by Zvegintzov (1984): *(1) it is no longer needed, (2) it no longer runs on its hardware; (3) its hardware is replaced; (4) it is not adapted to changing real-world conditions; (5) superior alternative software is developed or is available for purchase.* Berghout (1997) included Harrison’s decision making (Harrison, 1999) into the IS life cycle, suggesting continuous evaluation during the life cycle; Berghout (1997) did not include the abandonment phase. Based on the IS life cycle (Berghout & Nijland, 2002; Lehner, 1989) the continuous evaluation during the abandonment phase is added. Figure 11 illustrates the decision making process (Harrison, 1999) including continuous evaluation (Berghout, 1997).

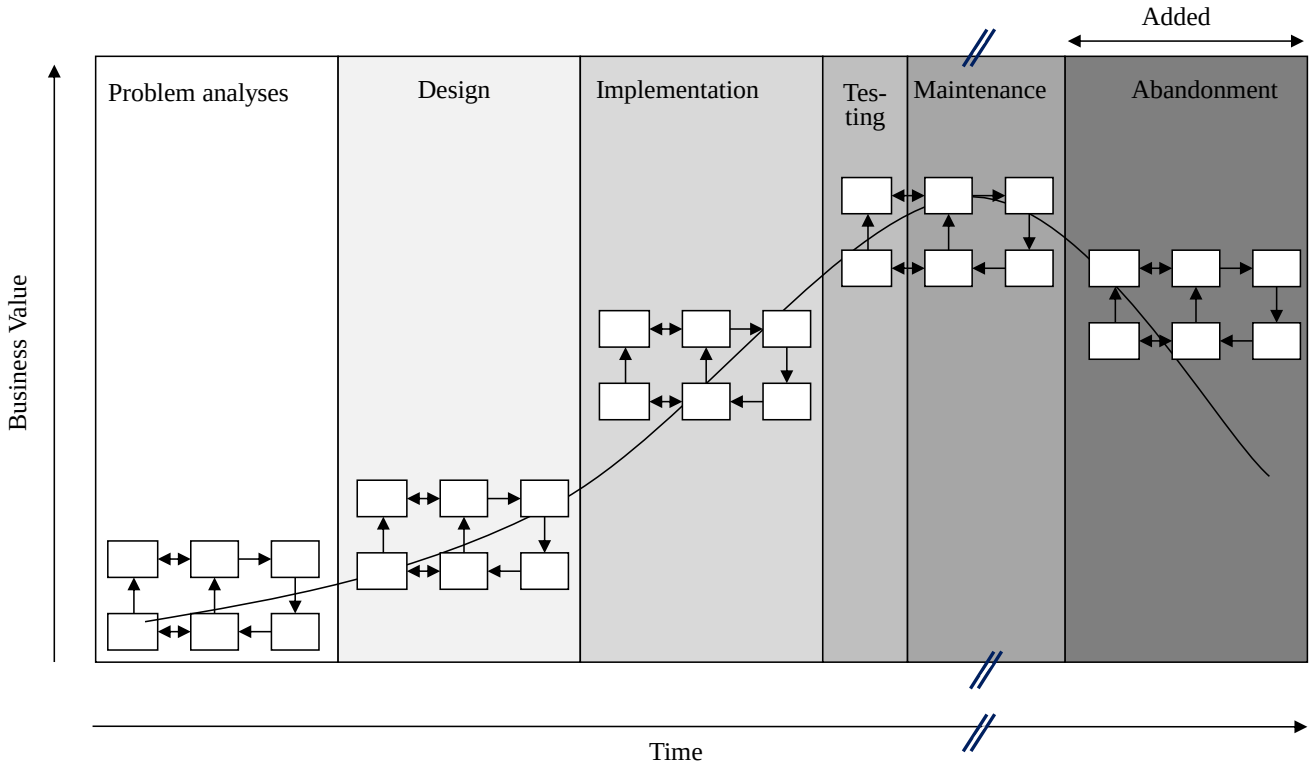


Figure 11. Decision making process and continuous evaluation of an IS during the complete life cycle of an IS (based on the IS life cycle Berghout and Nijland, 2002; Lehner, 1989) see Figure 3 (p. 8) for the contents of the boxes

Abandoning a legacy IS often requires that an alternative exists (Comella-Dorda et al., 2000). In this dissertation abandoning a legacy IS is defined as a process of removing a current constellation and combination of software, hardware, data sets, people and procedures.

When an IS is managed according to a life cycle approach, a life cycle plan should be available (Kyte, 2008). It might be argued that the life cycle approach of individual IS could lead to abundant details of the underlying components of the IS and their relationships. These components, again, should all be optimized to ensure maximum business value for the organization. There is also literature regarding the individual components of the IS, such as application life cycle management, hardware life cycle management and data life cycle management. Regarding people there is personnel development planning (e.g. staff education, career paths, retirement). Regarding procedures there is the renewal, maintaining or abandoning of procedures.

3.2.2 Portfolio management

IS portfolio management draws on the ideas of Nobel laureate Markowitz (1952). His work on financial portfolios has been applied in many other business functions, for example “product portfolio management” (Henderson, 1970). IS portfolio management was identified by McFarlan in 1981, taking a holistic view and analyzing the effects applications have on organizations. Decker (2008) defines portfolio management as a governance process that strives to treat applications as assets whose value, cost and risk should be thoughtfully managed over time. Portfolio refers to the notion that IS are collectively managed as an interrelated set of assets. Often portfolios are presented into four quadrants (Maizlish & Handler, 2005; McFarlan, 1981; Sommerville, 2007; Van den Heuvel, 2007; Ward, 1988; Warren, 1999), where the X-axis represents the technical quality and the Y-axis represents business value.

The assessment of legacy IS is described by Warren (1999). He distinguishes a technical viewpoint in which application software, support software and hardware attributes of legacy IS are scored versus business value attributes (including business importance and expected lifetime) of the legacy IS and plots them on a graph with two orthogonal dimensions: business value and technical value. The business value and technical value are calculated as the weighted means of quantifiable product or process metrics (Van den Heuvel, 2007). Sommerville (2007) suggest also a portfolio approach, by assessing all the legacy systems on system quality and business value and plotting them into a chart. Management can decide which strategy is appropriate for the system. The system with high quality and high business value should continue in operation using normal system maintenance (Sommerville, 2007). The legacy IS with low quality and low business value should be abandoned (Sommerville, 2007). The IS with low quality and high business value make an important business contribution, but are expensive to maintain and should be re-engineered or replaced if a suitable system is available (Sommerville, 2007). In case re-engineering is not viable, then the legacy IS should be abandoned and replaced by a new IS (Warren, 1999). High quality, low business value should be replaced by “Commercial off-the-shelf” (COTS), scrapped completely or maintained. Examples how to assess the business value and the technical quality are provided by Sommerville (2007) and Warren (1999). A portfolio diagram is illustrated in Figure 12.

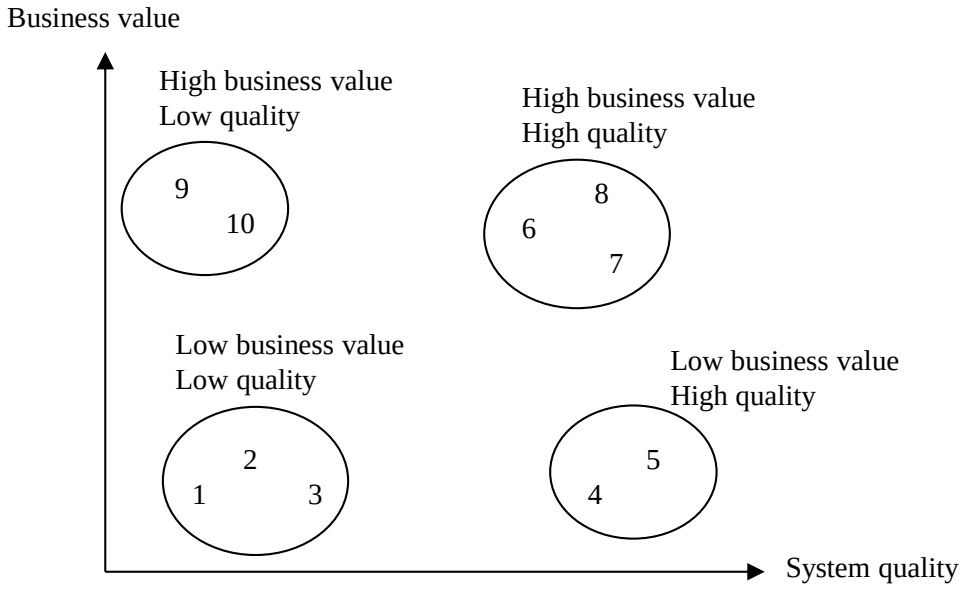


Figure 12. Assessment of 10 systems (scored on business value and system quality) (Sommerville, 2007)

The power of portfolio management is, rather than examining an individual case, enabling management to holistically look at the entire portfolio and identify the set of resources that are poorest in performing or weakest in quality (Benson et al., 2004). IS portfolio management particularly compares all high level characteristics of the IS within information functions, in contrast to IS life cycle management where the relationships between the components of the individual IS are optimized. In general it can be argued that an IS portfolio approach tries to be more comparative amongst IS at the expensive of giving up life cycle details of each individual IS.

A finance director or board of directors are often more interested in this high-level perspective of portfolio approaches. By examining the entire set of IS, management can determine which legacy IS should be abandoned or replaced (Benson et al., 2004). Stakeholders closer to the individual IS might be more interested in a life cycle view, which examines individual IS in-depth. Similar to IS life cycle management, IS portfolio management can also often be applied to individual components of an IS e.g.: "application portfolio management" or "hardware portfolio management" (Benson et al., 2004; Maizlish & Handler, 2005). Van den Heuvel (2007) notices that, although portfolio management is an effective tool for decision making by management, the method suffers from serious shortcomings. For example estimating the business value and technical value is a herculean task because of a lack of hard quantifiable and validated data and metrics (Van den Heuvel, 2007).

3.3 Systematic literature review on legacy information system decision making

In this section, a systematic literature review of legacy IS decision making is presented. It follows the methodology which is described by Kitchenham (2004); Kitchenham and Charters (2007) and Kitchenham et al. (2009; 2010). The eleven most important IS journals have been reviewed and this selection is based on the weighted average ranking of IS journals published by the AIS[1]. The list of journals include: (1) *Management Information Systems Quarterly* (MISQ), (2) *Information Systems Research* (ISR), (3) *Communication of the Association for Computing Machinery* (CACM), (4) *Journal of Management Information Systems* (JMIS), (5) *European Journal of Information Systems* (EJIS), (6) *Information and Management* (I&M), (7) *Communications of the Association for Information Systems* (CAIS), (8) *Journal of the Association for IS* (JAIS), (9) *Information System Journal* (ISJ), (10) *Journal of Strategic Information Systems* (JSIS) and (11) *Journal of Information Technology* (JIT). The selection criteria for incorporating a paper in this literature review are: the paper should be written in

[1] The Association for Information Systems (AIS), source: (http://ais.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=432), referenced, May, 2013.

English (restriction) and is cited in at last five other peer-reviewed publications. Google Scholar was used to check these citations. Relevant books or chapters from books are listed separately. Conference papers and working papers are excluded in this literature review.

The following 26 keywords have been applied: “Legacy information system”, “Legacy information systems”, “Legacy IS”, “Legacy system”, ”Legacy systems”, “Legacy application”, “Legacy applications, “Legacy application system”, “legacy application systems”, “legacy IT”, “legacy information technology”, “legacy ICT”, “legacy information communication technology”, “legacy information system decision making, “legacy information systems decision making”, “legacy IS decision making”, “legacy system decision making”, ”legacy systems decision making”, “legacy application decision making”, “legacy applications decision making”, “legacy application system decision making”, “legacy application systems decision making”, “legacy IT decision making”, “legacy Information technology decision making”, “legacy ICT decision making”, “legacy information communication technology decision making”. Journals are searched using the most appropriate search engine, which resulted into the following selection of journals and search engines/databases: (1) MISQ via search engine/database “AISel”, (2) ISR via: “Informs”, (3) CACM via: “ACM” (Digital Lib), (4) JMIS via: “Ebsco host”, (5) EJIS via: “ProQuest” (ABI/Informs), (6) I&M via: “Elsevier” (Science direct), (7) CAIS via: “AISel”, (8) JAIS via: “AISel”, (9) ISJ via: “Wiley”, (10) JSIS via: “Elsevier” (Science direct), (11) JIT via: “ProQuest” (ABI/Informs). Google Scholar and Google[2] also have been used as search engines for above 26 keywords. This led to a massive 157,367, respectively, 11,521,386 hits, which indicate the massive societal interest in legacy IS and abandonment decision making. Table 3 provides an overview of the outcome of all searches. The number in Table 3 indicates the number of identified papers in the eleven journals. The date of this search was May 2013 and the resulting hits are illustrated in Table 5.

[2] Google, search engine via www.google.com/ncr (ncr stands for: "No Country Redirect")

Keywords	Journals											Search engine/database		
	(1) MISQ	(2) ISR	(3) CACM	(4) JMIS	(5) EJIS	(6) I&M	(7) CAIS	(8) JAIS	(9) ISJ	(10) JSIS	(11) JIT	AISel	Google Scholar	Google. com
Legacy information system	0	0	1	0	0	6	0	0	4	2	0	9	626	185,000
Legacy information systems	0	1	4	1	0	6	13	2	4	2	5	59	2,550	350,000
Legacy IS	1	1	6	66	0	10	4	0	0	2	0	24	36,000	4,000,000
Legacy system	2	1	25	11	0	130	23	3	36	38	1	234	23,700	1,330,000
Legacy systems	21	13	110	34	3	130	85	13	36	38	3	802	54,400	2,080,000
Legacy application	2	1	10	0	0	23	4	1	1	5	0	28	8,800	336,000
Legacy applications	3	3	29	4	2	23	20	2	1	5	0	110	18,700	904,000
Legacy application System	0	0	0	0	0	0	0	0	0	0	0	0	49	330,000
Legacy application Systems	0	0	0	0	0	0	0	0	0	0	0	5	173	507,000
Legacy IT	1	1	2	4	0	2	3	2	2	3	0	32	12,200	702,000
Legacy information technology	0	0	0	0	0	0	2	0	0	0	0	2	87	794,000
Legacy ICT	0	0	0	0	0	0	0	1	0	0	0	2	81	3,380
Legacy information communication Technology	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Legacy information system decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Legacy information systems decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Legacy IS decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Legacy system decision making	0	0	0	0	0	0	0	0	0	0	0	0	1	2
Legacy systems decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Legacy application decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Legacy applications decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Legacy application system decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Legacy application systems decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Legacy IT decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Legacy information technology decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Legacy ICT decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Legacy information communication technology decision making	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total keyword hits (including double publications)	30	21	187	120	5	330	154	24	84	95	9	1,307	157,367	11,521,386
Total keyword hits within the 11 journals	1,059													

Table 5. Results and number of hits on 26 keywords in 11 journals

In total there were 1,059 hits within these eleven selected journals. First, double hits were removed. Subsequently, all journal publications were scanned on relevant content. However, many papers only mentioned the word "legacy IS" somewhere in the body text, but were not dedicated to this topic in any way and were, therefore, not relevant for this literature review.

Only five papers within the eleven selected journals included studies on legacy IS decision making. These are, Alderson and Shah (1999); Furneaux and Wade (2011); Kelly et al. (1999); Krishnan, Mukhopadhyay, & Kriebel (2004) and Swanson and Dans (2000). The references of these papers have been further scanned "downward" (1 tier). This resulted into another six studies. After the identification of the literature, the literature was analyzed (see the following sections). Table 6 provides an overview of the eleven studies on legacy IS. A distinction could be made between papers that take a more holistic approach to legacy IS and papers that take

a more technical approach to legacy IS. More holistic studies, such as Alderson and Shah (1999), include the business purposes of the legacy IS. More technical studies, such as Krishnan et al. (2004), restrict themselves to studying effects on software, hardware and data. Table 6 provides an overview of relevant papers. It also indicates whether the study takes a holistic viewpoint (marked as "IS") or is restricted to technical issues (marked as "IT").

Name	IT/IS artifact	Data set	Result
Swanson and Dans (2000)	IT/IS	Data from 758 systems among 54 organizations is used.	This paper examines system maintenance and prospective replacement in new theoretical terms, positing that managers "equilibrate" (balance) their allocation of maintenance effort with their expectations of a systems remaining life. How managers ultimately make the retirement and replacement decision remains unanswered.
Furneaux and Wade (2011)	IS	Two tier survey, first tier 33 interviewees, second tier 172 interviewees.	A multi-method empirical study was undertaken to improve understanding of organizational level information system discontinuance. Research commenced with the development of a broad theoretical framework consistent with the technology–organization–environment (TOE) paradigm. Results suggest that system capability shortcomings, limited availability of system support, and low levels of technical integration were key determinants of increased intentions to replace an existing system.
Krishnan et al. (2004)	IT	Not specified.	A stochastic software decision making model for maintenance of software, describing three maintenance options, major upgrade (replacement), minor upgrade or no upgrade.
Alderson and Shah (1999)	IS	Not specified.	Four different viewpoints on legacy IS are suggested and elaborated: developmental, operational, organizational and strategic.
Kelly et al. (1999)	IT	15 Cases.	Unstable business environment (which are characterized by a high rate of change) experience more difficulties to adapt because of legacy IT compared to more stable environments resulting in a gap (being misalignment) between IT and strategic vision.
Richmond, Nelson and Misra (2006)	IT	Five cases, data of 181 systems were captured.	A normative decision planning model concerning the system lifetime which is tested.
Adolph (1996)	IT	Not specified.	Case study on abandoning a legacy IS.
Bennet (1995)	IS	Not specified.	The legacy dilemma is explained and articles of dealing with legacy IS are provided.
Sakthivel (1994)	IT	Not specified.	An economical decision model for deciding between maintaining existing software or replacing it with redeveloped software.
Chan, Chung and Ho (1996)	IT	Not specified.	Theoretical model describing when to replace software.
Zvegintzov (1984)	IT/IS	Not specified.	Describes five reasons for retiring and replacing software: it is no longer needed, it no longer runs on the hardware, its hardware is replaced, it is not adapted to changing real world conditions or superior alternatives are available.

Table 6. Overview of papers on legacy IS decision making

Besides scholarly reviewed journal publications, the literature research also identified six scholarly books on legacy IS (Brodie & Stonebraker, 1995; Seacord et al., 2003; Ulrich, 2002; Ulrich & Newcomb, 2010; Van den Heuvel, 2007; Warren, 1999). One article was found in a journal that was outside the initial set of journals upfront (Brooke & Ramage, 2001). Furthermore, two dissertations are worth mentioning, Furneaux (2009) and Sellars (2004). A number of more general books include chapters on legacy IS (Brooke, 2002; Larsen, 1998; Maizlish & Handler, 2005; Sommerville, 2007).

In summary, this literature review identified the following literature dedicated to legacy IS: twelve journal papers, six books and two dissertations. The low number of scientific papers (twelve) of which seven had been published before 2000 indicates that there is not much research published on this topic. Within this literature there is a major difference between research focused on IS in organizations and research limited to technological aspect of IS (IT).

The following sections elaborate upon: the characteristics of legacy IS, abandonment triggers to abandon legacy IS, decision making options of legacy IS, eclectic decision making approaches to abandon legacy IS, the identification of legacy IS and, finally, the practical abandoning of legacy IS.

3.4 Characteristics of legacy information systems

Berghout and Nijland (2002) articulate an IS life cycle approach for IS management (see Figure 9, p. 23). In the life cycle theory, the management objectives differ per IS life cycle stage (Berghout & Nijland, 2002; Berghout et al., 2011). Throughout the years, in the operations and maintenance activity, the IS will increasingly resist meeting the requirements of the organization. This increasing resistance is referred to as "aging" of the IS (Lehman, 1980a; Swanson & Dans, 2000). Finally, the IS becomes a legacy IS and is characterized by IS legacy characteristics. In order to increase the understanding of legacy IS and their specific characteristics, this section investigates characteristics of legacy IS. It provides an overview of all legacy characteristics that are mentioned in the literature. First characteristics of legacy IS are provided and subsequently all individual parts of the IS (application, hardware, data sets, people and procedures) are described (Brussaard & Tas, 1980). The systematic literature review from Section 3.3 (p. 26) is used as starting point. Furthermore, references from this literature are included and, subsequently, their references. This search was continued until additional searches did not provide additional legacy characteristics. For each legacy IS characteristic the most prominent authors are included as reference.

The distinction between individual components of a legacy IS and the legacy IS as an entity is not always well-defined. For instance, a paper on application software, also mentions hardware and then refers to inadequate documentation of the "system". Then it is unclear if the inadequate documentation is related to the application software only, or the combination of hardware and software. Another example is research that states that applications are expensive, but does not address whether the application is expensive to maintain, or whether there are opportunity costs (there is a better and cheaper alternative available), or whether there are high license costs.

The aim of this research was to identify as many legacy IS characteristics at the most elementary level from the literature as possible; relative frequency or relative importance did not play a role. Due to the fact that a lot of legacy IS characteristics were found in the literature, resulting in pages of unreadable tables with legacy characteristics and their reference, these characteristics of IS and the individual components of IS (application, hardware, data sets, people and procedures) are described in Appendix C (p. 185). An overview of the legacy characteristics of IS and the individual components (application, hardware, data sets, people and procedures; Brussaard & Tas, 1980) are subsequently described in boxes. Box 1 provides an overview of IS-related legacy characteristics, Box 2 provides an overview of application-related legacy characteristics, Box 3 provides an overview of hardware-related legacy characteristics, Box 4 provides an overview of data-related legacy characteristics, Box 5 provides an overview of the people-related legacy characteristics and Box 6 provides an overview of procedure-related legacy characteristics.

Box 1 provides an overview of IS-related legacy characteristics (in Appendix C.1, Table 52 [p. 186] 25 different legacy characteristics from an IS perspective as found in the literature are described).

Organizations have an architecture including architecture principles. Architecture principles define the underlying general rules and guidelines for the use and deployment of all IT resources and assets across the enterprise and are chosen to ensure alignment of IT strategies with business strategies and visions. Legacy IS are characterized by a limited architectural fit (Ulrich, 2002) or aging legacy architectures (Alderson & Shah, 1999; Ulrich, 2002). Legacy IS are designed in stove pipe fashion or silos (Daga et al., 2005; Kelly et al., 1999; Ulrich, 2002; Van den Heuvel, 2007). Legacy IS may hinder progress and are not agile (Daga et al., 2005) or are a barrier to strategic innovation (Kelly et al., 1999). Legacy IS are no longer meeting the needs of the environment (Brooke, 2002) or are significantly resisting modification and evolution in order to meet new and constantly changing business requirements (Alderson & Shah, 1999; Brodie & Stonebraker, 1995; Kelly et al., 1999). Legacy IS do have increasing security issues (Alderson & Shah, 1999).

Legacy IS are characterized by inadequate documentation of the system (Adolph, 1996; Warren, 1999) or the documentation becomes increasingly inaccurate thereby making future changes even more difficult (Parnas, 1994). The understanding of legacy IS details is often lacking (Bisbal et al., 1999; Warren, 1999) and system skills are hard to find (Alderson & Shah, 1999; Kelly et al., 1999). Legacy IS have diminished support from the supplier (Furneaux & Wade, 2011) and are difficult if not impossible to extend (Bisbal et al., 1999). Legacy IS have increasing complexity (Kelly et al., 1999; Richmond et al., 2006; Ulrich, 2002). Legacy IS have declining system reliability (Furneaux and Wade, 2011) or inadequate performance (Alderson & Shah, 1999). Legacy IS have inadequate interoperability, ambiguous interfaces prohibits integration with other systems (Alderson & Shah, 1999; Bisbal et al., 1999; Brodie & Stonebraker, 1995). Legacy IS have a lack of integration of systems, a dispersed landscape (Kelly et al., 1999).

Legacy IS prevent organizations from fully exploiting computers and the value they bring in organizations (Ulrich, 2002), Alderson and Shah (1999) notice that the benefits of legacy IS are lower than costs. Legacy portfolios are 30-50% more expensive to run than their packaged systems or newer technology (Hunter & Aron, 2006). Legacy IS are expensive to maintain and operate (Brodie & Stonebraker, 1995; Warren, 1999), they have excessive support costs (Furneaux and Wade, 2011).

Box 1. Overview of IS-related legacy characteristics.

Apart from the legacy characteristics from an IS perspective, in the legacy IS literature also legacy characteristics of the individual components of the IS were found. Box 2 provides an overview of application-related legacy characteristics (in Appendix C.1, Table 53, [p. 188] 25 legacy characteristics from an application perspective as found in the literature are described).

Application software is maintained during the years, during the maintenance the application software will structurally deteriorate. Legacy application software is characterized by inadequate software quality, this can happen for example that, due to time pressure new functionality is added and the software is not programmed in a structured way (old code is not removed by a programmer, resulting in dead code, in the software and does not support any operational functionality anymore, or there is duplicated code which is the result of "quick and dirty" programming). As a result of gradually deteriorating structure its performance often degrades and often becomes "buggy" because of errors introduced when changes are made. Legacy application software was written in the past when the main design considerations were memory optimization and processing speed instead of program structure (Adolph, 1996; Bennet, 1995; Brodie & Stonebraker, 1995; Chan et al., 1996; Kelly et al., 1999; Krishnan et al., 2004; Lehman, 1980a; Lehman, 1980b; Lehman & Ramil, 2001; Parnas, 1994; Richmond et al., 2006; Sakthivel, 1994; Seacord et al., 2003; Swanson & Dans, 2000; Van den Heuvel, 2007).

Legacy application software has increasing complexity of the software (Chan et al., 1996; Krishnan et al., 2004; Lehman, 1980a; Lehman, 1980b; Seacord et al., 2003; Warren, 1999) and reduced or inadequate performance (e.g. due to growing application size and inadequate design) (Brodie & Stonebraker, 1995; Krishnan et al., 2004; Parnas, 1994). Legacy application software has decreasing reliability due to errors which are introduced during the application maintenance (Krishnan et al., 2004; Parnas, 1994). Legacy application software is characterized by a lack of maintenance of the application (Swanson & Dans, 2000) and use of older program languages, or availability of newer program languages (technical opportunities, such as higher generation languages or object oriented methods) (Adolph, 1996; Bennet, 1995; Brodie & Stonebraker, 1995; Chan et al., 1996; Krishnan et al., 2004; Richmond et al., 2006; Zvegintzov, 1984).

Legacy supporting software is characterized by compilers which are no longer supported anymore by hardware suppliers (Sommerville, 2007; Warren, 1999) or support from application supplier has ended (Furneaux & Wade, 2011) or there is no support from hardware, the platform on which the application is running (Adolph, 1996; Parnas, 1994; Zvegintzov, 1984). The software does not have an architectural fit (Richmond et al., 2006) and there is limited agility (changes take longer) (Parnas, 1994).

Legacy IS could have a "dying" application program language, which is hardly taught at universities anymore (Computerworld, 8th April, 2013). It is characterized by skills which are hard to find, staff with knowledge skills is retired or has left the organization and taken a substantial amount of knowledge with them. The application is not well-documented; new programmers do not have this knowledge and up-to-date documentation of the application is usually lacking (Alderson & Shah, 1999; Adolph, 1996; Brodie & Stonebraker, 1995; Daga et al., 2005; Parnas, 1994; Seacord et al., 2003). There is inadequate, or limited or missing documentation of the application (Bennet, 1995; Bisbal et al., 1999; Brodie & Stonebraker, 1995; Daga et al., 2005; Parnas, 1994; Zvegintzov, 1984).

The software is embedded in the hardware (Adolph, 1996; Ulrich, 2002) or there is embedded business logic (Warren, 1999). Understanding of system details is often lacking (Bisbal et al., 1999; Daga et al., 2005; Parnas, 1994) or tracing faults within applications is costly and time consuming (Bisbal et al., 1999).

Legacy application software is characterized by functionality that does not meet the requirements (Bennet, 1995; Krishnan et al., 2004; Lehman, 1996; Parnas, 1994; Van den Heuvel, 2007; Zvegintzov, 1984) or changing user perception (Lehman, 1996; Parnas, 1994; Seacord et al., 2003). There are alternatives or opportunity cost (a better or cheaper IS available) (Sakthivel, 1994; Zvegintzov, 1984), it is expensive to maintain (Bisbal et al., 1999; Bennet, 1995; Brodie & Stonebraker, 1995; Chan et al., 1996; Daga et al., 2005; Krishnan et al., 2004; Parnas, 1994; Sommerville, 2007; Swanson & Dans, 2000) or there are increasing maintenance costs of applications (Chan et al., 1996; Lehman, 1980b; Lehman, 1980a; Richmond et al., 2006; Sakthivel, 1994; Seacord et al., 2003).

Box 2. Overview of application-related legacy characteristics.

Box 3 provides an overview of hardware-related legacy characteristics (in Appendix C.1, Table 54 [p. 189] 10 legacy characteristics from a hardware perspective as found in the literature are described).

Computer hardware has become infinitely more powerful through the years, resulting in an exponential growth of computing performance. Aging legacy hardware, however, is characterized by slow or inadequate performance (Adolph, 1996; Bisbal et al., 1999; Daga et al., 2005; Kelly et al., 1999; Parnas, 1994). Due to numerous technical improvements the hardware reliability has increased. Aging legacy hardware, however, is characterized by poor reliability (Parnas, 1994).

Legacy hardware is characterized by an old hardware type (Brodie & Stonebraker, 1995; Sommerville, 2007; Warren, 1999) or an old platform type (Adolph, 1996; Parnas, 1994) and there is typically a lack of knowledge of old hardware (Alderson & Shah, 1999). Legacy IS run on obsolete hardware (Bisbal et al., 1999; Daga et al., 2005), the hardware supplier discontinues the product line (Adolph, 1996). The hardware is not compatible with current procurement policies (Sommerville, 2007) or expensive to maintain (Bisbal et al., 1999; Brodie & Stonebraker, 1995; Daga et al., 2005 Sommerville, 2007) or legacy hardware is embedded with software (Adolph, 1996).

Box 3. Overview of hardware-related legacy characteristics

Box 4 provides an overview of data-related legacy characteristics (in Appendix C.1, Table 55 [p. 190] eight legacy characteristics from a data perspective as found in the literature are described).

Organizations are amassing and storing more data than ever before, the amount of data in our world has been exploding. Legacy data characteristics are an immense volume of data which has accumulated over the lifetime of the system, this data may be inconsistent and may be duplicated in several files (Sommerville, 2007), consists of aging data architecture; aging databases (Adolph, 1996; Brodie & Stonebraker, 1995; Ulrich, 2002). It might be defined and stored redundantly across multiple stovepipe business units and applications (Ulrich, 2002).

The same or similar data which is defined inconsistently across multiple systems or the same data terminology which is used to define different data across multiple applications and business units (Ulrich, 2002). The integrity of the data may be poor and contain information it should not contain (Ulrich, 2002).

Data which may not be easily accessible by modern systems or through user-based inquiries and data which cannot be readily shared across systems, business units and organizational boundaries (Ulrich, 2002).

Box 4. Overview of data-related legacy characteristics

IS professionals are part of the IS, as other components of the IS they can gain legacy characteristics, named legacy people characteristics. Box 5 provides an overview of the people-related legacy characteristics (in Appendix C.1, Table 57 [p. 191] four legacy characteristics from a people perspective as found in the literature are described).

People-related legacy characteristics are, skills which are hard to find, retiring staff, or staff lacking technical expertise (Adolph, 1996; Daga et al., 2005; Warren, 1999). Many people who are doing software development do not have education appropriate to the job (Parnas, 1994).

Changes made by people who do not understand the original design concept almost always cause the structure of a programme to degrade; after many such changes the original designers no longer understand the modified product Parnas (1994).

A growing dislike of the tedious work on legacy IS, *"if something is difficult, tedious and slow people will try to avoid doing it"* (Adolph, 1996). Engineers prefer working on new system development instead of maintaining old software (Bennet, 1995; Ulrich, 2002).

Box 5. Overview of the people-related legacy characteristics

The following Box 6 provides an overview of the procedure-related legacy characteristics (in Appendix C.1, Table 56, [p. 190] two legacy characteristics from a procedure perspective as found in the literature are described).

Procedure-related legacy characteristics are: do not take advantage of streamlined facilities and standard routines; use programming and design practices that current management would not permit (Zvegintzov, 1984).

Business policy and rules, definitions of how the business should be carried out and constraints on the business. Use of the legacy application system may be embedded in these policies and rules (Sommerville, 2007).

Box 6. Procedure-related legacy characteristics

Although the research on legacy IS aging is limited, it is possible to identify many legacy IS aging characteristics. In this section 74 legacy characteristics have been derived from the literature. Legacy characteristics may concern the overall IS and all of its components (application software, hardware, data sets, people and procedures) (Brussaard & Tas, 1980). There is hardly any literature about the relative importance of these legacy IS aging characteristics, making this area relatively unattended.

3.5 Abandonment triggers of legacy information systems

During operation, the IS will age and at some point in time will be referred to as legacy IS. The previous section identified 74 legacy IS characteristics. At some point in time there is a "trigger" that makes organizations abandon legacy IS. The following eleven legacy IS abandonment triggers (categorized into IS, or its constituents) could be derived from the literature (see Table 7).

Trigger to abandon	Remarks	Source	IS	Applications	Hardware	Data sets	People	Procedures
High IS cost	High legacy IS cost, lower costs alternative possible.	Chan et al. (1996); Gode, Barua, & Mukhopadhyay (1990); Richmond et al. (2006); Sakthivel (1994)	X					
Aged/old hardware	It no longer runs on its (aging) hardware, taxing the limits of the legacy systems host computer.	Adolph (1996); Furneaux and Wade (2011); Zvegintzov (1984)			X			
Inadequate software quality	Due to ongoing software maintenance, the software structure deteriorates and maintenance becomes problematic.	Adolph (1996); Chan et al. (1996); Furneaux and Wade (2011); Kelly et al. (1999); Sakthivel (1994); Swanson and Dans (2000)		X				
New technological opportunities	E.g. superior hardware and software alternatives are available.	Chan et al. (1996); Furneaux and Wade (2011); Sakthivel (1994); Zvegintzov (1984)		X	X			
Limited or ended supplier support	Limited or ended supplier support (software and hardware).	Adolph (1996); Furneaux and Wade (2011); Kelly et al. (1999); Zvegintzov (1984)		X	X			
Redundant IS	Redundant systems due to a merger.	Kelly et al. (1999)	X					
Limited or no technical skills available	Reduced availability of individuals with the required expertise; technical skills not available anymore.	Furneaux and Wade (2011); Kelly et al. (1999)					X	
Infrastructure overhaul	Partial or complete disassembly of the infrastructure.	Swanson and Dans (2000)			X			
Application functionality no longer needed	When the functionality is no longer needed the IS can be abandoned.	Zvegintzov (1984)		X				
Capability shortcomings	When an IS is not or only partly capable of delivering the capability as is required by an organization, there are system capability shortcomings. When the gap between business needed IS capability and supplying capability is growing and is too large, then the IS is candidate for abandoning.	Comella-Dorda et al. (2000); Furneaux and Wade (2011); Kelly et al. (1999); Larsen (1998); Zvegintzov (1984)	X					
Complexity	Low-levels of technical integration (low complexity) were key determinants of increased intentions to replace an existing system.	Furneaux and Wade (2011)		X	X			

Table 7. Legacy IS abandonment triggers

In the following sections decision making and identification related to legacy IS are further elaborated.

3.6 Decision making options of legacy information systems

During maintenance and operation, there is a decision to continue or discontinue the legacy IS (see Figure 9, p. 23; Berghout & Nijland, 2002; Sommerville, 2007). A decision to continue the legacy IS means continuing with regular legacy IS maintenance in which functionality is added or re-engineering of the legacy IS in which the structure of the system is improved to make future maintenance easier and cheaper. Re-engineering can mitigate significant risks associated with replacement and continued maintenance (Warren, 1999). A decision to abandon a legacy IS may result in the abandoning and scrapping of the legacy IS, referred to as decommissioning by Warren (1999), or the decision to abandon and replace all or parts of the legacy IS with a new system developed from scratch (in case continued maintenance or re-engineering is not feasible anymore) (Warren, 1999). These decision making options of legacy IS are illustrated in Figure 13.

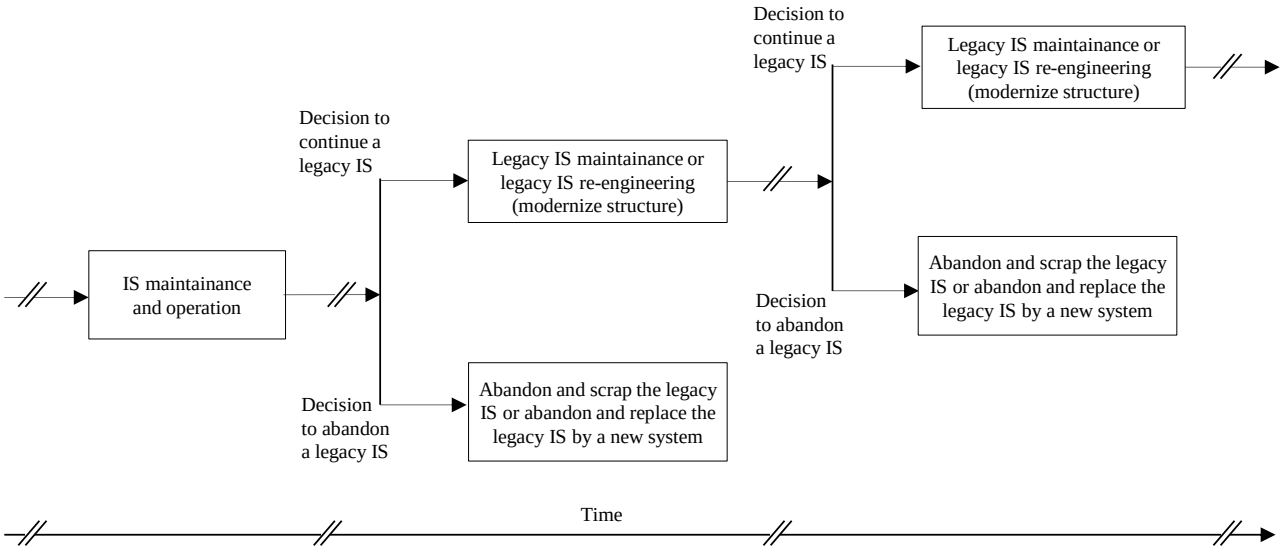


Figure 13. Decision making options with legacy IS during its life

This dissertation focuses on the decision to abandon a legacy IS. An interesting viewpoint is described by Sellars (2004): as opposed to a decision to abandon a legacy IS, he proposed a model to determine appropriate life extension strategies of legacy IS based on certain legacy IS characteristics. Sellars (2004) identified the following methods: *"re-engineering, new build, design recovery, wrapping, ignore problem, functional discovery"* and he has defined the accompanying IS legacy characteristics categorized in: "*Cost/Benefit, documentation, changes, maintenance, organization issues, people issues, system attributes and schedule".*

Warren (1999) describes a method for system evolution (re-engineering). His stance is to create an evolutionary system, capable of accommodating changes over an extended operational lifetime. Van den Heuvel (2007) distinguishes within the evolution of a legacy IS "*maintenance, modernization, replacement and phase out*". Within modernization he distinguishes black box or white box modernization. Black box modernization means that (parts of) the legacy IS are encapsulated and are subsequently integrated with new applications or components (e.g. wrapping). Knowledge about the legacy IS is focused on external service. This approach is called a non-invasive strategy (Van den Heuvel, 2007). White box modernization requires detailed knowledge of the legacy IS to allow legacy code and data to be converted. This approach is called an invasive strategy (Van den Heuvel, 2007). The replacement of existing systems with a new system, developed from scratch, might be necessary where re-engineering is not viable technically or for other reasons such as organizational policy or radical change of the business process (Warren, 1999). Replacement, according to Van den Heuvel (2007), seems to be attractive for management. However, practice has shown that this strategy bears large risks and many unpredictable pitfalls; it is costly and complex and data code conversion has to made in order to save past investments in legacy systems. It is difficult to avoid expensive downtime of the existing enterprise application and lastly it is usually not possible to guarantee that the new system will outperform the existing application in terms of both functionality and extra functional properties such as security and robustness.

Cold Turkey implies rewriting the system from scratch. This strategy carries substantial risks because: a better system is expected, business conditions continuously evolve, specifications might not exist, there may be undocumented dependencies, data migration might be cumbersome, legacy IS can be too big to cut over data, management of large projects is hard, lateness is seldom tolerated, large projects tend to bloat and homeostasis is prevalent. As an answer to the problems of Cold Turkey approach, Brodie and Stonebraker (1995) suggest the Little Chicken approach. Wu et al. (1997) lateron introduced the Butterfly approach. These are technology focused methods for data migration.

3.7 Eclectic approaches to legacy information system decision making.

Decision making as a process is described in Section 1.1.3 (p. 7). Decision making in this dissertation is defined as: "*a moment of choice in an ongoing process of evaluating alternatives for meeting an objective, at which expectations about a particular course of action impel the decision maker to select that course of action most likely to result in attaining the objective"* (Harrison, 1999). Decision making applies to all phases within the life cycle of an IS. In this section, an eclectic approach concerning the decision making regarding legacy IS is explored. Eclectic means: *"selecting what appears to be best in various doctrines or styles"* (Harrison, 1999). Harrison (1999) describes an interdisciplinary framework of decision making. Based on this interdisciplinary decision making, the decision to abandon a legacy IS is elaborated. The interdisciplinary framework of decision making according to the model of Harrison (1999) contains the following disciplines: "*philosophy, psychology, mathematics, law/anthropology/political science, sociology/social psychology and economics/statistics".*

Concerning the abandonment decision of legacy IS, also other perspectives or disciplines were mentioned. Several authors suggest a technical perspective on the legacy IS (Adolph, 1996; Furneaux & Wade, 2011; Swanson & Dans, 2000; Warren, 1999; Zvegintzov, 1984). In this dissertation the technical discipline is labeled Computer science. Others suggest a business functionality perspective (Comella-Dorda et al., 2000; Seacord et al., 2003; Van den Heuvel, 2007). They describe a gap between the supplied business functionality of the application and the required business functionality. Brooke (1994; 2002) also describes this for IS. Business as a discipline to express functionality is suggested here. Due to the fact that these perspectives (technical and functional) might be important for decision making related to legacy IS, these perspectives were added as new disciplines to the interdisciplinary framework of Harrison (1999). Different interdisciplinary approaches to decision making may be viewed as decision making models because they represent a particular segment of the real world at a given time and place under varying conditions (Harrison, 1999). Harrison (1999) states that all the processes within decision making can be seen in a different perspective. The disciplines in the interdisciplinary framework of decision making are illustrated in Figure 14.

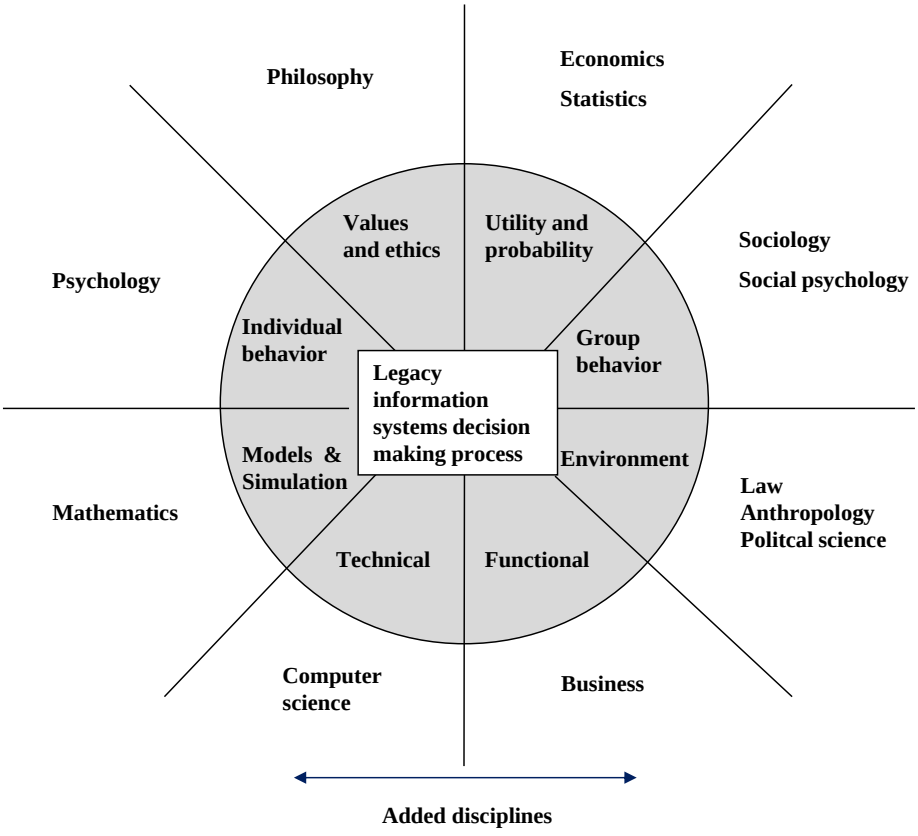


Figure 14. Interdisciplinary framework of decision making (based on Harrison, 1999)

In the following subsections the legacy IS decision making disciplines mentioned in the interdisciplinary framework (Figure 14) are elaborated. The systematic literature review from Section 3.3 (p. 26) is used as a starting point. This literature was searched for decision making disciplines of legacy IS, including references from this literature. Decision making disciplines from these references were documented. It should be noted that this is a different viewpoint from the legacy IS characteristics as presented in Section 3.4 (p. 30) and the abandonment triggers as presented in Section 3.5 (p. 33). The goal in these two sections is to explore what the characteristics and abandonment triggers are of legacy IS and its constituents. In this section the literature related to the abandonment decision making of legacy IS is explored. Based on the prevalent literature, eclectic approaches are categorized into respectively technical, functional, economical and other. Legacy IS decision making methods are explored and are categorized within the eclectic approaches.

3.7.1 Technical perspective on legacy information systems decision making

This section explores legacy IS abandonment decision making from a technical perspective. In Section 3.4. (p. 30) a lot of legacy characteristics as found in the literature are presented. However, not every legacy characteristic is a reason to decide to abandon a legacy IS. In Section 3.5 (p. 33) abandonment triggers of legacy IS are presented. This section documents technical reasons to abandon legacy IS.

Technical reasons to abandon a legacy IS were: (1) *it no longer runs on its (aging) hardware*, *taxing the limits of the legacy systems host computer* (Adolph, 1996; Furneaux & Wade, 2011; Zvegintzov, 1984), (2) *due to ongoing software maintenance, the software structure deteriorates and maintenance becomes problematic* (Adolph, 1996; Chan et al., 1996; Furneaux & Wade, 2011; Kelly et al., 1999; Sakthivel, 1994; Swanson & Dans, 2000), (3) *new technological opportunities (e.g. superior hardware and software alternatives) are available* (Chan et al., 1996; Furneaux & Wade, 2011; Sakthivel 1994; Zvegintzov, 1984), (4) *limited or ended supplier support concerning software and hardware* (Adolph, 1996; Furneaux & Wade, 2011; Kelly et al., 1999; Zvegintzov, 1984), (5) *redundant systems due to a merger* (Kelly et al., 1999), (6) *limited or no technical skills available* (Furneaux & Wade, 2011; Kelly et al., 1999), (7) *infrastructure overhaul* (Swanson & Dans, 2000), (8) *complexity* (Furneaux & Wade, 2011).

3.7.2 Functional perspective on legacy information systems decision making

This section documents functional reasons to abandon legacy IS. An IS is used to identify states of a controlled system (Brussaard & Tas, 1980). These states may refer to the past (reports), the present (registrations) and the future (plans). IS can be used to transfer data: to other places (communication), in time (storage), in contents (processing). When the application functionality is "*no longer required"* the IS can be abandoned Zvegintzov (1984). When an IS only partly capable of delivering the functionality as required by an organization, eventually there are system capability shortcomings. When this gap between business needs and IS capabilities keeps growing, the IS becomes a candidate for abandoning (Comella-Dorda et al., 2000; Furneaux & Wade, 2011; Kelly et al., 1999; Larsen, 1998; Zvegintzov, 1984). This is illustrated by Figure 15. The required functionality of an IS (business needs) will increase over time and by maintaining the IS, the IS will follow the business needs until there is a moment when the maintenance effort can no longer adopt all business needs. The gap between the required functionality and that delivered by the IS is widening and the IS becomes a legacy IS and eventually a candidate for retirement studies and abandoning.

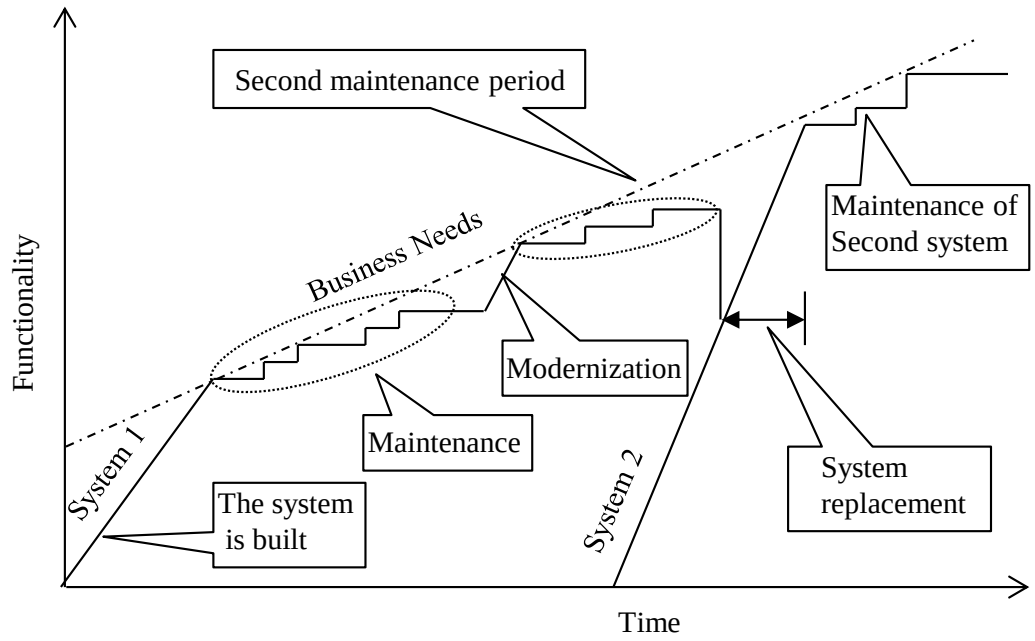


Figure 15. IS life cycle (Comella-Dorda et al., 2000)

3.7.3 Economical perspective on legacy information systems decision making

This section explores the economical[3] perspective of legacy IS abandonment decision making. This perspective assumes that the decision makers select the alternative with the highest expected value for maximizing their utility (Harrison, 1999). Economic decision making can be supported by the Net Present Value (NPV) method (Brealey & Myers, 1991) which summarizes future cash flows and discounts these cash flows with the appropriate discount factor. A decision to abandon a legacy IS will also impact future cash flows, therefore NPV can be used to calculate the economic benefits of the decision to abandon a legacy IS. Chan et al. (1996); Gode et al. (1990); Richmond et al. (2006) and Sakthivel (1994) describe a normative economic decision model, which determines an economical optimum to abandon and replace an IS. Their economic models are based on the assumption that software structurally deteriorates with age and that therefore maintenance of a legacy IS becomes so costly that it is beneficial to rewrite the IS using new technology. The normative decision model of Sakthivel (1994) follows the life cycle of software. Sakthivel (1994) argues that software maintenance costs will increase in time, due to the fact that each time maintenance is performed the logical software structure is deteriorated. This software structure deterioration makes it more expensive to add another change (Sakthivel, 1994). Sakthivel (1994) also introduces another cost: the obsolescence cost. This is an opportunity cost of not using the latest software engineering environment. These two maintenance cost components increase during the age of software. An example (based on Sakthivel, 1994) is presented in Figure 16: this figure illustrates the increase of the total maintenance cost due to deterioration and obsolescence within time.

[3] Within the original model of Harrison (1999) economics and statistics were combined, referring to utility for economics and probability for statistics. In this dissertation, however, statistics was combined with mathematics.

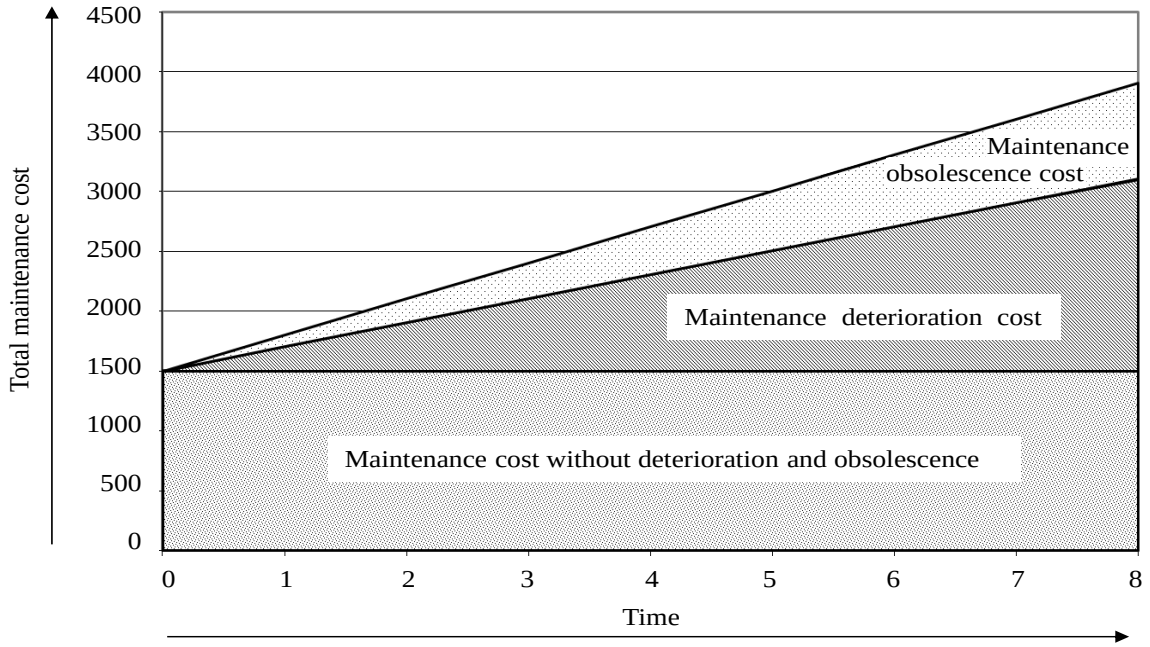


Figure 16. Total maintenance costs, including an increasing deterioration and obsolescence component (based on Sakthivel, 1994)

At a certain point in time the increase in software maintenance costs due to deterioration and obsolescence makes it beneficial to redevelop the software, including an initial investment in a new system, and thus start without deterioration maintenance cost and obsolescence maintenance cost. In his investment model Sakthivel (1994) takes the time value of money into account: he introduces the "*equivalent annual lifespan development costs*", which include initial software development[4]. Further he introduces the "*equivalent annual lifespan maintenance costs"*, which include the total maintenance deterioration costs (corrective, adaptive and perfective). By summing the equivalent annual lifespan development costs and the equivalent annual lifespan maintenance costs, the: "*equivalent annual total lifespan costs"* can be calculated (Sakthivel, 1994). This is illustrated in Figure 17.

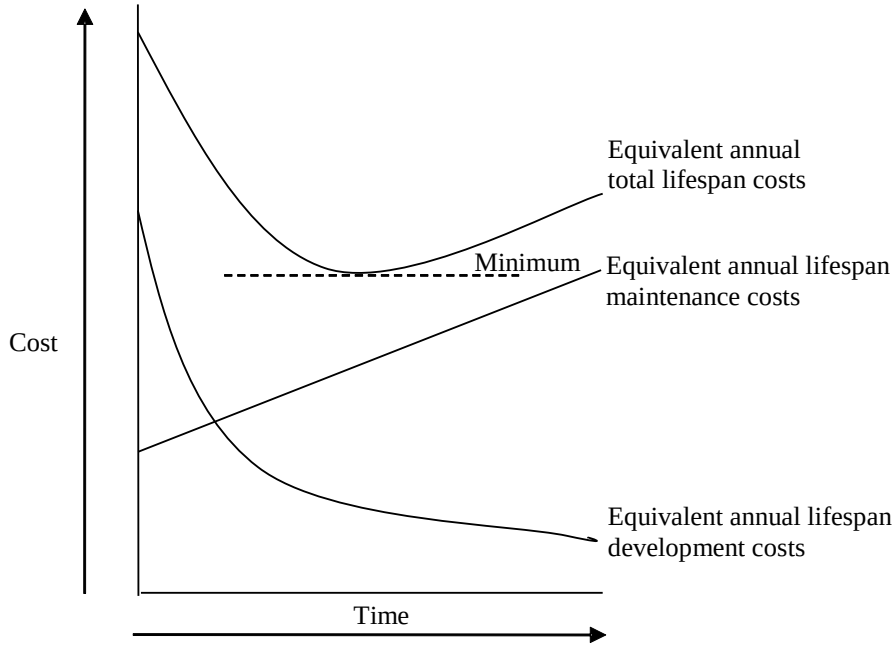


Figure 17. Equivalent annual total lifespan costs (Sakthivel, 1994)

[4] This is a simplified model of Sakthivel's (1994) original work.

Figure 17 illustrates that there is an optimal replacement point (minimum) from an economical perspective.

3.7.4 Other eclectic perspectives on legacy information systems decision making

In this section the other eclectic perspectives related to legacy IS abandonment decision making are presented. The mathematical perspective primarily appears in the decision making process through the development of models (Harrison, 1999). In the decision making process, models are used as follows: (1) Models may help determine the feasibility and effectiveness of various objectives and goals prior to the spending of time and effort on search and evaluation (Harrison, 1999). (2) Models may be used in formal search activities to uncover a comprehensive range of relevant alternatives with a minimum of effort (Harrison, 1999). (3) Alternatives may be evaluated and compared by revising the model until the actual results appear to conform to the objective, thereby reducing the uncertainty of the outcome (Harrison, 1999). (4) Once the decision is made and implemented, the model can still be used as a control device by monitoring the occurrence of the expected outcome within established time and cost constrained (Harrison, 1999). Models can be supported by statistics. Swanson and Dans (2000) modeled the relations between system size, portfolio complexity, system age and maintenance effort and life expectancy and tested these relations statistically.

Swanson and Dans (2000) suggested that management should “equilibrate” (balance) its allocation of system maintenance effort with its estimate of a system´s remaining life. An increase in the maintenance effort may extend the system´s estimated remaining life. However, the estimate of a shortened remaining life may call for a reduction of the system maintenance effort. Swanson and Dans (2000) also identified that system age, size and portfolio complexity play significant roles in accounting for the maintenance effort and remaining life expectancy and that the latter two are positively related. They also suggested that larger systems may exist longer than smaller systems and that older systems were indeed associated with shorter lives (Swanson & Dans, 2000). Further Swanson and Dans (2000) suggested that larger systems had a longer life expectancy while normative research (Gode et al., 1990) suggested the opposite.

Furneaux and Wade (2011) modeled an IS replacement intention based on the technical organization environment (TOE) framework (Tornatzky & Fleischer, 1990). The model (Furneaux & Wade, 2011) included the following change forces (in favor of IS discontinuance): system capability shortcomings, system reliability, system support availability and system support cost. In contrast with this they defined the following continuance inertia: system investment and technical integration. The model was statistically tested and it was disclosed that system capability shortcomings, limited availability of system support and low-levels of technical integration were key determinants in increased intentions to replace an existing system.

There was a psychological perspective on legacy IS abandonment decision making: it was difficult to refute the relevance of psychology to the behavior of the decision maker. For the most part human behavior is learned, not inherited (Harrison, 1999). The basic psychological force affecting a decision maker is his or her personality (Harrison, 1999). Personality may be said to encompass the characteristic traits and patterns of adjustment of the person in his interrelationship with others and his environment (Harrison, 1999). Personality effects on decision making are a preference for high, low or moderate risks, a preference to look for problems and to keep control of the situation, or alternatively, a preference to give up control and to wait for problems to emerge on their own and a preference for innovation or for proven methods (Harrison, 1999).

Although no literature on the psychology perspective on legacy IS decision making has been found, it can, however, be assumed that these aspects will play a role in legacy IS decision making. This also holds for sociological, philosophical, law, political science and anthropology perspectives to legacy IS abandonment decision making. No literature has been found, but it can be assumed that these aspects will play a role in legacy IS decision making.

3.8 Identification process of legacy information systems

Several authors propose methods to identify legacy IS. These methods all start by having an IS asset repository: the IS asset repository can be a list of IS, stating IS characteristics (including legacy characteristics). Examples of application lists with attributes are given by Benson et al. (2004), Gottling and Torgnysdotter (2002) and Warren (1999). Warren (1999) suggests collecting information from different sources, e.g. business documents, system documentation, expert interviews and the operational system. This results in, for example, source code, change history documents, development documents and user manuals. Warren (1999) states that the written documentation does not always reflect the current state of the system. In case the documentation is inadequate, reverse engineering could be used to help construct the system models. Experts are a valuable source in this process and not limited to technical experts. Business experts, who understand the business needs are also invaluable. When the IS asset repository list is complete, organizations can identify the legacy characteristics of the IS. However, the problem is that organizations rarely evaluate IS quality adequately (Larsen, 1998).

3.9 Practical abandoning of legacy information systems

If the decision to abandon the legacy IS is made, then the next step is the practical abandoning of a legacy IS. This is also referred to as the disposal phase (Rittinghouse, 2004; White, 2007). The disposal activities ensure the orderly termination of the system and the preservation of vital information of the system (Rittinghouse, 2004; White, 2007). Several approaches have been described in the literature.

The disposal phase should also include provisions for safeguarding sensitive information (Howard, 2011). It is important to ensure that resources and assets are protected (Grance et al., 2004). Media sanitization, protection of IS hardware usually requires that residual magnetic or electrical representation of data be deleted, erased, or overwritten and that any system components with non-volatile memory are erased, because any residual information may allow data to be reconstructed, providing access to sensitive information by unauthorized individuals. The removal of information from, a storage medium is called sanitization (Grance et al., 2004).

Different kinds of sanitization provide various levels of protection (Grance et al., 2004). A distinction can be made between clearing information and purging information (Grance et al., 2004). Clearing information is removal of sensitive data from a storage devise at the end of a processing period in such ways that there is assurance, proportional to the sensitivity of the data, that the data may not be reconstructed using normal system capabilities (Grance et al., 2004). Purging is the removal of data from a storage device at the end of a processing period in such a way that there is assurance, proportional to the sensitivity of the data that the data may not be reconstructed except through open ended laboratory techniques (Grance et al., 2004). Degaussing, overwriting and media destruction are some of the methods for purging information (Grance et al., 2004). Degaussing is a process for erasing the magnetic media; overwriting is a process for writing non-sensitive data in storage lavations previously containing sensitive data (Grance et al., 2004). The following processes may destroy media (Grance et al., 2004): destruction at an approved metal destruction facility (e.g. smelting, disintegration, or pulverization), incineration, application of an abrasive substance to a magnetic disk (Grance et al., 2004).

When the legacy IS are practically abandoned, a disposal plan should be written (Rittinghouse, 2004). An example of such plans is provided by the U.S. Department of justice, the department of justice systems development life cycle guidance document, chapter 12 (January 2003), www.justice.gov/archive/jmd/irm/lifecycle/ch12.htm (accessed December 2014). Another disposal plan (a template, a checklist and a practice guide) is provided by the U.S. department of Health and Human services www.hhs.gov/ocio/eplc/Enterprise%20Performance%20Lifecycle%20Artifacts/eplc_artifacts.html (accessed December 2014).

3.10 Summary and conclusions

In this chapter, the literature concerning legacy IS and decision making was elaborated. First, IS management was discussed. IS management literature describes that the number of relations between components of an IS increases exponentially with the added number of components. This implies that larger IS or ecosystems of IS will be increasingly difficult to manage. Furthermore, life cycle management and IS portfolio management have been described. IS life cycle management describes in-depth the abandonment phase from an individual legacy IS perspective, whilst IS portfolio management uses a more holistic view and describes the abandoning of an individual legacy IS within a set of IS. Furthermore, a systematic literature review on legacy IS and abandonment decision making was performed. This resulted in a meager twelve relevant research papers. This low number, of which seven have been published before 2000, expresses that research in this area remains scant. Concerning the aging of legacy IS, it was found that different authors define different characteristics of legacy IS and an overview was created including 74 legacy IS characteristics. These characteristics can be categorized in IS and its constituents (applications, hardware, data sets, people and procedures). All components influence the aging of legacy IS. Subsequently, 11 abandonment triggers were identified (i.e. what makes organizations decide to abandon particular legacy IS?) (see Table 7, p. 34).

Although there was a limited amount of literature on IS portfolio management, research on how managers actually make the decision to abandon legacy IS remained unattended (Swanson & Dans, 2000). Also, good practices or guidelines in this area were hardly found. Abandonment decision making methods of legacy IS were based on the fact that the responsible management should decide to: (1) continue the maintenance, (2) re-engineer the legacy IS, (3) abandon the legacy IS and dismantle the IS, (4) replace the legacy IS by a new IS (see Figure 13, p. 35). Another decision making method was based on a legacy IS portfolio assessment. Legacy IS were scored on business value and technical quality. Legacy IS scoring low on both business value and technical quality were typical candidates for abandoning. Furthermore, there were methods in the literature that have a legacy IS continuance perspective. These were focused on the maintenance of IS or the evolution of legacy IS by modernizing the legacy IS (Seacord et al., 2003; Van den Heuvel, 2007; Warren, 1999). If legacy IS evolution is no longer viable, this will result in an abandonment decision.

Finally the process of identification of legacy IS and practical abandoning of legacy IS were described. Identification of legacy IS, always started with initializing an IS repository. Within practical abandoning of legacy IS, the disposal phase should include provisions for safeguarding sensitive information (Howard, 2011). Good practices were hardly found in the literature.

4 INTRODUCTION OF EXPLORATORY CASE A

In this chapter Case A is introduced. Case A was a large semi-public administrative national organization (located in Europe). Case A was the result of a complex consolidation of six similar branch organizations. Therefore, Case A included an exceptional amount of redundant IS. Case A was selected, because of this complexity and the fact that Case A initiated a program to rationalize the IS portfolio. Senior management provided full access to all documentation and fully supported this longitudinal research. The longitudinal research provided in-depth insight in the arguments of all stakeholders and the researcher could also capture changes in time. Overall, 471 legacy IS were identified and abandoned in a five year period. Furthermore, several good practices and methodological guidelines to abandon legacy IS were identified.

Within this case two subcases were investigated, referred to as Trail 1 and Trail 2. Chapter 4 focuses on the description of Case A and the chronological overview of both research trails. The actual research questions and their answering are described in Chapter 5. First the antecedents of Case A are described (Section 4.1). Then the two research trails are described (Section 4.2), Trail 1 describes the merger of six information functions (Section 4.3) and Trail 2 described the practical abandoning of legacy IS (Section 4.4). Finally there is a summary and there are conclusions (Section 4.5).

4.1 Antecedents of Case A

Case A was a semi-public administrative organization with almost 18,000[5] employees. Case A emerged from the consolidation of six former branch-oriented service organizations (referred to as: "organization 1 to 6"[6]). The six branch organizations also ignited from mergers and consolidations (twenty-five branch organizations had been counted in 1997). The six branch organizations all supplied social security services such as unemployment benefit schemes and insurance for (temporary) disablement, along with other services, such as pension fund services. These services were delivered with help of the information provisioning of each branch organization. In 2002 the government decided that the unemployment benefit schemes and insurance for (temporary) disablement service supplied by the branch organizations had to merge into the central Case A organization. Consequently, each branch organization had to untangle their social security services and other services, like pension services.

At inception, Case A operated six different information provisionings. The six information provisionings had been entangled into information functions supporting business units. Successively these information functions were consolidated by the newly incepted business units by means of consolidation programmes. These consolidation programmes were focused on consolidating the social security business processes of Case A within each individual business unit. The consolidation programmes focused on reducing all redundant business processes. It took six years to standardize and consolidate the business working processes including the supporting IS. Case A implemented four major consolidation programmes, which were associated with the four main business units. Besides these consolidation programmes, some smaller business units and support staff such as Finance, HRM, Procurement and ICT also standardized their work processes including supporting IS. At the inception of Case A in 2002 there were more than 800 applications, of which more than 600 applications were abandoned in 2011 (nine year period).

The ICT departments also went through several reorganizations. The former branch organizations all had their proprietary ICT departments and ICT suppliers. Overall, they had been staffing thousands of ICT professionals. The general policy guideline was that most ICT departments and ICT suppliers would remain with the former branch organizations and should provide contracted ICT services to Case A. This meant that much "ICT knowledge" (being maintenance and development of applications and data center) remained with the former

[5] In 2007.

[6] Organization 2 and organization 3 were formally already merged before they joined Case A, however, practically only their name was changed. Therefore, they are considered different organizations in this study. Besides these branch organizations also the government organization responsible for social security merged with Case A.

original branch organizations or their suppliers. Many of these contracts were lumpsum based, implying that Case A lacked detailed ICT cost insight and detailed knowledge of the information function.

Furthermore, two of the original branch organizations sold their data centers to large private ICT service suppliers and later on these suppliers again merged with other large suppliers. The management of the central ICT department was responsible for consolidating the generic ICT infrastructure including, WAN, LAN, telecom, client and peripherals (PCs and printers) and data centers. In accordance with European law, Case A subsequently tendered various logical ICT domains (telephone, WAN, LAN, distributed computing, software, application development, application maintenance and data center). In 2005, Case A tendered the logical ICT domain “data center” and a new data center supplier was selected. This data center merger was the trigger for Case A to start a programme named: “Abandonment and Migration” (A&M). The goal of this A&M programme was to abandon as many applications as possible, before the remaining applications were migrated to the new data center.

The consolidation programme managers were responsible for deciding which information function or parts of the information functions from the former branch organization were preferred as target information function and decided upon the planning of any merging or abandoning of IS. Decision making was done at application level. Figure 18 illustrates the consolidation programmes. It illustrates the information provisionings of the six former branch organizations, each operating proprietary IS and supporting the business processes of their business units. These former branch organizations were dismantled and all similar information functions were integrated into new Case A business units. These newly created business units decided upon their new standardized working processes and accompanying IS.

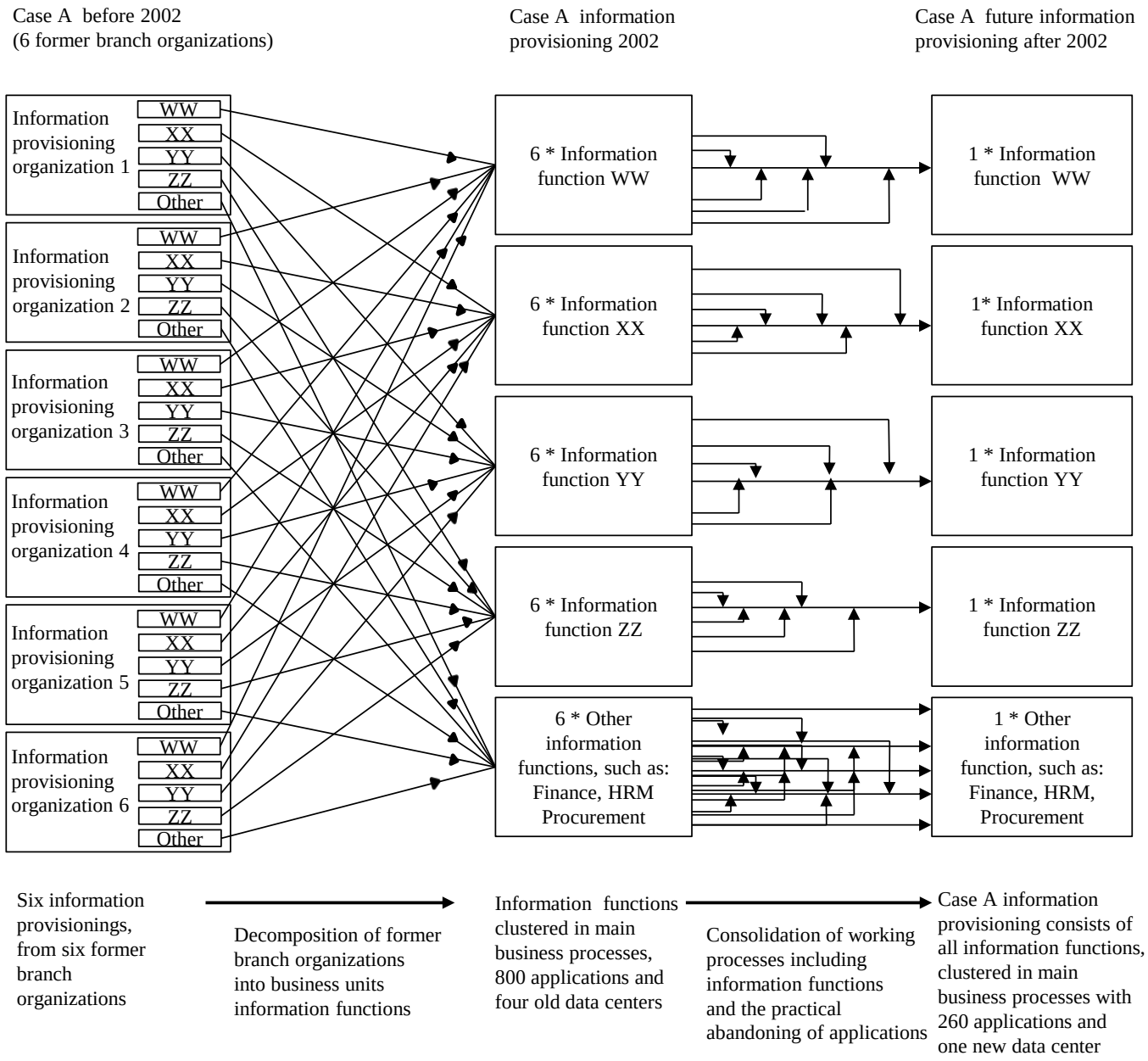


Figure 18. Six former branch organizations consolidate into Case A

4.2 Description of the research trails

Two particular research trails have been investigated in this dissertation. Trail 1 concerned the consolidation of unemployment benefit schemes within the business unit WW. This was a merger of five information functions to one target information function. Trail 2 concerns the A&M programme, which abandoned 471 applications and four data centers.

Trail 1 focused on the decision making in the unemployment domain where five information functions were merged into one existing information function. This trail provided insights into the decision making concerning legacy IS and the impact on stakeholders affected by this decision. It also provided information concerning good practices (problems and success factors).

Trail 2 focused on the processes related to the practically abandoning of legacy IS in the A&M programme. This trail provided insight into the process of practical abandoning applications and, due to the fact that these applications had different ownership, different technology, different architecture, different size and different

suppliers, it provided information concerning good practices (problems and success factors). Trail 1 and Trail 2 are described in the two following sections.

4.3 Trail 1 - the merger of six information functions

Trail 1 concerned a longitudinal study of the merger of six information functions within the newly established business unit WW of Case A. Detailed information about this merger was available, including business cases, financial data, and interviews with responsible decision makers and other stakeholders. Stakeholders affected by the abandonment decision of these legacy IS were interviewed.

In this section the merger of six different unemployment benefit scheme information functions are described. It was enforced by law that Case A should start on 1 January 2002. Before January 2002 there had been different pre-consolidation initiatives. The former original branch organizations initiated research to consolidate the unemployment benefit schemes. Architects from all the original branch organizations evaluated the information functions of the existing unemployment benefit schemes. They compared all the IS in their six information functions on the following aspects: production volume, functionality, quality and life expectancy, maintenance effort and cost, exploitation cost, reusability and flexibility, constraints and limitations, and ownership (e.g. proprietary software, intellectual property or supplier ownership).

There had also been an initiative of the management of the unemployment benefit schemes business unit. They proposed a new consolidation strategy to the board of directors. The concept of this consolidation strategy was that the old information functions should be kept for the time being and that these old information functions should be wrapped in a uniform front end in order to provide one uniform corporate identity and one standard way of working. At a later stage the back end of the information function would be standardized. The board of directors, however, preferred the solution to subsequently merge each former information function into one already existing information function. The unemployment benefit scheme (performed by the business unit WW) was one of the many other consolidations of the Case A business units.

In November 2002, the management of the WW business unit agreed upon the merger strategy. The decision had been made to adapt the IS of former organization 1 for the entire business unit and subsequently migrate the other five information functions. Organization 1 already covered 60% of all Case A clients. For each of the five information functions an impact analysis for a merger was made, including applications, interfaces, functionality, working processes and technical infrastructure. Subsequently, first organization 2 and organization 3 were merged, followed by organization 4, organization 5 and finally organization 6. Overall, it took six years to consolidate all information functions into the WW business unit. The time frame of the merger of the six information functions into one information function is illustrated in Figure 19.

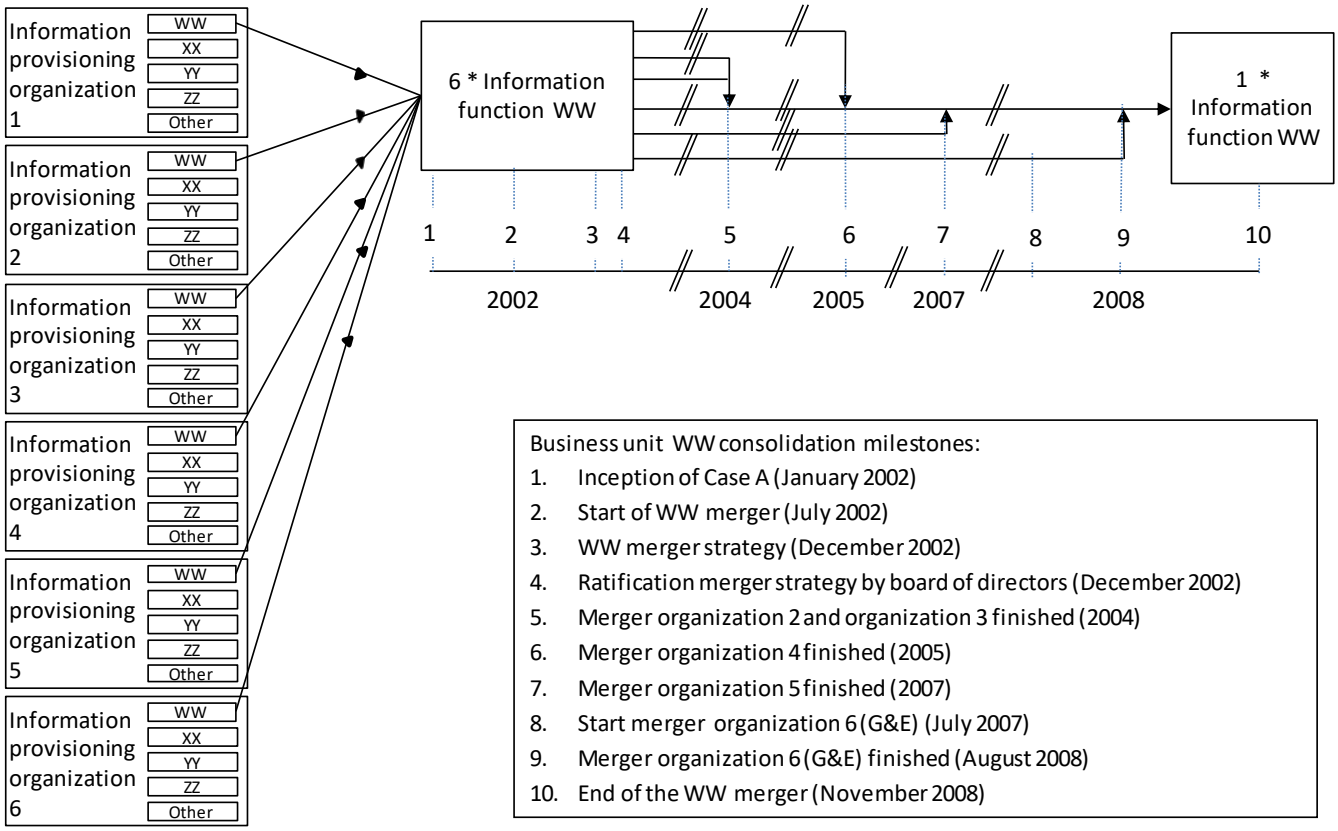


Figure 19. Trail 1 - merger of six information functions within the WW business unit

4.3.1 Subcase study Government and Education, the information system ABC

In Trail 1 the abandoning of five information functions was investigated in detail. This section describes a perspective from abandoning the major IS within the information function "Government and Education" (G&E) (G&E is the name of organization 6). G&E is the information function supporting the primary process of unemployment benefit schemes within the Government and Education sector. In 2000 the main IS within the G&E information function (named "ABC") had been developed in order to replace two older systems. Initially, the ambition level regarding the functionality of ABC was high. However, as time passed and the deadline of implementing ABC was coming closer, ambitions were lowered. Although the information function including ABC were operational on the first of January 2001, there was still a gap between the delivered and initially agreed and required functionality. In the following years the information function (including ABC) was improved, closing the gap between required functionality and supplied functionality. However, in 2002 the decision was made to abandon the smaller ABC of organization 6 and to migrate to XYZ of organization 1 which served more users, although ABC was superior in functionality. When finally ABC was abandoned, the users (5%) of the total population migrated to XYZ and lacked the functionality they had before with their proprietary ABC system.

4.4 Trail 2 - the practical abandoning of legacy information systems

In Trail 2 the practical abandoning of 471 applications and four data centers was investigated. These abandonings were the result of the A&M programme, which started on the first January of 2006. This section first explains the background of this programme, then describes the strategy and finally the results.

As a result of the consolidation of the six organizations, Case A inherited a lot of similar IS including the support of four main external ICT suppliers. The strategy was to consolidate the information functions and to reduce the underlying number of IS. Therefore target IS were defined (sometimes even new IS were introduced). Subsequently, mergers to target IS were initiated. The decision making was done by business management.

Initially in the period 2002-2004, the focus of the programme was on merging, while the abandoning of IS received less attention. In accordance with European law, Case A was obliged to tender for data center functionality in 2004. This also meant that Case A had to migrate all IS to this new data center. This was the trigger for Case A to initiate the A&M programme. A&M targeted both applications and data center technology. Goals of the A&M programme were:

1. Lowering cost of exploitation.
2. Lowering functional management effort.
3. Optimal usage of computer capacity and additional economies of scale.
4. Realization of uniform/standardized system platforms.

These goals were summarized at the start of the A&M programme in 2005 as follows: "*efficiently phase out before 2009 the current four ICT data centers, abandon as many applications as possible and migrate the remaining applications to the new data center if abandoning of the applications is not possible"*. In February 2006, 410 applications from four suppliers were selected for migration to the new data center and 260 were selected for abandoning.

4.4.1 Abandonment and migration strategy

At the start of the A&M programme in 2006, Case A was confronted with four data center suppliers (who also did do the application maintenance on the IS) and one new "target" data center supplier. Although Case A started with six different information provisionings in 2002, two ICT suppliers from the former branch organizations had already merged, since 2002. Therefore in 2006 there were only four data centers. After the tender as many applications as possible were abandoned, the other applications were moved to the new data center. However, maintenance and development were still provided by the old suppliers. This is illustrated in Figure 20.

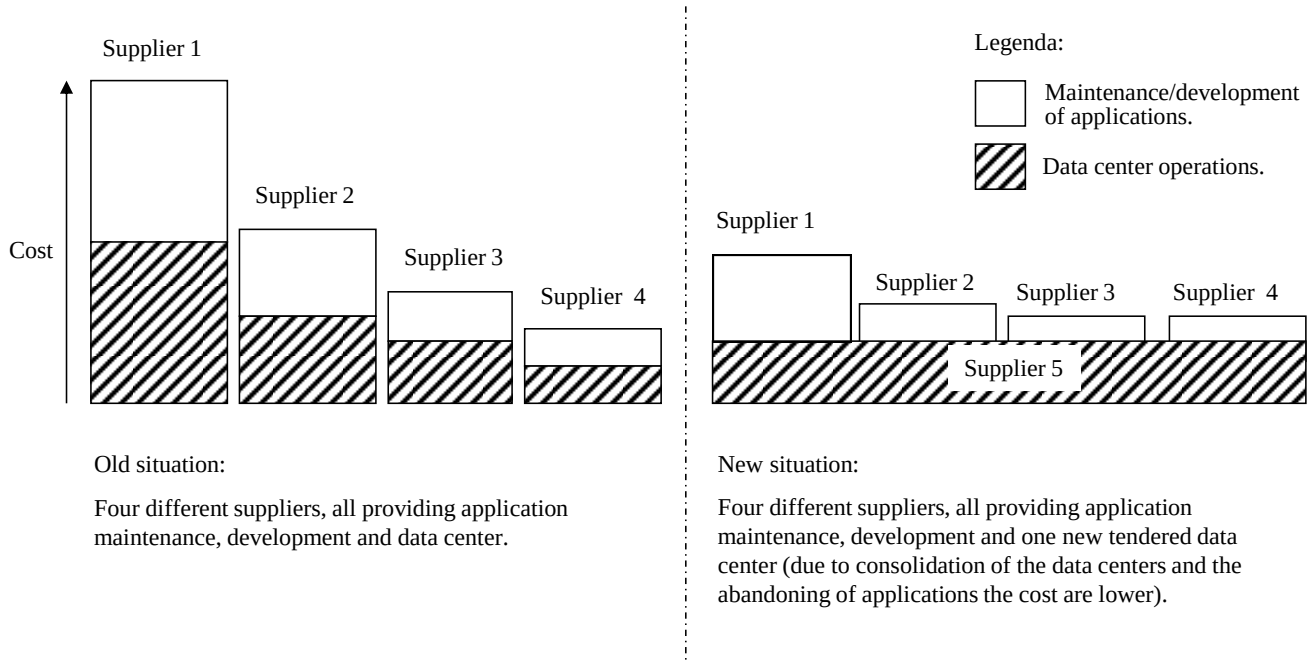


Figure 20. Old and new situation after the tendering of the Case A data center

In 2006, Case A decided on a phase out strategy and a migration strategy of the current four data centers to the newly selected data center. It was decided to phase out data centers sequentially (this means supplier by supplier). In 2006 the applications of the first supplier were abandoned or migrated to the new data center, followed in 2007 by supplier two and three. In 2008 the applications of the fourth supplier were abandoned or migrated to the new data center. The first supplier was the smallest supplier and was selected to gain experience

with these A&M processes. Then in 2007 the applications of the second and the third smallest suppliers were abandoned and migrated. In 2008 the start was made to abandon or migrate the applications of the largest supplier.

4.4.2 Abandonment and migration strategy per application

After choosing the A&M strategy, Case A further refined the A&M strategy for the individual applications. A small project team (consisting of senior ICT managers and external industry experts) applied the following process which was described in the A&M programme plan[7]: in the first stage named "*strategy and implementation preparation*", information about applications was collected and analyzed. In some cases additional information had to be obtained. The decision to abandon or migrate included a concept planning. The following four steps were performed as part of stage one:

1. Data collection:
   - Which applications run on which hardware from which supplier and within which contract?
   - What is the strategy concerning the application (maintain and migrate, or abandon)?
   - What are the requirements of the business concerning the date of migration or abandoning and the way of migration or abandoning?
   - What are the capabilities of the new supplier concerning production capacity?
   - What are the capabilities of the new supplier concerning migration capacity?
   - What are the financial consequences concerning the migration of the platform and applications?
   - What are the necessary process steps for abandoning or migration and how much time will this cost?
   - What are the expected costs for abandoning or migration of the application?
   - What capacity is available within Case A to fulfill the abandoning or migration activities?

2. Pre-intake:
   - If there were technical uncertainties concerning the decision of an application abandon or migration strategy, then a pre-intake is suggested with the current data center supplier, the target data center supplier and Case A. Goal is to analyze the quality of the application and assess available documentation.

3. Decision to abandon or migrate (the decision was always made by the IS owner, e.g. director, business unit director).

4. Creation of the concept cluster plan including:
   - Applications:
     - Overview of applications (name, hardware platform, supplier applications, maintenance applications).
     - Importance of the application (low, medium, high).
     - Strategy (maintain and migrate, or abandon).
     - Specials (deviations from the standard process).
   - Planning:
     - Dependencies between (clusters) of applications.
     - Required milestone data.
   - Budget.
   - Governance (contact persons).
   - Contractual obligations (what contacts and agreements are there with the current supplier).

[7] Internal company report, April 2006 (p. 15-21).

In the second stage named *"process operation",* the practical abandoning or migration operation was performed. The input for this phase is the concept cluster plan from stage one. In stage two there was a further elaboration of data. The following steps were performed as part of stage two:

1. Intake applications. In this step there were meetings with the IS owner (e.g. director, business unit director) of the applications. The decisions to migrate or abandon were evaluated and further discussed; these discussion would lead to an application baseline. This application baseline was the basis for the abandoning or migration of the application. The intake included the following activities:
   - Scan of the system (platform) environment, including operating system from the current suppliers.
   - Scan application environment (including middleware and databases) from the current suppliers.
   - Scan available documentation (maintenance documentation, production manuals, and service level reports).
   - Scan on known errors and workarounds.
   - Scan on maintenance organization, functional maintenance organization, application maintenance and development organization and technical maintenance organization ITIL.
   - Optionally an audit could be performed of the application or the hardware platform in case the intake did not provide enough information to prepare the application base.

2. When the application baseline was complete for the migration of applications, agreements with the new supplier were negotiated (this was only for migration).

3. Detailed cluster plan. Based on the application baseline and the concept cluster plan from phase 1 and the agreement with the new supplier, a detailed cluster plan was produced and finalized. Five scenarios for A&M were possible: conditional and unconditional abandoning and within migration there was a customized or identical implementation or there was lift and shift. This is explained in Figure 21.

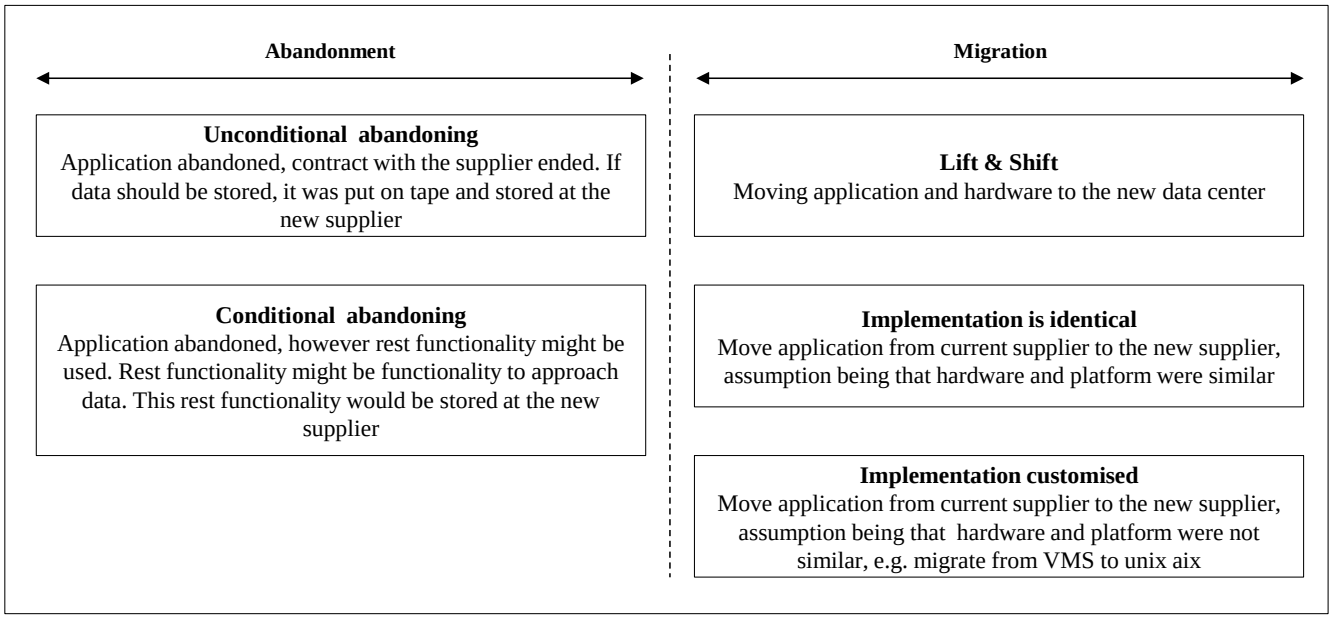


Figure 21. The five scenarios for abandonment and migration

4. The following activities were performed when abandon or migration was selected:
   - Abandon:
     - Communicate with users that application was to be abandoned.
     - Prepare for abandoning.
     - Restrict users from accessing the application.
     - Convert data and secure data.

- Check for incidents.
- In case of conditional abandon, test access to data with the rest functionality before abandoning and do final check whether all activities were performed.

- Migration:
  - Within migration a lot of activities were performed. However, because migration is not the core subject of this dissertation, this is not described.

5. Acceptance test. In the case of a conditional abandon, the user was to test the accessibility and the rest functionality of the abandoned application.

6. When an application was migrated or abandoned with a conditional abandon and was accepted by the user, A&M management would support these users for one month.

All documents (including concept cluster plan, application baseline and finalized cluster plan) were also used for planning purposes in the A&M calendar. This A&M calendar was used for program management and for communicating the status of the project throughout the organization.

Case A started with a baseline of 670 applications in 2006, and at that time had the plan to abandon 260 of these applications and to abandon four datacenters. The initial plan was, therefore, to migrate 410 applications to a new data center. However, in 2011, 471 applications had been actually abandoned and 192 applications had been actually migrated. This revised insight, has also been researched in this longitudinal study. The planned number of applications varied as is illustrated in Figure 22.

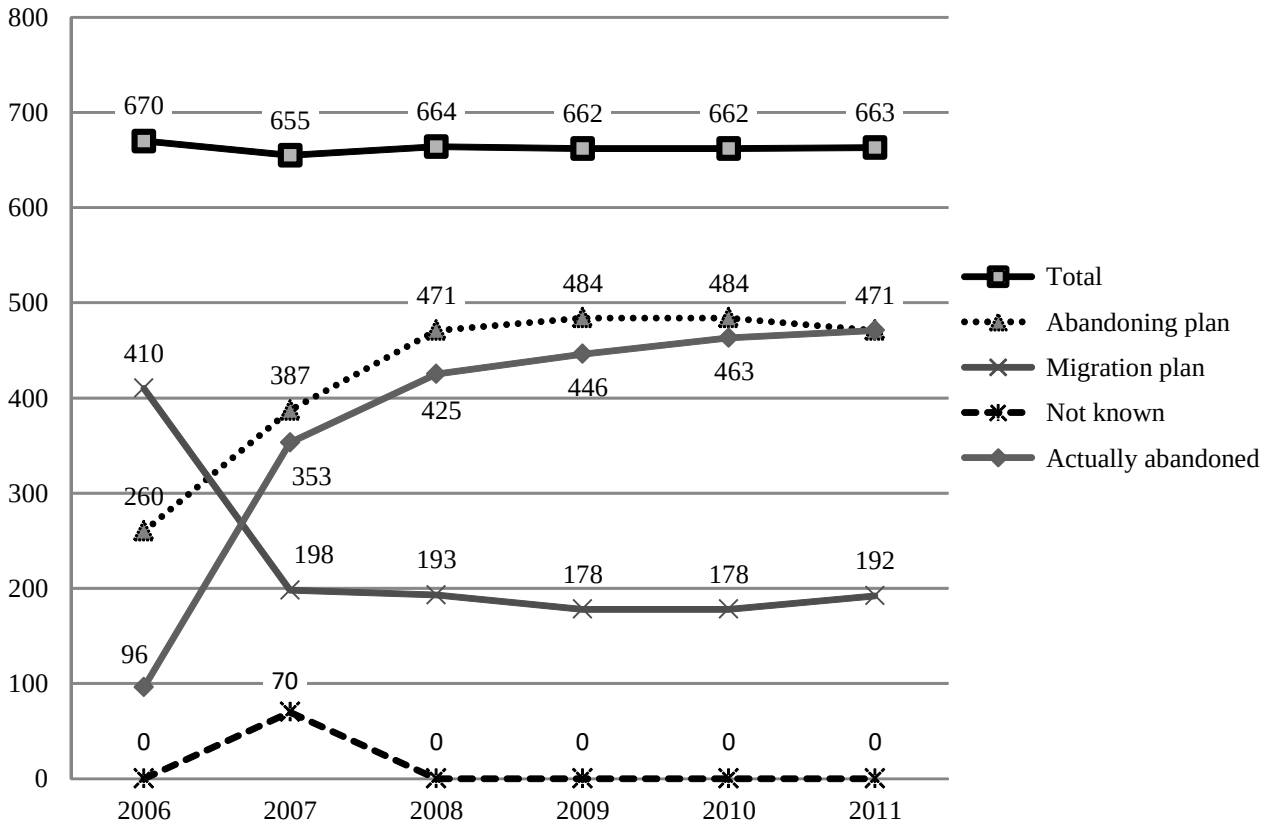


Figure 22. Envisioned numbers of abandoned applications throughout 2006-2011

Overall, this longitudinal research took five years. An advantage of a longitudinal research is that it will also capture change. Longitudinal studies have considerable potential for yielding rich data that can trace changes over time, and with great accuracy (Gorard, 2001).

4.4.3 Executing the abandoning process

In the former section the A&M programme was described. In this chapter the results of the operational abandoning activities are elaborated upon. The abandoning process of Case A can be summarized as follows. First, a detailed impact analysis per abandoned application was made. This impact analysis resulted in a document describing the abandoned functionality and information about which data to keep. Business was responsible for this impact analysis. Then a signature from the department responsible for keeping all the data in Case A (documentary information department) was necessary. When it was necessary to keep the data there were two possibilities. It was possible to burn the data on a CD or it was possible to store the data in a special archive named: "*Historical data storage*". Then a detailed plan for abandoning the application was made (date, risks and cost). Next the supplier was informed by letter that Case A would end the contract and that the application could be practically removed (sometimes exit costs were paid). The business was subsequently responsible to disconnect the users and inform the abandon manager of the A&M programme. Then the supplier would abandon the application. Finally, there was a check that the contract had changed (the application was removed), that the benefits of not paying for exploitation and maintenance cost were booked as a benefit of the A&M programme and that the configuration management database (CMDB) was updated. This process was facilitated by a practical application abandon document, which the different participants had to sign for approval and completion. The costs[8] of abandoning one application were estimated between 15k Euro and 300k Euro. Furthermore the A&M programme estimated that the weighted average cost of abandoning a single application was approximately 75k Euro.

4.5 Summary and conclusions

In this chapter the characteristics of Case A have been described. Case A was established through the consolidation of six former branch organizations and eventually abandoned 471 applications. Case A could be researched as a longitudinal case throughout a five year period. Senior management supported unrestricted access to stakeholders and documentation. An advantage of this longitudinal research is that it yielded rich data that traced changes over time (Gorard, 2001).

Two trails of research are distinguished. Trail 1 focused on the decision making of legacy IS within information functions of former branch organizations. Due to the fact that Case A tendered a new data center, it was necessary to migrate all applications to this new data center or to abandon legacy IS, this was referred to as the A&M programme. Trail 2 investigated this A&M programme.

In 2006, Case A maintained 670 applications and had plans to abandon 260 applications of these applications and close four datacenters. The resulting 410 applications had to be migrated to a new data center. However, in 2011, 471 applications were actually abandoned and only 192 applications were migrated. The research results, which were obtained from Case A, will be further described in Chapter 5.

[8] Internal company report, April 2006 (p. 27).

5 DATA ANALYSIS OF THE EXPLORATORY CASE STUDY

"Research is formalized curiosity. It is poking and prying with a purpose".
---Hurston, Z. N. (1942)---

5.1 Introduction

In this section the results of the exploratory case study, introduced in Chapter 4, are described. First the exploratory research model (Section 5.2) is described; this includes the research questions and the sub-research questions (SRQ). Subsequently, the exploratory case study protocol (Section 5.3) and the process of analyzing the interview transcripts (Section 5.4) are described. Then the aging process of legacy IS (Section 5.5), the abandonment decision making process (Section 5.6) and the practical abandoning process (Section 5.7) of legacy IS are described. Finally, other results are presented (Section 5.8) and this chapter ends with a summary and conclusions (Section 5.9).

5.2 Exploratory research model

The analysis of the case is divided in two sections and based on the inductive approach (see Section 2.3, p. 13). The first section is based on the three predominant life cycle stages in the maintenance and abandon phase (Berghout & Nijland, 2002), being, (1) *the aging process of legacy IS*, (2) *the abandonment decision making process of legacy IS*, and (3) *the practical abandoning process of legacy IS*. Together they will provide input for the second section of this research model which is to (4) *propose a method to abandon legacy IS*. The rationale for these stages, including their supporting research questions and sub-research questions (SRQ), is described in this section.

IS follow a life cycle like other systems: they are incepted, operated, maintained and finally abandoned (Berghout & Nijland, 2002) (see Figure 9, p. 23). During the maintenance phase IS are subject to structural deterioration and obsolescence with age (Lehman, 1980a; Swanson & Dans, 2000) and throughout the years, the IS will increasingly resist meeting the requirements of the organization. This increasing resistance is referred to as "aging" of the IS (Lehman, 1980a; Swanson & Dans, 2000) and these systems are often referred to as legacy IS. In the literature concerning the aging of legacy IS (see Section 3.4), several authors referred to different definitions and several characteristics of legacy IS (Adolph, 1996; Alderson & Shah, 1999; Brodie & Stonebraker, 1995; Brooke & Ramage, 2001; Brooke, 2002; Daga et al., 2005; Kelly et al., 1999; Lawrence, 2007; Long, 2006; Sommerville, 2007; Ulrich, 2002; Van den Heuvel, 2007; Warren, 1999). These authors describe characteristics of the aging of legacy IS. However, an in-depth analysis of this legacy IS aging was not found in the literature. The first life cycle stage of a legacy IS is: (1) *the aging process of legacy IS* (Berghout & Nijland, 2002). The first associated research question was:

Research question 1:
How do legacy IS age?

In order to answer this research question, the following five sub-research questions (SRQ) were proposed.

Many studies do not define the IS artifact of research (Orlikowski & Iacono, 2001), and studies that define legacy IS (Brodie & Stonebraker, 1995; Brooke, 2002; Brooke & Ramage, 2001; Lawrence, 2007; Long, 2006; Ulrich, 2002; Sommerville, 2007) apply dissimilar definitions. This study preferred the definition of Brodie and Stonebraker (1995) which is: *"a legacy IS is any IS that significantly resists meeting organizations' requirements"*. To study whether this definition was applicable in Case A, the first SRQ 1 was:

- SRQ 1: How are legacy IS defined in practice?

In the literature a lot of different characteristics of legacy IS and its constituents are provided, e.g. legacy IS are designed in stove pipe fashion or silos (Daga et al., 2005; Kelly et al., 1999; Ulrich, 2002; Van den Heuvel, 2007), or there is inadequate documentation of the system (Adolph, 1996; Warren, 1999), or system skills are hard to find (Alderson & Shah, 1999; Kelly et al., 1999) (see Section 3.4, p. 30). The information was fragmented and no overall view was found in the literature. In order to gain more knowledge of the characteristics of the legacy IS and its constituents in Case A, the following SRQ 2 was defined:

- SRQ 2: What are specific characteristics of legacy IS?

During their life cycle (Figure 9, p. 23) IS are evaluated (Berghout & Nijland, 2002). Different viewpoints on legacy IS have been suggested e.g. developmental; operational; organizational and strategic (Alderson & Shah, 1999). In order to further explore different viewpoints in Case A the following SRQ 3 was defined:

- SRQ 3: Which perspectives can be identified regarding the evaluation of legacy IS?

Swanson and Dans (2000) wrote that organizations should regularly conduct "health" assessments of individual systems to identify legacy IS. However, the problem was that organizations rarely evaluate IS quality adequately (Larsen, 1998). To gain more knowledge on the evaluation of legacy IS in practice the following SRQ 4 was defined:

- SRQ 4: How are legacy IS evaluated in practice?

Methods on how to identify legacy IS are provided by several authors, these all start by having an IS asset repository. The IS asset repository can be a list of IS, including IS (legacy) characteristics. Examples are provided by Benson et al. (2004); Gottling and Torgnysdotter (2002) and Warren (1999). In order to gain more knowledge concerning the identification of legacy IS in Case A, the following SRQ 5 was defined:

- SRQ 5: How can the identification of legacy IS be efficiently organized?

The next life cycle stage of a legacy IS is: (2) *the abandonment decision making process of legacy IS* (Berghout & Nijland, 2002) (see Figure 11, p. 24). After an IS is labeled as legacy IS, the decision has to be made whether or not to abandon the legacy IS (see Figure 13, p. 35). Therefore the second associated research question was:

Research question 2:
How do organizations decide to abandon legacy IS?

In order to answer this research question, the following five SRQ were proposed.

In the literature, suggestions are made why organizations abandon legacy IS; several authors provide different reasons (Adolph, 1996; Chan et al., 1996; Furneaux & Wade, 2011; Gode et al., 1990; Kelly et al., 1999; Richmond et al., 2006; Sakthivel, 1994; Swanson & Dans, 2000; Zvegintzov, 1984) (see Section 3.5, p. 33). The information seems to be fragmented and no overall view concerning the abandonment triggers of legacy IS was found in the literature. Due to the fact that only twelve papers were found (see Section 3.3, p. 26) and the interesting and important question: *"how do managers ultimately make the retirement and replacement decision?"* remains unanswered (Swanson & Dans, 2000), the following SRQ 6 and SRQ 7 were defined:

- SRQ 6: How is the decision making process to abandon legacy IS organized in practice?
- SRQ 7: What are triggers to abandon legacy IS?

From the literature it was concluded that there seems to be limited knowledge concerning the role and responsibility of stakeholders in the decision to abandon legacy IS (Swanson & Dans, 2000). To gain more

knowledge concerning the participation and the responsibilities in the abandonment decision making of legacy IS in Case A, the following SRQ 8 and SRQ 9 were defined:

- SRQ 8: Which stakeholders participate in abandonment decision making of legacy IS?
- SRQ 9: Who is held responsible for the abandonment decision of the legacy IS?

Good practices concerning the legacy IS abandonment decision making process were hardly found in the literature (Swanson & Dans, 2000). In order to further explore good practices in Case A the following SRQ 10 was defined:

- SRQ 10: What are good practices for legacy IS abandonment decision making?

According to the decision making process described by Harrison (1999) (Figure 3, p. 9), after the decision to abandon a legacy IS is taken the decision should be implemented and realized. Therefore the next and final life cycle stage identified is: (3) *the practical abandoning process of legacy IS* (Berghout & Nijland, 2002). Due to the fact that the literature concerning the practical abandoning is scant (see Section 3.9, p. 41), it is concluded that the theory concerning the practical abandoning of legacy IS should be improved. The third associated research question was:

Research question 3:
How do organizations practically abandon legacy IS?

In order to answer this research question, the following three SRQ were proposed.

Research concerning the practical abandoning process (also named disposal phase) remains scant (Rittinghouse, 2004; White, 2007). To gain more knowledge about the practical abandoning process in Case A, the following SRQ 11 was defined:

- SRQ 11: How are legacy IS practically abandoned?

The next question refers to the stakeholders and their responsibilities, Swanson and Dans (2000) state that little is known about who should be hold responsible for the practical abandoning of legacy IS. To gain more knowledge concerning the participation and the responsibilities of stakeholders in the practical abandoning of legacy IS in Case A, the following SRQ 12 was defined:

- SRQ 12: Who is held responsible for practically abandoning legacy IS?

Good practices concerning the legacy IS practical abandoning process were hardly found in the literature (Rittinghouse, 2004; Swanson & Dans, 2000; White, 2007). In order to further explore good practices in Case A the following SRQ 13 was defined:

- SRQ 13: What are good practices for practically abandoning legacy IS?

In summary each predominant life cycle stage includes an associated research question, this is illustrated in Table 8.

Life cycle stage:	Research question:
1) The aging process of legacy IS.	1) How do legacy IS age?
2) The abandonment decision making process of legacy IS.	2) How do organizations decide to abandon legacy IS?
3) The practical abandoning process of legacy IS.	3) How do organizations practically abandon legacy IS?

Table 8. Predominant life cycle stages and associated research questions

Subsequently, the research questions and SRQ have been used to identify the case study questions, which are included in Appendix A (p. 173).

The first section of this explorative research model described the three life cycle stages (1), *the aging process of legacy IS*, (2) *the abandonment decision making process of legacy IS*, and (3) *the practical abandoning process of legacy IS*. Together they will provide input for the second section of this research model which is to (4) *propose a method to abandon legacy IS*. For example answering SRQ 13: "What are good practices for practically abandoning legacy IS?", will provide input for such a practice guide.

5.3 Exploratory case study protocol

A Case Study Protocol (CSP) is required to increase the reliability of case study research and guide researchers in their data collection (Yin, 2003). The CSP should include the following sections: (1) Overview of the case study project, (2) Field procedures, (3) Case study questions and (4) Guiding section (Yin, 2003). The CSP is a formal and detailed master plan that specifies the research (Remenyi et al., 2005). Concerning evidence collection: *"whatever research strategy and tactics are used, the quality of the evidence is improved if the research is well planned and designed"* (Remenyi et al., 2005).

Based on the exploratory research model (see Section 5.2, p. 53), a CSP including a questionnaire was designed. In the CSP the questionnaire was described, and how possible answers could be interpreted. Furthermore the method of analyses was described. The questionnaire was discussed with the dissertation advisor and there was an initial test interview. The design of the exploratory case study questions is provided in Appendix A.1 (p. 173). Also in this section the explorative analysis model of the interviews is described (Appendix A.2, p. 177). The analysis model should provide insight into the relation between the primary research question, the research questions, sub-research questions and questions in the questionnaire. All interviews were recorded and transcribed. To avoid seeking conformance the CSP described that the interviewer had to have a critical reflection and had to do further probing of issues. The transcriptions were returned to the interviewees to allow adding or changing information. The recorded and transcribed audio files were time stamped and saved in the case study database. This was also the case for additional documents that were provided by the interviewees or identified through desk research.

5.4 Analyzing the interview transcripts

Conform the CSP, all interviews were recorded. Through "snowballing" each interviewee could identify subsequent interviewees. The research strategy was to continue interviewing until no new information was collected. In total 36 people were interviewed. The transcriptions were sent back to the interviewees for approval. Analyses were performed by using ATLAS.ti qualitative data analysis software.

Qualitative research distinguishes "first cycle coding" and "second cycle coding" methods (Saldana, 2009). Within the first cycle coding method, structural coding is done. Structural coding is the appropriate coding method when there are multiple participants and there is a semi-structured data gathering protocol (Saldana, 2009). Within the second cycle coding method the primary goal is to develop a sense of categorical, thematic, conceptual, and/or theoretical organization from an array of first cycle codes (Saldana, 2009). Axial coding and theoretical coding is appropriate within second cycle coding (Saldana, 2009). Axial coding extends the analytic work from the initial structural coding, the purpose is to relate categories to subcategories and specify the dimensions of each category (Saldana, 2009). Subsequently, theoretical coding functions like an umbrella that covers and accounts for all other codes and categories. In theoretical coding all categories and subcategories become systematically linked with the central core category, the one that has the greatest explanatory relevance for the phenomenon (Saldana, 2009). In the first cycle coding, all the transcribed interviews were loaded in ATLAS.ti; then codes based on the questionnaire and the sub-research questions were defined and were structurally coded on the data, see Table 9.

Codes	Related questions in questionnaire (Appendix A, p. 173).
Q1 Architect	Q 1
Q1 Finance manager	Q 1
Q1 Business manager	Q 1
Q1 Manager other disciplines (such as HRM, Legal, …)	Q 1
Q1 Supplier	Q 1
Q1 User	Q 1
Q1 High level manager	Q 1
Q1 Middle level manager	Q 1
Q1 Low level manager	Q 1
Q10 Personal triggers to abandon legacy IS	Q10
Q11 Organization triggers to abandon legacy IS	Q11
Q18 Stakeholders of legacy IS	Q18
Q19 Stakeholders involved in decision making of legacy IS	Q19
Q21(1) Success criteria legacy IS abandonment decision making	Q21
Q21(2) Failures legacy IS abandonment decision making	Q21
Q38 Critical incidents in abandoning legacy IS	Q38
SRQ 1 How are legacy IS defined in practice?	Q2-Q9
SRQ 2 What are specific characteristics of legacy IS?	Q2-Q9
SRQ 3 Which perspectives can be identified regarding the evaluation of legacy IS?	Q2-Q9
SRQ 4 How are legacy IS evaluated in practice?	Q2-Q9
SRQ 5 How can the identification of legacy IS be efficiently organized?	Q2-Q9
SRQ 6 How is the decision making process to abandon legacy IS organized in practice?	Q10-Q22
SRQ 7 What are triggers to abandon legacy IS?	Q10-Q22
SRQ 8 Which stakeholders participate in abandonment decision making of legacy IS?	Q10-Q22
SRQ 9 Who is held responsible for the abandonment decision of the legacy IS?	Q10-Q22
SRQ 10 What are good practices for legacy IS abandonment decision making?	Q21
SRQ 11 How are legacy IS practically abandoned?	Q23-Q39
SRQ 12 Who is held responsible for practically abandoning legacy IS?	Q23-Q39
SRQ 13 What are good practices for practically abandoning legacy IS?	Q26-Q39

Table 9. Codes based on answering the questions in the questionnaire and the sub-research questions

After the initial structural coding, the second cycle coding method started within ATLAS.ti by means of axial coding. The answers to the sub-research questions were categorized and further analyzed, also the additional information (e.g. received documents) from the trails of research were added (for triangulation purposes). Based on this analysis, the answers (results) to the sub-research questions within the processes were described. Coding with a tool like ATLAS.ti made it possible to identify additional relations. For example, the question "What are the triggers for the organization when to abandon legacy IS?" (Q11), was answered by interviewees with triggers to abandon a legacy IS. However, these interviewees also answered with "countervailing triggers" not to abandon a legacy IS. The initial structural code: "Q11 Organizational triggers to abandon legacy IS" was decomposed into: "Q11(1) Organization triggers to abandon a legacy IS" and "Q11(2) Organization triggers not to abandon a legacy IS". Interviewees were categorized in functions as performed in stakeholder analyses. The results are presented in Table 10, which provides an overview of 36 transcribed interviewees and their stakeholder role. This research covers all typical stakeholder roles, when further in this dissertation quotes of interviewees are referred to, the role and the interviewee number will be provided. For example Business information manager (1) said: "..."; Architect (2) said: "...".

Function	Number of interviewees	Management level	Department Case A
CIO/ICT director (1)	1	H	Other
Architect (1, 2, 3, 4 & 5)	5	M	Other
Business information manager (1, 2, 3, 4, 5, 6, 7 & 8)	8	M	WW, Other
Abandonment and migration project manager (1, 2, 3, 4 & 5)	5	H, M	A&M
IS owner (1) (e.g. director, business unit director)	1	H	WW, including G&E
Functional maintenance manager (1 & 2)	2	M	WW, Other
Technical maintenance manager (1)	1	M	Other
Business operations manager (1)	1	M	WW
Finance director (1)	1	H	Other
ICT controller (1)	1	M	ICT Control dept.
Renewal programme manager (1)	1	M	Other
Senior user (1 & 2)	2	L	Other, G&E
Consolidation project manager (1 & 2)	2	M	WW, Other
Supplier and contract manager (1)	1	M	Other
Consolidation programme manager (1 & 2)	2	H	WW, Other
IS merger expert (1) (worked at Case A consolidation)	1	M	WW
Account manager technology and services supplier of Case A (1)	1	M	External supplier

Table 10. Interviewed stakeholders, between parentheses their unique number, which will be used for quotations

5.5 The aging process of legacy information systems

In the predominant life cycle stage: "*the aging process of legacy IS*", the first research question is: "*how do legacy IS age?*" The results of Case A are subsequently discussed following the sub-research questions (SRQ), defined in Section 5.2 (p. 53):

- SRQ 1: How are legacy IS defined in practice?
- SRQ 2: What are specific characteristics of legacy IS?
- SRQ 3: Which perspectives can be identified regarding the evaluation of legacy IS?
- SRQ 4: How are legacy IS evaluated in practice?
- SRQ 5: How can the identification of legacy IS be efficiently organized?

SRQ 1: How are legacy IS defined in practice?

Within first cycle coding the legacy IS definitions were structurally coded. Based on the analyses of the interview questionnaires (the structural coding) it can be concluded that there were many diverse opinions about the definition of a legacy IS. Based on axial coding within the answers of SRQ 1 it was found that most "technical" interviewees (e.g. architects or a Business information manager with a background in COBOL programming) consider any IS a legacy IS, starting day one in production. Alderson and Shah (1999) define this viewpoint as the developmental viewpoint. Axial coding also indicated that about 75% of the interviewees mentioned that the word: "*legacy*" in legacy IS had a negative connotation; 15% of the interviewees know the word but were indifferent about its connotation. Some 5% did not know the word legacy at all and 5% had a positive connotation with legacy IS. Although most of the interviewees mentioned that legacy IS had a negative connotation, about 50% did not agree with this connotation. Architect (1) said: "*most legacy IS do their job*". Architect (3) said: *"IS designed today will automatically be tomorrow's legacy IS".*

In Case A, no formal definition of a legacy IS was available. Architect (1) said: *"the term legacy IS is not formally defined in this organization, however, it is often used and most people have a negative connotation*". Interviewees in Case A defined legacy IS differently, for example, Consolidation project manager (2) said: *"these are old systems; which are difficult to maintain have inadequate documentation and the remaining knowledge of the system in the organization was limited".* Architect (1) contradicts that: *"every operational system is a legacy IS; there was no formal or generally accepted definition of a legacy IS within Case A; although it is aged, COBOL is still a very good programming language for transactional systems"*. Business information manager (4) said: "*legacy IS are systems which are not developed by us, they do not fit into our technical architecture, it is*

literally something you inherited, most of the time it has a negative connotation, it is something which is poorly maintainable, unstable, it burdens a risk at the organization, it is difficult to interface with other systems and therefore you want to abandon the legacy IS". According to Architect (1): "*legacy is very quickly translated into technical aging, it is a mainframe, so it is legacy and it is built in the 70s, so it is legacy; however, I do not agree with this*".

According to interviewees, external circumstances or changing organization requirements will often make an IS a legacy IS, for example: redundant systems supporting similar functionality due to the fact of a merger/consolidation (happened in Case A), bankruptcy of a technical supplier or support that was stopped by a supplier (was mentioned several times). Something new/better comes along (which might also be outsourcing). It was not possible within this research to gain enough detailed quantitative information about the approximately 670 applications and consequently identify quantitative relations between abandoned applications and their age, language, or data structure. In summary it can be stated that most interviewees articulate that legacy IS are not defined by age, language and platform or data structure type. Although no formal definition could be identified in the organization, the findings in Case A support the earlier definition of a legacy IS being an IS that significantly resists meeting organizations' requirements.

SRQ 2: What are specific characteristics of legacy IS?
Within first cycle coding the specific characteristics of legacy IS were structurally coded (635 specific characteristics of legacy IS were coded). Subsequently, in second cycle coding, the specific characteristics of legacy IS were axially coded (codes were: IS-related legacy characteristics, application-related legacy characteristics, hardware-related legacy characteristics, data-related legacy characteristics, people-related legacy characteristics and procedure-related legacy characteristics). Again, each individual legacy characteristic was further axially coded (third cycle coding) resulting in a list of unique legacy characteristics of IS and its components: applications, hardware, data sets, people and procedures (Brussaard & Tas, 1980).

The aim of this case study research was to identify as many unique legacy characteristics at an elementary level as possible. Frequency or relative importance of the characteristics was not considered here. In order to improve the readability of this dissertation, a detailed overview of the legacy characteristics has been moved to Appendix C (p. 185). Legacy characteristics of IS and the individual components (application, hardware, data sets, people and procedures) and quotes from interviewees are described below.

IS-related legacy characteristics were provided by interviewees. For example, Business information manager (6) said: *"these systems are from another age, they use old technology, old human machine interfaces, these systems are difficult to change and not agile anymore".* Architect (4) said: "*the legacy IS does not meet the organization requirements, ... does not fit in the current business process, … is not conform the architecture, ... have security issues".* Business information manager (3) said: "*the legacy IS are so complex that we do not dare to touch or maintain them"*. The IS owner (1) said: "*for me it was even difficult to understand the architecture drawings. Remember that 109 systems were connected and all had their own difficulties and problems. ..., even the technical staff that should explain it to me, had difficulties to understand the architecture ..., finally, I knew how the 109 systems interfaced*".

In second cycle coding, 302 IS-related legacy characteristics were coded. By means of third cycle coding, 32 unique IS-related legacy characteristics were identified. From the literature research (see Section 3.4, p. 30), already 25 IS-related legacy characteristics were identified. In the literature research and Case A research 15 similar IS-related legacy characteristics were identified. Furthermore 10 IS-related legacy characteristics were identified in the literature, which were not found in Case A. In Case A, 17 IS-related legacy characteristics were identified, which were not mentioned in the literature. In total 42 unique IS-related related legacy characteristics were identified, these are described in Table 11.

By coding the interview data another result emerged. Architect (2) said: *"a supplier can go bankrupt, this is something out of control of the organization, and you cannot prevent this".* Based on Sakthivel (1994), see

Section 3.7.3 (p. 38) and Brooke and Ramage (2001) and Gold (1998) and the results of Case A, the concept of *internal* and *external* aging factors were introduced in this dissertation. This meant that legacy IS characteristics were categorized into internal aging factors, which were within the boundaries of the system and could be influenced by organizations, and external aging factors, which were outside the boundaries of the system and could not be influenced by an organization. Internal aging factors were, for example, dead code or incomplete documentation. External aging factors were, for example, opportunities arising from new technology. This distinction could be important because internal aging will always occur and external aging factors could be more decisive for legacy decisions. Internal and external aging factors could also interact.

Table 11 provides an overview of the 42 IS-related legacy characteristics in the literature and in Case A. It provides the number of references as identified in the literature and is marked "X" when the IS-related legacy characteristics was identified in Case A. Furthermore these legacy characteristics were categorized into internal or external aging factors. These legacy characteristics are described in detail in Appendix C (p. 185).

No.	IS-related legacy characteristics.	Literature # sources	Case A	(I)nternal (E)xternal
1	Limited architectural fit.	1	X	I
2	Aging legacy architectures.	2	X	I/E
3	Systems designed in stove pipe fashion or silos.	4	X	I
4	Legacy IS may hinder progress (not agile).	1	X	I
5	Legacy IS as a barrier to strategic innovation.	1		I/E
6	The documentation becomes increasingly inaccurate.	1		I
7	Inadequate documentation of the system.	2	X	I
8	Understanding of system details is often lacking.	2	X	I
9	System skills hard to find.	2	X	I/E
10	Diminishing support from the supplier.	1	X	E
11	Difficult, if not impossible, to extend.	1		I
12	Increasing complexity.	3	X	I
13	Declining system reliability.	1	X	I
14	Inadequate performance.	1		I
15	Inadequate interoperability, ambiguous interfaces.	3	X	I
16	Lack of integration of systems, dispersed landscape.	1	X	I
17	No longer meeting the needs of the environment.	1		I/E
18	Significantly resisting modification and evolution in order to meet new and constantly changing business requirements.	3		I
19	Growing gap between capabilities of the system and the needs of the business in which it is used.	5		I/E
20	Increasing security issues.	1	X	I/E
21	Benefits of system are lower than costs.	1	X	I
22	Risks.		X	I
23	Prevent organizations from fully exploiting computers and the value they bring in organizations.	1		I
24	Legacy portfolios are 30-50% more expensive.	1		I
25	Excessive system support costs.	1		I
26	Expensive to maintain and operate, high IS cost.	2	X	I/E
27	No cost awareness of users of the IS.		X	I
28	IS costs and IS benefits are unknown.		X	I
29	Fewer customers helped by more expensive IS, high cost per user.		X	I/E
30	Cheaper and/or better alternative.		X	I/E
31	There is no cost transparency resulting in worse decision making.		X	I
32	Similar or redundant IS capabilities (e.g. due to a merger).		X	I
33	A poor quality of the IS.		X	I
34	Lack of maintenance/dis-investment for many years on the IS.		X	I
35	Unstable IS.		X	I
36	Workarounds on legacy IS, unintended use of the IS.		X	I
37	The capabilities provided by the IS are less used, less needed or no longer needed anymore by the users or the customers.		X	I/E
38	Non-scalability, the IS is not capable to meet rising demands.		X	I/E
39	Lots of manual interventions.		X	I
40	(In an outsourced situation) legacy IS characteristics are: vendor lock-in, dependency on a vendor, resulting in optimization from a vendor perspective.		X	E
41	IS contracted as part of lumpsum with suppliers or that include "exit" costs of abandoning an IS.		X	E
42	Individual IS which constitutes different contracts with different suppliers.		X	I/E

Table 11. Overview of the IS-related legacy characteristics as found in the literature and identified in Case A

Besides IS-related legacy characteristics, also individual IS component-related legacy characteristics were coded. Application-related legacy characteristics, were provided by interviewees. For example, Business information manager (2) said: *"legacy application software has a structural lack of maintenance, an application is similar to*

a car, if you stop maintenance, you do not put oil in your engine, you know your car will break down one day, it can be tomorrow, or over a year, but it will definitely stop one day". Business information manager (5) gave the following analogy concerning changing the user or customer perception: "*decades ago nobody had an air conditioning in their car and nobody missed it, nowadays a car without an air conditioning makes it a legacy car".* Business information manager (4) said: "*in time, the software became spaghetti, therefore, maintenance will become more and more difficult and also more expensive".* Business information manager (4) said: *"this week we had a problem with an application, there is hardly knowledge of this application, the supplier stopped support and I cannot find anybody to help".*

In second cycle coding, 210 application-related legacy characteristics were coded. In the third cycle, by means of axial coding, 38 were coded unique within Case A. From the literature research (see Section 3.4, p. 30), already 25 application-related legacy characteristics were identified. In the literature research and Case A research 17 similar application-related legacy characteristics were identified. Furthermore eight application-related legacy characteristics were identified in the literature, which were not found in Case A. In Case A, 21 application-related legacy characteristics were identified, which were not mentioned in the literature. In total 46 unique application-related related legacy characteristics were identified. These application-related legacy characteristics were categorized into internal and external aging factors.

Table 12 provides an overview of the 46 application-related legacy characteristics in the literature and in Case A. It provides the number of references as identified in the literature and is marked "X" when the application-related legacy characteristics was identified in Case A. These legacy characteristics are described in detail in Appendix C (p. 185).

No.	Application-related legacy characteristics.	Literature # sources	Case A	(I)nternal (E)xternal
1	Inadequate software quality.	15	X	I
2	Complexity of the software (increases).	6	X	I
3	Reduced or inadequate performance.	3		I
4	Decreasing reliability.	2		I
5	(Structural) lack of maintenance on the application.	1	X	I
6	Use of older program language, or technical opportunities available.	7		I
7	Supporting software: compilers no longer supported by hardware supplier.	2	X	E
8	Support from application supplier has ended.	1		E
9	No hardware support for the platform on which the application is running.	3	X	E
10	Software does not have an architectural fit.	1	X	I
11	Limited agility (changes take longer).	1	X	I
12	"Dying" application programme language.	1	X	E
13	Skills are hard to find, staff with knowledge skills concerning the application are retired or have left the organization.	6	X	I
14	Limited or missing documentation of the application.	6	X	I
15	Inadequate documentation of the application.	4	X	I
16	Software embedded in the hardware.	2		I
17	Embedded business logic.	1		I
18	Understanding of system details is often lacking.	3		I
19	Tracing faults within applications is costly and time consuming.	1		I
20	New functional requirements.	4	X	I
21	Functionality does not meet the requirements.	5	X	I
22	Changing user perception.	3	X	I/E
23	Alternatives/opportunity cost (a better or cheaper IS available).	2	X	E
24	Expensive to maintain.	9	X	I
25	Increasing maintenance costs of applications.	6	X	I
26	High application costs.		X	I/E
27	Source code not available.		X	I
28	No access to source code.		X	E
29	No knowledge of the application.		X	I
30	Supplier goes bankrupt.		X	E
31	The core functionality of a system which is static and very stable is surrounded by less stable and dynamic process related functionality.		X	I
32	Application is difficult to maintain.		X	I
33	No programmer skills available.		X	I/E
34	Lower versions of the software, not using latest release.		X	I
35	Software not developed with "computer aided software engineering" (CASE) tooling.		X	I
36	Not service oriented, software architecture is not service oriented.		X	I
37	Not latest design environment.		X	I
38	Applications are programmed in older generation languages.		X	I
39	Redundant (double or similar) functionality.		X	I
40	Growing functionality gap.		X	I
41	It is no longer possible to do major functionality changes as the business requires.		X	I
42	Functionality no longer needed, no users anymore, or no products anymore.		X	I
43	Functionality of an application becomes of lesser value.		X	I
44	Workarounds by users (e.g. functionality is incorrectly used).		X	I
45	Not meeting user expectations.		X	I
46	Lack of objective measures, like cost per function points, cost per release.		X	I

Table 12. Overview of the application-related legacy characteristics as found in the literature and identified in Case A

Hardware-related legacy characteristics were provided by interviewees. For example, Abandonment and migration project manager (1) said: *"now and then we find unstable systems, we will wash them (bring them to*

the latest operating system and replace them by newer and faster hardware) after the wash the system is more stable and faster". Abandonment and migration project manager (4) said: *"within a year the hardware is not supported anymore".*

In second cycle coding, 28 hardware-related legacy characteristics were coded. In the third cycle, by means of axial coding, 16 were coded unique within Case A. From the literature research (see Section 3.4, p. 30), already 10 hardware-related legacy characteristics were identified. In the literature research and Case A research seven similar hardware-related legacy characteristics were identified. Furthermore, three hardware-related legacy characteristics were identified in the literature, which were not found in Case A. In Case A, nine hardware-related legacy characteristics were identified, which were not mentioned in the literature. In total 19 unique hardware-related related legacy characteristics were identified. These hardware-related legacy characteristics were categorized into internal and external aging factors.

Table 13 provides an overview of the 19 hardware-related legacy characteristics in the literature and in Case A. It provides the number of references as identified in the literature and is marked "X" when the hardware-related legacy characteristics was identified in Case A. These legacy characteristics are described in detail in Appendix C (p. 185).

No.	**Hardware-related legacy characteristics.**	**Literature # sources**	**Case A**	**(I)nternal (E)xternal**
1	Inadequate performance or slow.	5	X	I
2	Poor reliability.	1	X	I
3	Lack of knowledge of old hardware.	1		I/E
4	Old hardware type.	3	X	I
5	Old platform type.	2	X	I
6	Legacy IS run on obsolete hardware.	2		I
7	Hardware supplier discontinues the product line.	1	X	E
8	Not compatible with current procurement policies.	1		I
9	Expensive to maintain.	4	X	I
10	Hardware embedded with software.	1	X	I
11	Inadequate stability.		X	I
12	Poor availability.		X	I
13	No architectural fit.		X	I
14	"Dying" platform.		X	E
15	Supplier bankrupt.		X	E
16	Four decentralized data centers instead of one centralized. Redundant data centers (not being fail over).		X	I
17	Alternative (better and/or cheaper) available (opportunity costs).		X	I/E
18	High hardware costs.		X	I
19	High exploitation costs.		X	I

Table 13. Overview of the hardware-related legacy characteristics as found in the literature and identified in Case A

Data-related legacy characteristics were provided by interviewees. For example, Consolidation project manager (2) said: *"we still have decentralized database on office buildings, these data is collected once a day and sent to another database".* Abandonment and migration project manager (1) said: *"another legacy characteristic is that we use under this important system an old CODASYL database".*

In second cycle coding, 21 data-related legacy characteristics were coded. In the third cycle, by means of axial coding, eight were coded unique within Case A. From the literature research (see Section 3.4, p. 30), already eight data-related legacy characteristics were identified. There appeared to be no overlap between the procedure-related legacy characteristics in the literature and Case A. This implies that overall 16 unique data-

related related legacy characteristics were identified. These data-related legacy characteristics were categorized into internal and external aging factors.

Table 14 provides an overview of the 16 data-related legacy characteristics in the literature and in Case A. It provides the number of references as identified in the literature and is marked "X" when the data-related legacy characteristics was identified in Case A. These legacy characteristics are described in detail in Appendix C (p. 185).

No.	**Data-related legacy characteristics.**	**Literature # sources**	**Case A**	**(I)nternal (E)xternal**
1	Immense volume of data has accumulated over the lifetime of the system. This data may be inconsistent and may be duplicated in several files.	1		I
2	Aging data architecture; aging database.	3		I
3	Data is defined and stored redundantly across multiple stovepipe business units and applications.	1		I
4	The same or similar data is defined inconsistently across multiple systems.	1		I
5	The same data terminology may be used to define different data across multiple applications and business units.	1		I
6	The integrity of the data may be poor and contain information it should not contain.	1		I
7	Data may not be easily accessible by modern systems or through user-based inquiries.	1		I
8	Data cannot be readily shared across systems, business units and organizational boundaries.	1		I
9	Corrupted data or unreliable data.		X	I
10	Redundant data, redundant data sources.		X	I
11	Old database, no support from database supplier.		X	E
12	Database not conform to the database architecture.		X	I
13	Security.		X	I
14	Data was included in the software as a file system, instead of in a relational database. The data is not stored in relational database, but in hierarchical databases or even flat files (being part of the software).		X	I
15	Performance and capacity of a database is not sufficient anymore.		X	I
16	Decentralized databases (each office has a decentralized database, data is collected once every day and sent to a data warehouse).		X	I

Table 14. Overview of the data-related legacy characteristics as found in the literature and identified in Case A

Procedure-related legacy characteristics were provided by interviewees. For example, Architect (3) said: *"a system was unstable, we had to upgrade the testing procedure documentation*". Abandonment and migration project manager (1) said: *"we had to document the testing procedures and management procedures of applications, these were documented in a specific manual".*

In second cycle coding, 20 procedure-related legacy characteristics were coded. In the third cycle, by means of axial coding, four were coded unique within Case A. From the literature research (see Section 3.4, p. 30), already two procedure-related legacy characteristics were identified. There appeared to be no overlap between the procedure-related legacy characteristics in the literature and Case A. This implies that overall 6 unique procedure-related legacy characteristics were identified. These procedure-related legacy characteristics were categorized into internal and external aging factors.

Table 15 provides an overview of the procedure-related legacy characteristics in the literature and in Case A. It provides the number of references as identified in the literature and is marked "X" when the procedure-related legacy characteristics was identified in Case A. These legacy characteristics are described in detail in Appendix C (p. 185).

No.	Procedure-related legacy characteristics.	Literature # sources	Case A	(I)nternal (E)xternal
1	Do not take advantage of streamlined facilities and standard routines; use programming and design practices that current management would not permit.	1		I
2	Definitions of how the business should be carried out and constraints on the business. Use of the legacy application system may be embedded in these policies and rules.	1		I
3	Inadequate support documentation.		X	I
4	Maintenance procedures outdated or not in place.		X	I
5	Outdated test procedures.		X	I
6	Internal hiccups in the procedures and processes of supporting the IS.		X	I

Table 15. Overview of the procedure-related legacy characteristics as found in literature and identified in Case A

Furthermore staff or people-related legacy characteristics were provided by interviewees. For example, Business information manager (2) said: *"abandoning an IS is abandoning the span of control, the company car, the status and the desk of a manager".* Business information manager (4) said: *"managers might have wrong incentives if the new IS might require less staff*". Architect (5) said: *"IS are like babies, you do not throw away your own baby".* Functional maintenance manager (1) said: *"the person responsible for this functionality left the organization almost immediately after hearing that the IS was going to be abandoned… it was a lot of work to transfer this functionality to another system".*

In second cycle coding, 54 people-related legacy characteristics were coded. In the third cycle, by means of axial coding, seven were coded unique within Case A. From the literature research (see Section 3.4, p. 30), already four people-related legacy characteristics were identified. There appeared to be no overlap between the people-related legacy characteristics in the literature and Case A. This implies that overall 11 unique people-related related legacy characteristics were identified. These people-related legacy characteristics were categorized into internal and external aging factors.

Table 16 provides an overview of the people-related legacy characteristics in the literature and in Case A. It provides the number of literature references and is marked "X" when the people-related legacy characteristics was identified in Case A. These legacy characteristics are described in detail in Appendix C (p. 185).

No.	People-related legacy characteristics.	Literature # sources	Case A	(I)nternal (E)xternal
1	Lacking technical expertise, due to retiring staff and skills which are hard to find.	3		I/E
2	Dislike of the tedious work on legacy IS, "if something is difficult, tedious and slow we will try to avoid doing it". Engineers prefer working on new system development instead of maintaining old software.	3		I
3	Changes made by people who do not understand the original design concept almost always cause the structure of a programme to degrade; after many such changes the original designers no longer understand the modified product.	1		I
4	Many people who are doing software development do not have an education appropriate to the job.	1		I
5	No knowledge of the IS available anymore (skills left the organization resulting in no knowledge of the IS anymore); application programme languages not taught at the university anymore.		X	I/E
6	Expertise is no longer available anymore or only available at high costs.		X	I/E
7	Expertise of the system is lacking, making the IS vulnerable, due to the fact that people left.		X	I
8	Knowledge and expertise only available in the heads of persons and not on paper makes an organization dependent on these persons (knowledge is power).		X	I
9	Expertise of the IS is splintered across different stakeholders within different business units in the organization.		X	I
10	IS professionals derive, power and status from IS, abandoning IS is abandoning power and status.		X	I
11	Myopia of stakeholders concerning their own IS, not open to something new, or different.		X	I

Table 16. Overview of the people-related legacy characteristics as found in the literature and identified in Case A

SRQ 3 Which perspectives can be identified regarding the evaluation of legacy IS?

Alderson and Shah (1999) discern four different viewpoints on legacy IS. They discern developmental, operational, organizational and strategical viewpoints. In Case A different stakeholders indeed employ different viewpoints. The architects considered a legacy IS to be any operational IS (e.g. Architect [3] said: *"IS designed today will automatically be tomorrow's legacy IS"*). An ICT manager with a background in finance and business processes distinguished all aspects. Overall, the stakeholders employed divergent perspectives. The architecture function within Case A indeed supported the developmental view as described by Alderson and Shah (1999). The other three viewpoints as described by Alderson and Shah (1999) were not distinguished. Because many interviewees had been working in the industry for many years, their professional background and career included various/multiple stakeholder perspectives, such as IS developer, followed by project manager, financial manager and information manager. Their answers were the accumulation of all acquired skill. This research was therefore inadequate to validate the viewpoints of Alderson and Shah (1999).

SRQ 4: How are legacy IS evaluated in practice?

Within first cycle coding the legacy IS evaluations in Case A were structurally coded. Then within second cycle coding by means of axial coding the following different evaluation practices were distinguished: an evaluation method at the inception of Case A (due to the merger), an evaluation method for the European tendering of data centers and an evaluation method for consequent operational legacy IS. These evaluation methods will be subsequently described.

At the inception of Case A, the IS within the WW domain were evaluated by the department responsible for architecture. The architects were selected from the former branch organizations and this team created a multicriteria scoring model to evaluate the different systems. The multicriteria model included the following aspects: production volume, functionality, quality and life expectancy, maintenance effort and cost, exploitation cost, reusability and flexibility, constraints and limitations and ownership (e.g. proprietary software, intellectual property or supplier ownership). Effort was made to make the evaluations objective. For instance, architects were only allowed to assess IS of other business domains. Furthermore, an independent consultancy firm was consulted to plot several major systems on business processes. This method illustrated redundancies or

shortcomings of the IS support to business processes. Another consultancy firm again reviewed the consolidation and suggested a consolidation strategy.

Other business units followed an approach similar compared to business unit WW. This implied that the efficiency of IS supporting business process was compared and scenarios were developed to merge, consolidate, or keep IS in parallel. These scenarios included risk analyses and assessments of required time. Business unit consolidation decisions were made at the inception of Case A. After these decisions were made the multicriteria scoring stopped. The particular merger strategy was subsequently deployed and there were no more formal multicriteria evaluations.

The next major evaluation practices concerned tendering of logical ICT domains such as telecom, client and peripherals (PCs and printers) and data centers. By means of dedicated multicriteria models for each ICT domain, suppliers were selected. For instance, when Case A tendered the data center domain in accordance with European laws, a multicriteria model was used to score all suppliers (for instance typical data center services such as cost per MIP, storage costs and redundancy). Based on this tender a new data center was selected and subsequently four data centers were abandoned. The order in which data centers were abandoned was based on the evaluation model, which is described in Section 4.4 (p. 47). Furthermore, with help of an external benchmark supplier, Case A regularly benchmarked all ICT domains. These benchmarks (e.g. cost per application function point development) were used for evaluation purposes.

The next major evaluation practice concerned the evaluation of legacy IS during operations. Before implementing new business functionality to support, for instance new laws, into the IS, a study was performed and the impact of this new law on the IS was evaluated and also communicated with the responsible politicians. This was called an impact study. Case A was always able to implement all new requirements. However, if the costs or risks were too high, Case A would advise politicians to consider particular changes.

Finally with the help of an external consultant, Case A performed a software code review on their largest legacy IS, in which software code was assessed on various quality aspects. This indicated that the software of the large legacy IS was well maintained.

It can be concluded that Case A regularly evaluated all their IS and also regularly evaluated particular IS. This was done by means of multicriteria methods, benchmarks or quality measurements, and performed by internal staff or external consultants or both.

SRQ 5: How can the identification of legacy IS be efficiently organized?
The process of identification of legacy IS was already developed before the merger (see Section 4.3, p. 46); however, concerning the post-consolidation phase, this process was subject to further development. A success factor for the identification of legacy IS was to actually have records of all applications, such as, names, core functionality and users. Architect (1) (responsible for the corporate information planning) said: *"based on a yearly process by means of portfolio management techniques the identification and managing of legacy IS should be performed. A team with business and technical representatives should perform application portfolio scoring on all applications in terms of functional and technical quality and subsequently should decide which legacy IS to abandon and which to continue".* The importance to score IS on business value was also stressed by other interviewees. For example, Business information manager (6) said: "*an IS designed for 400,000 clients, but serving only 4,000 clients might be extremely expensive*". Legacy IS of limited value should preferably be abandoned.

5.6 The abandonment decision making process of legacy information systems

In the predominant life cycle stage: "*the abandonment decision making process of legacy IS*", the second research question is: *"how do organizations decide to abandon legacy IS?"*

This section describes the decision making processes to abandon legacy IS in Case A; the related sub-research questions (SRQ) are addressed:

- SRQ 6: How is the decision making process to abandon legacy IS organized in practice?
- SRQ 7: What are triggers to abandon legacy IS?
- SRQ 8: Which stakeholders participate in abandonment decision making of legacy IS?
- SRQ 9: Who is held responsible for the abandonment decision of the legacy IS?
- SRQ 10: What are good practices for legacy IS abandonment decision making?

SRQ 6 How is the decision making process to abandon legacy IS organized in practice?
Based on the transcripts and information provided by additional desk research, the process of legacy IS abandonment decision making is described. The decision making process to abandon legacy IS is primarily done in Trail 1 by business managers (and not by the centralized ICT department). According to the most senior business manager of Business Unit WW (IS owner [1]), this is done based on rational arguments: "*everybody knew from the beginning that it was not possible to maintain six information functions from former branch organizations supporting similar laws, it was clear from the beginning that there would be one process, one way of working and one information function*". All scenarios have been evaluated. In the WW domain (Trail 1), it was decided to prefer the information function of the former organization 1, which contained the predominant target IS named "XYZ". Organization 1 also served the largest customer base. After this decision other information functions from former branch organizations containing many IS were merged and successively abandoned.

Due to the fact that Case A tendered for a new data center, an "Abandonment and Migration" programme (A&M programme) was initiated that facilitated the process of abandoning and migration of the Case A applications (Trail 2). A list of all applications was made and the Business information manager provided information on which applications to migrate to the new selected data center and which application to abandon. For some applications within the business there was already a "strategy". In mutual discussions with business managers and with the A&M programme members this list was further refined.

Chapter 4 described the longitudinal research of Case A. It was illustrated (Figure 22, p. 51) that the number of applications to be abandoned grows during the years. According to the interviewees and additional supporting documentation the following factors were found explaining this difference between the planning in 2006 and the actual practical abandoning in 2011. Due to the tendering of a new data center, senior management initiated a new programme called Abandonment and Migration (A&M). The goal of this programme was to abandon as many applications as possible and to migrate the other applications. Senior business management would make the final decision as to which applications would be abandoned and which migrated. Right after the acquisition of the new datacenter these business managers considered business continuity more important than the abandoning of applications and, therefore, first decided to abandon 260 applications and to keep 410 applications. One year later (2007), however, the number of applications to be abandoned increased to 353 applications. Furthermore, a considerable number of applications (70) that earlier were candidates for Abandon or Migrate was labeled unknown in 2007, meaning that the decision to Abandon or Migrate had to be taken again.

The following reasons explained the different numbers of abandoning legacy IS throughout the years. There were new insights in the usage of applications and in migration cost. Migration required substantial additional investments in documentation and interfacing. It was considered a waste of energy and money to migrate an application that would soon be abandoned. To migrate a single application from an old data center to a new data center was an investment, it required a lot of work and costed money. For example the application knowledge and documentation was inadequate, interfaces were unknown and knowledge and documentation needed to be improved, before it could be handed over to the new data center. If there was no replacement,

than the abandonment decision was faster realized. Exist cost were mostly unknown or unclear and required additional negotiating with suppliers.

Abandonment and migration project manager (5) said*: "I had the idea, that at the start of Case A, all former branch organizations were protecting their own cultural heritage IS. After a few years staff was triggered to look to other things, then a process of disengagement concerning the IS starts. This is human nature, it needs time, after some time importance of the system seems less, and it is easier to abandon your own IS, I see it as a mortgage process".*

There were also other reasons mentioned by interviewees to initially postpone the abandonment of a legacy IS. Abandonment and migration project manager (1) said: "*abandoning applications means abandoning people, also the colleague that helped you in the past, who you could call on Friday night with a problem and, who would help the same Friday night, which is difficult*". Business information manager (2) mentioned: "*abandoning a legacy IS is like abandoning your baby, you are not going to abandon your baby"*. Architect (1) mentioned: "*there is always somebody, who uses the IS now and then, and claims the IS is necessary, and might not have to pay the costs from his own pocket"*. Business information manager (1) said: "*if a manager has a certain span of control over people, abandoning a legacy IS is also abandoning his span of control".* Furthermore, the A&M programme offered assistance to migrate or abandon.

In this section the managerial decision making process in Trail 1 is further elaborated. It will focus on decision making, which is described in Section 4.3 (p. 46), because Trail 2 is primarily focused on operational decision making within the A&M programme during the case study period.

According to the interviewees and grounded in the data of Case A, several hierarchies of disparate decision making of abandoning legacy IS emerged. At the highest level, which was not within the scope of this research, it was decided by the government and enforced by law (legal compliance) to incept Case A and to consolidate the six former branch organizations. In this research, from Case A study data, several hierarchies of decision making were identified. Decision making was found at information function (e.g. merger of six information functions into one information function) at IS Level and even at application level. To describe this phenomenon Harrison's (1999) managerial decision making process is used.

Harrison (1999) describes the managerial decision making processes as dynamic and interrelated. His managerial decision making process is illustrated in Figure 3 (p. 8). His model is used to illustrate the different levels of abstraction in research Trail 1 as found in the case organization. Within Trail 1 (see Figure 19, p. 47), three[9] functional abstraction levels for decision making were found: (1) the information function level, (2) the IS level and (3) the application Level. The information function contains several IS and one IS again contains several applications. These levels also interact with permeable boundaries (Harrison, 1999). Level 1 concerns decision making regarding the consolidation of the information function within the business unit WW, Level 2 concerns decision making of IS within the G&E information function and Level 3 concerns the individual applications. In Table 17 the decision making process based on Harrison (1999) as found within the three abstraction levels of Case A is described.

[9] A fourth higher hierarchical decision level to abandon legacy IS was distinguished. It was a political decision to incept Case A and to consolidate the six former organizations. This political decision making, ratified by law, is outside the scope of this research.

Functions of decision making process (Harrison, 1999)	**Information function (Level 1)**	**Information system (Level 2)**	**Application (Level 3)**
Setting managerial objectives	After inception of Case A in 2002 it was decided to merge from six information functions to one information function and to have one standardized working process within the business unit of WW.	The abandonment of the IS within the G&E information function. The G&E information function consisted of 14 applications, including hardware, interfaces, people (information management), of which the main application was ABC. The ABC IS (application, hardware, data sets, people and procedures) should be abandoned and the new IS should be the SWU/XYZ IS.	Abandon all 14 applications, migrate data, abandon working process, lay off staff, abandon hardware, abandon contracts with suppliers, abandon the applications.
Searching for alternatives	In the case organization alternatives were worked out to go from six information functions to one information function, which sequence to go from six to one, what was the possible target information function what were the costs. It is documented that first an inventory was made by architects of Case A, that a study was performed by a large and well-established consultancy firm. After that a smaller consultancy organization did a study and impact analyses were performed. There were different alternatives to consolidate, like a big bang scenario or successively and also the order of consolidation was elaborated in advance. Different alternatives were described, compared and evaluated.	In May 2007 a study was performed how the ABC IS should be abandoned.	All interfaces were identified in detail and ways to abandon the applications were explored.
Comparing and evaluating alternatives	During a three day session in November 2002, the alternatives were evaluated and the consolidation strategy was selected.	In June 2007 it was investigated whether all applications should be abandoned at once, or sequential.	Different scenarios were evaluated.
The act of choice	It was decided to opt for one information function and to successively merge the other information functions, starting with organization 2, followed by organization 3, organization 4, organization 5 and organization 6.	On October 2007 the board of directors approved the plan to abandon the IS within the G&E. The alternatives were compared and it was decided to first do a certain population and later another population.	A choice on how and when to abandon the applications was made.
Implementing decisions	This decision was implemented by a small consultancy firm. Consolidation activities were performed and information functions from former branch organizations were abandoned.	This decision was implemented.	This decision was implemented.
Follow up and control	There was programme planning and the highest business unit manager reported to the board of the organization.	The ABC IS was abandoned on November 2008, all applications were abandoned, hardware was abandoned, the working process was abandoned and replaced by another working process, staff was laid off or trained to	On November 2008 all 14 applications were practically abandoned.

		another system, and data was migrated. The implementation of abandoning the application was handed over to the A&M programme.	

Table 17. Legacy IS abandonment decision making on disparate functional levels

Based on the three decision making levels and the fact that these decision making levels interact with each other, it can be concluded that from a managerial perspective abandoning legacy IS is extremely complex.

SRQ 7 What are triggers to abandon legacy IS?

The aim of this case study research was to identify as many unique triggers to abandon legacy IS at an elementary level as possible. Frequency or relative importance of the triggers was not considered here. Interviewees were asked for reasons to abandon legacy IS. These reasons to abandon legacy IS were referred to as abandonment triggers. By means of first cycle coding, all abandonment triggers were structurally coded "abandonment triggers". This resulted in 95 abandonment triggers including references to the interviewees. Subsequently, in the second coding cycle all similar IS abandonment triggers were axial coded. This resulted in 25 unique abandonment triggers. These triggers were further categorized into IS, applications, hardware, data sets, people and procedures (Brussaard & Tas, 1980). This categorization is used because the staff of a data center department will mention other abandonment triggers than the staff of the application department. Finally, based on Sakthivel (1994), see Section 3.7.3 (p. 38) and Brooke and Ramage (2001) and Gold (1998) internal and external abandonment triggers were distinguished. This distinction is made to assess how much influence organizations have on triggers to abandon IS, because external triggers remain outside the organization. Table 18 illustrates the 25 unique IS abandonment triggers which were identified by interviewees in Case A. The first column provides the name of the identified abandonment trigger, the second column provides additional information by means of a summary of the interviews made by the researcher or original quotes from interviewees, the following columns describes if the triggers were related to the IS as a whole, or individual components such as applications, hardware, data sets, people and procedures, an "X" means that the trigger was found. The last column describes whether the abandonment trigger is categorized internally or externally rooted.

Triggers to abandon.	**Remarks.**	IS	Applications	Hardware	Data sets	People	Procedures	(I)nternal/(E)xternal
Does not fit in technological strategy.	Architect (5): *"the IS does not fit the technological strategy, it is not conform the architectural guidelines".*		X	X	X			I
Instability of the IS.	Business information manager (4): *"instability of the IS due to: technical changes, a platform which is unstable".*		X	X				I
Complexity.	Business information manager (4): "*by maintaining the IS, the IS is getting more complex and therefore maintenance will become more difficult and more expensive"; "it is like spaghetti".* CIO/ICT director (1): *"the complexity is uncontrollable".*		X	X				I
Non-scalability of the IS.	Demand is rising, however due to technical constraints it is not possible to supply this demand. A large IS can handle the small additional volumes relatively easily, while a small IS might have difficulties to handle large amounts of extra volume, or due to a merger extra volume. Architect (3): *"there was a small IS with superior functionality, it was however in the merger not scalable to supply whole demand of Case A".*		X	X	X			I
Limited or ended supplier support.	Senior user (1): *"supplier stops services (end of service) or is bankrupt".* This happens for example with a Bull mainframe. Business information manager (3): *"a system runs on an operating system were no hardware spare parts are available anymore".*		X	X				E
Technical aging.	No upgrades available anymore. No support for the assembler anymore. Outdated procedures. Consolidation project manager (1): "*a supplier which does not support an operating system, or application software or hardware anymore".*		X	X	X		X	E
The IS is non-agile.	Consolidation project manager (1): *"the organization is not able to quickly adapt to changing environmental circumstances".*	X						I
Structural lack of IS maintenance.	A lack of structurally maintaining the system will result in the trigger to abandon the IS. Business information manager (2): "*legacy application software has a structural lack of maintenance, an application is similar to a car, if you stop maintenance, you do not put oil in your engine, you know your car will break down one day, it can be tomorrow, or over a year, but it will definitely stop one day".*		X	X				I
Vendor dependency / vendor lock-in.	Business information manager (1): *"a strategic IS, was built and maintained by a supplier, there was a vendor lock-in, that was an unfavorable situation and the IS was abandoned and changed by an own IS (e.g. there was a pension IS within a former Case A branch organization which was functioning very well. However, it was built and maintained by an external supplier, by higher management it was decided that this was not a favorable situation and the pension system was rebuilt by the former Case A branch organization and the other IS was abandoned)".*	X						E
Effort to migrate applications from an old data center to a new data center.	Abandonment and migration project manager (2): *"due to the fact that Case A selected a new data center, for each application a decision had to be taken, to abandon or to migrate the application from a former data center to a new data center".* Migrating is a lot of work so it makes it more attractive to abandon a legacy IS. Abandonment and migration project manager (1): *"migration of applications from current data center to a new data center, it takes a lot of effort and investments to do this, the amount of work, makes it less attractive to keep the IS. It can help to make the abandonment decision of a legacy IS, it was named migration pressure".* This happened in Case A, in which four data centers were abandoned and centralized to a new data center (these were data centers of the former branch organizations).		X					I
Redundant data centers.	From four former data centers to one new selected data center (four data centers were abandoned in Case A).			X				I
Redundant data sources.	Business information manager (4): *"we had three data sources, which was an undesirable situation, we could not continue like that".*				X			I

Merger or consolidation.	Due to the fact that if two or more common organizations merge, having the same services and products a decision has to be made to abandon one or more IS because there is similar or double functionality.	X						I
New technological opportunities.	New technological opportunities, new technology available, opportunities. Architect (5): *"newer technologies for lower prices, for example: server client technology or database technology or more memory availability"*.		X	X	X			E
Redundant application functionality.	Similar or double application functionality within an organization. Abandonment and migration project manager (5): *"in this organization we had many similar systems for hour registration, it is not a matter to identify if you are missing functionality, and it is a matter of just choosing one".*		X					I
Application functionality gap/not sufficient.	The speed of changes or the changes are so fast, that it is not possible to follow the demanded functionality. Architect (3): *"the changes can no longer follow the required products or services which should be delivered by the IS"*. Functionality not sufficient. The gap between required functionality and delivered functionality is so big that the IS is abandoned.		X					I
Perception of the required IS capability is changed.	Changing user perception, users nowadays have computers at home, their perception of an IS is changing. For example: IS with monochrome displays and without mouse functionality. Perception of the needed capability is changed (by organization or by users or by customers). An analogy concerning changing perception was provided by Business information manager (3): *"20 years ago almost nobody had a car with air-conditioning, nowadays nobody wants to have a car without air-conditioning".*	X						I
Abandoning products or services supported by the IS.	No justification of the IS anymore, another positioning of the Case A services. For example if an IS supports a working process which delivers five products or services and three services are abandoned, the IS might not longer be justifiable for the remaining two products or services.	X						I
Capability not needed.	Architect (3): *"the legacy IS was abandoned when the capability of the legacy IS was not needed (e.g. product or service delivered by the IS is discontinued)"*. For example by law enforcement a product line is discontinued.	X						E
New demands from the market which cannot be provided by the IS.	If clients expect another service this might be a trigger to abandon a legacy IS (e.g. clients use to correspond in the past with Case A with letters and Case A scanned this letters into an IS. Nowadays clients are communicating by email or phone and the IS providing the scanning functionality should be abandoned). Case A is working more and more in supply chains, there is the: "National Reference Architecture", it describes references and principles on behalf of the government. It is used to ensure that several different Government bodies are able to communicate and share information with each other. In case the data structure of an IS, or the communication protocol is not according to the National Reference Architecture and not working in the information chain the legacy IS might be abandoned. New demands from the market (external market environment) which could not be provisioned by the current IS, changing external market environment.	X						E
Cost-benefit.	No business value anymore. IS owner (1): *"it is a rational decision based on: cost, benefit and return on investment"*.	X						I / E
High IS cost.	Abandonment and migration project manager (2): *"every legacy IS not abandoned, will consume scarce resources"*. Business information manager (4): *"growing marginal maintenance cost due to the fact that an IS becomes more complex, opportunity costs".*	X						I / E
High cost per user or customer.	High cost per user or customer. Only supporting a few users (or no users while costs are fixed). Senior user (1): *"if there are only 10 people working with an IS, I can imagine that the IS is abandoned*". Business information manager (6): *"the legacy IS is built for 400,000 customers and only serving 4,000 customers"*.	X						I / E
Legal compliance (law).	By law, business functionality from Case A went to Internal Revenue Service, making the IS redundant. New law required different calculations. Abandonment and migration project manager (5): *"changing law by politicians. When by law a certain social security product line or service line within social security that the case organization is supplying to the clients is abandoned, the IS should be abandoned".* In the past this happened with Case A, former branch organizations supplied more services like pension services. It was decided by the government that this was no core business anymore and the IS was abandoned.	X						E

Security issues.	Consolidation project manager (1): *"when an IS is vulnerable for cyber-attacks, a decision to abandon an IS might be due to security issues"*.		X	X	X			I

Table 18. Unique triggers to abandon operational IS in Case A

Besides triggers to abandon, also one countervailing trigger was identified. Interviewees mentioned "complexity" of IS, both a reason to abandon and not to abandon legacy IS. Triggers opposing abandoning legacy IS have not been explicitly included in this research. The concept "complexity" probably has different implications for various interviewees and could probably be refined in subsequent triggers. This countervailing trigger is presented in Table 19.

Equivocal trigger to abandon (+) and a trigger not to abandon (-).	**Remarks.**	IS	Applications	Hardware	Data sets	People	Procedures	(I)nternal/(E)xternal
Complexity	If a legacy IS is very complex, this might be an argument to abandon a legacy IS, due to the fact that implementing changes in a complex legacy IS is risky. On the other hand: abandoning a complex legacy IS is more difficult and risky than a less complex legacy IS. If a legacy IS is very complex, this might be an argument not to abandon or to postpone abandoning of a legacy IS. Business information manager (5) said: *reduction of complexity is a reason to abandon legacy IS*". Architect (2) said: *"a trigger to abandon a legacy IS, is when the legacy IS become too complex, and it cannot be managed anymore"*. Business information manager (4) said: "*we had three legacy IS, we built workarounds on the legacy IS, we wrapped the legacy IS and we became into a situation where the systems became too complex, we could not continue like that and decided to do complexity reduction, by successively abandoning legacy IS".* CIO/ICT director (1) said: *"we made the systems more and more complex, ... complexity of hardware, network, software, interfaces, ... maintenance will become more and more difficult".* Abandonment and migration project manager (1) said: *"if you decided to abandon the legacy IS, and after you researched the system deeper, it can turn out that it is more complex than you thought at forehand, than it is maybe better not to abandon the legacy IS".* Business information manager (8) said: *"the complexity of large systems makes that you rather keep them for some time instead of abandoning them and change them for a new IS"*.		X	X				I

Table 19. The equivocal complexity trigger to abandon or retain operational legacy IS

From the literature research (see Table 7, p. 34) 11 abandonment triggers were identified. From Case A, 25 abandonment triggers were identified. There was an overlap of four abandonment triggers. Resulting in 32 unique identified legacy IS abandonment triggers. One trigger is an equivocal trigger, this is "complexity". Interviewees mentioned that complexity is also a trigger not to abandon a legacy IS. These triggers to decide to abandon a legacy IS were further categorized into internal or external triggers. An overview of the abandonment triggers is provided in Appendix C.3 (p. 193).

Regarding SRQ 7 it can be concluded that in Case A there were 25 unique triggers to abandon legacy IS. Most of these triggers relate to the IS as a whole, or to application functionality or to hardware. Economic obsolescence seems to be something that primarily "happens" to the IS (through external developments).

SRQ 8: Which stakeholders participate in abandonment decision making of legacy IS?

Based on the questionnaire (Q18) and (Q19) (see Appendix A.1, p. 173) stakeholders of legacy IS and the stakeholders who participate in the decision making to abandon legacy IS were elaborated. Table 20 illustrates the stakeholders that participate in abandonment decision making of legacy IS in Case A within Trail 1 and Trail

2. The "Y" implies that the particular stakeholder participated in the decision to abandon legacy IS at any time in the decision making process, "N" means that an identified stakeholder did not participate in the decision to abandon any legacy IS.

Function (English)	Participate in decision to abandon a legacy IS	
	Trail 1	Trail 2
CIO/ICT Director (1)	Y	Y
Architect (1, 2, 3, 4 & 5)	Y	Y
Business information manager (1, 2, 3, 4, 5, 6, 7 & 8)	Y	Y
Abandonment and migration project manager (1, 2, 3, 4 & 5)	N	Y
IS owner (1) (e.g. director, business unit director)	Y	Y
Board of Case A	Y	Y
Functional maintenance manager (1& 2)	Y	Y
Technical maintenance manager (1)	Y	Y
Business operations manager (1)	Y	Y
Finance director (1)	Y	Y
ICT controller (1)	N	N
Renewal programme manager (1)	N	N
Senior user (1 & 2)	N	N
Consolidation project manager (1 & 2)	Y	Y
Supplier and contract manager (1)	N	Y
Consolidation programme manager (1 & 2)	Y	N
IS merger expert (1) (worked at Case A consolidation)	Y	N
Account manager technology and services supplier of Case A (1)	N	N
Government	N	N
Clients	N	N

Table 20. Legacy IS abandonment decision making stakeholders identified during the interview

Not all of the identified stakeholders participated in the abandonment decision making. The users, clients, ICT controller, renewal programme manager and suppliers were not invited to participate. The so called IS owner, who can be the business unit director for important/major IS or Business information managers representing the minor IS was always involved. The involvement of the rest of the stakeholders was dependent on the specific situation. It was recognized that external consultants can be a major change agent in large organizations, even though they might have different experiences and motivations for IS change compared to internal actors, which were not identified in this research.

SRQ 9: Who is held responsible for the abandonment decision of the legacy IS?

In Case A the responsibility to abandon legacy IS was primarily taken by business managers; the IS owner (e.g. director, business unit director) was always accountable. An IS merger expert (1) said*: "the IS owner, (the director) of the business processes which are supported by the IS, is at the end the one who takes the decision to abandon a legacy IS".* At the highest organizational level there was the political decision (enacted by law) to consolidate the six former branch organizations into Case A. Subsequently, business unit directors took responsibility to reorganize their information functions. At the level of operational IS, project managers were held responsible for abandoning targeted legacy IS.

SRQ 10: What are good practices for legacy IS abandonment decision making?

Interviewees were also asked to state their good practices by means of stating success criteria and failures in the process of legacy IS abandonment decision making (see Q21, Appendix A.1, p. 173). These were found by the organization by accumulating and applying knowledge about what is working and not working in different situations and contexts, including lessons learned and the continuing process of learning, feedback, reflection and analysis (Alberti & Bertucci, 2006). Due to the limited scope of this research and the absence of comparisons with "better practices", the practices will be labeled "good practices". The organization identified the good practices by repeated experiments. Further additional research by other researchers should validate if good practices are best practices.

By means of first cycle coding all good practices were structurally coded and 19 good practices were identified. Subsequently the unique good practices were identified through second cycle coding, in which all the similar IS abandon good practices were axial coded. This resulted in 16 unique good practices, which are presented in Table 21.

No.	**Good practices for legacy IS abandonment decision making.**
1	The decision to abandon a legacy IS should always be driven by business economics (even though the organization was a public organization). IS owner (1) said: *"it is a rational decision based on cost, benefit and return on investment*".
2	To support the legacy IS abandonment decision making, in-depth analyses of this decision was necessary.
3	Plotting applications on the business processes helped in visualizing overlap of functionality; this was used for decision making.
4	A success factor for abandoning legacy IS decision making was to actually know the existence of all applications (have the application base organized, e.g. know their names, functionalities and who were using them).
5	Commitment by board management was a success factor.
6	Success factors in decision making were that goals were made clear to everybody and management acted in accordance with these goals.
7	In case of mergers, it was an advantage to have decision makers who did not personally purchase or built the IS.
8	IS owners should pay for system maintenance. According to Architect (1): *"a success factor was that IS owners paid for system maintenance .... In these cases, they saved resources through abandoning overly expensive or redundant systems. They were able to prioritize within their total cost of ownership".*
9	Pressure on the expense budgets helped to make decisions to abandon legacy IS (every abandoned IS would no longer consume resources, e.g. money and staff).
10	In the decision making to abandon legacy IS, the programme that realized the abandoning of the legacy IS operated as a team, the client of Case A was the focus, not the internal powers in Case A.
11	It was suggested to have regular meetings (e.g. once or twice a year), as part of information planning and to evaluate all IS and to decide which IS could possibly be abandoned.
12	Three stakeholders should always be part of the decision making process: business, ICT and finance. The IS owner (e.g. director, business unit director) should be kept accountable for the decision whether and when to abandon a legacy IS.
13	Prohibiting adding new IS, otherwise growth might still outnumber abandoning.
14	Scoring models including criteria and weighing factors supported legacy IS abandonment decision making.
15	Sometimes the system was redundant, but its data needed to remain accessible (for instance compliance). A solution offered to store this data (e.g. on CD) helped the abandonment decision making.
16	Adequate information concerning the legacy IS (interfaces, impact, costs and contracts) was very important.

Table 21. Good practices for legacy IS abandonment decision making

In summary it can be stated that this research found that many components of an IS were affected by abandoning a legacy IS. Further it was found that abandoning the individual components, software, hardware, data and procedures seemed to differ from abandoning the people working with this IS. It is concluded that the decision to abandon a legacy IS cannot be taken solely on evaluating only one component within the legacy IS.

Concerning the available IS documentation, contradicting arguments were identified. According to several interviewees adequate legacy IS documentation is important e.g. *"if you know the quality of the system, you know what you are abandoning".* Contradicting arguments concerning the state of documentation were found, according to Architect (4): *"it is normal that a system lacks documentation, programmers do not like to document, so the quality of the legacy IS documentation is per definition inadequate".*

Based on examples provided by interviewees. It is concluded that*:* a decision to abandon a "legacy IS" cannot be taken in isolation; the impact of abandoning a legacy IS to other IS should be known. An example provided by Architect (1) was: *"an application was abandoned in January and eleven months later in December, the system had to generate input to a yearly list with information that had to be sent to the Minister".* In Case A it was particularly important that all interfaces with other IS were identified; also the impact of data delivered from the legacy IS to the remaining IS is crucial.

5.7 The practical abandoning process of legacy information systems

In the predominant life cycle stage: “the practical abandoning process of legacy IS”, the third research question is: “*how do organizations practically abandon legacy IS?*”

This section describes the practical abandoning process of legacy IS in Case A. It is based on the answers provided by the interviewees on the following sub-research questions (SRQ):

- SRQ 11: How are legacy IS practically abandoned?
- SRQ 12: Who is held responsible for practically abandoning legacy IS?
- SRQ 13: What are good practices for practically abandoning legacy IS?

SRQ 11 How are legacy IS practically abandoned?
To practically abandon the legacy IS, Case A started with a list of all applications. This list also included supplier names and platform names (this process is described in Chapter 4). The IS owners decided which application was to be abandoned and which application was to be migrated to the new data center. IS owners primarily included directors and business unit directors. Then for the applications which were to be abandoned an abandoning calendar was drawn up and the applications were delivered user free from the business to the A&M programme. In case the application owner also required data storage, this data was stored in a temporary file (Historical data storage). Subsequently, the contract with the supplier was terminated. The progress of the A&M programme was monitored by a so called “abandonment thermometer”, which visualized the "to be" abandoned applications and the practically abandoned applications. This abandonment thermometer was also used to communicate the successes of the A&M programme within the organization (it was published on the intranet and on banners which were placed on several places within the buildings).

SRQ 12: Who is held responsible for practically abandoning legacy IS?
In Case A, the IS owner (director or business unit director) was always held accountable for the decision to abandon any legacy IS. Practical abandoning was delegated to the responsible A&M programme manager. The A&M programme also ensured that the legacy IS was practically abandoned.

SRQ 13: What are good practices for practically abandoning legacy IS?
Interviewees were also asked for success criteria (good practices) in the process of practically abandoning legacy IS (see Q26 and Q27, Appendix A.1, p. 173). These were found by: *“accumulating and applying knowledge about what is working and not working in different situations and contexts, including lessons learned and the continuing process of learning, feedback, reflection and analysis”* (Alberti & Bertucci, 2006). The organization identified the good practices by repeated experiments. Due to the limited scope of this research and the absence of comparisons with “better practices”, the practices will be labeled “good practices”. Further research by other researchers should validate if good practices are best practices.

By means of first cycle coding, all success factors and failures were structurally coded and 46 good practices were identified. Subsequently, the unique good practices were identified through second cycle coding, in which all the similar good practices were axial coded. This resulted in 40 unique practical abandoning good practices, which are presented in Table 22.

No.	Good practices for practically abandoning legacy IS.
1	Have a fall back scenario.
2	Have a strategy to back up your data, specific database (named: "*Historical data storage"* in the case organization), data on disk, or throw away (dispose) media.
3	Identify the interfaces to other IS. It was a success factor if organizations knew how a system interfaced with other systems and how this would affect the other systems.
4	Start with an IS database (a baseline) that includes indicators such as: application names, costs, number of users, contracts and suppliers. The completeness of such a database was called a success factor.
5	In case people had to abandon their own legacy IS and also their own job (they were dismissed after the abandoning of the legacy IS) two things were important: an explanation about why the IS was abandoned and an adequate exit compensation.
6	In case an IS was abandoned due to the fact that there was a merger and another IS was preferred, avoid detailed discussion with stakeholders about the arguments.
7	In case an IS is succeeded by another system, make sure that there is adequate training on the new system.
8	We worked with the beep mechanism: we turned an IS off and when somebody was complaining we turned it on.
9	It was a success factor to make the business (IS owner, e.g. director, business unit director) accountable for the practical abandoning of an IS (ownership of the business) and to offer help from a decentralized organizational unit.
10	In case several IS were abandoned, creating a programme and making a standardized process was a success factor. Also put this programme outside of regular business operations. Otherwise problems in primary processes might prevail.
11	It cost a lot of time and resources to untangle all the IS which were to be abandoned (spaghetti).
12	Sometimes applications were found that were not listed in the company's application database.
13	When IS are practically abandoned, all data should be removed from these IS.
14	Inform stakeholders about the plans to abandon a legacy IS. According to Consolidation project manager (1): *"communication of abandoning a legacy IS is important, because abandoning an IS will influence the work life of the stakeholders and this communication must also be regularly repeated".*
15	Give stakeholders from an abandoned IS a new job (in the new IS). According to IS owner (1): *"stakeholders would better help to terminate the old legacy IS, if a new job was offered".*
16	Test alternative IS extensively before abandoning the legacy IS.
17	The pressure to migrate to a new data center was useful in abandoning the legacy IS.
18	Avoid lumpsum contracts with suppliers. For instance, 10 applications including maintenance and operation were on one server. When a legacy IS was abandoned, it was not clear which part of the costs were recovered.
19	Avoid dispersion of the IS contracts and make them transparent. For instance the IS was split amongst (different) suppliers into test environment contracts, data center contracts, software contracts, database administrator contracts, maintenance contracts and were sometimes part of a lumpsum.
20	When IS were outsourced, professional contract management had to be supported by an adequate database.
21	IT staff should talk business language.
22	In case it was a decision of the board of directors, a clear decision of the board of directors to abandon or not to abandon a legacy IS had to be made.
23	Between the decisions to abandon a legacy IS and the practical abandoning of the legacy IS there was some leap time. In that case the maintenance was reduced to a bare minimum; only "must have" functionality was maintained. This implied that maintenance budgets were reduced. The bare minimum maintenance which was allowed was documented very well, to avoid additional abandoning work when the legacy IS was finally abandoned.
24	Technological complexity or the availability of support was a critical factor. In one case the legacy IS included reporting functionality and it was a lot of work to transfer this functionality to another system, also because the person responsible for this functionality left the organization almost immediately after hearing that the IS was going to be abandoned.
25	If an IS service provider that provided data center services and application maintenance services was abandoned, success was higher if a representative from the application department rather than a representative from the hardware department was approached. In outsourcing situations it was better to approach the applications people than the data center people. It seems that there was a hierarchy and that the application staff is higher in hierarchy than the hardware staff.
26	Communicate the progress of your abandon programme throughout the organization; in Case A the "abandonment thermometer" visualized the successes of the A&M programme.
27	If data stored at suppliers was abandoned, the suppliers should also practically abandon this data by contract.

28	Abandoning an IS was very painful for some stakeholders because these stakeholders identified themselves with a system and the importance of these systems. Pay attention to this from a management perspective.
29	Bring an IS to the grave. People forget that a system was often built with a lot of affection. Abandoning this system was therefore not always easy, make sure to have a party afterwards.
30	Consistently and consequently follow the goal to abandon a legacy IS.
31	Adequate project management skills in the abandoning of a legacy IS.
32	A sponsor on the highest board of the organization was absolutely necessary. According to Business information manager (3): "*if there is a board member, who weekly asks whether the legacy IS has already been abandoned, this will definitely happen".*
33	Between the decision to abandon and the actual practical abandoning there could be a long time lapse (e.g. in case of a merger, years). To ensure that the abandoning of a legacy IS was still accepted by the owner, a few weeks before the actual practical abandoning of the IS the owner, of the IS was asked to sign a document stating that he agreed to abandon the IS.
34	Abandon the contracts with suppliers and cash the benefits by not paying anymore for the IS.
35	It was suggested to start with the simple low risk IS and to cluster similar IS.
36	Apply continuous improvement, "plan-do-check-act" cycles to evaluate and learn abandoning activities.
37	Adequate project or programme management to practically abandon legacy IS.
38	Pressure on the expense budgets of the business helped IS owners (e.g. directors, business unit directors) to decide to practically abandon legacy IS (every abandoned IS would not consume resources, e.g. money and staff).
39	After the decision was taken to abandon the legacy IS, it took some time before the legacy IS was practically abandoned. At the practically abandoning, there were emotions involved of people attached to the legacy IS. From a managerial perspective, pay attention to these emotions.
40	Select the right project members for the abandoning project. According to Business information manager (8): *"there are always things which will go wrong, you know this from the start, and you need to have a knowledgeable and enthusiastic team to handle these problems".*

Table 22. Good practices associated with practical abandoning

Furthermore, the role of communication was researched (see Q39, Appendix A.1, p. 173). It is concluded that communication about the abandoned legacy IS amongst stakeholders is important, because the associated systems often significantly impact their work life. Project manager consolidation (1) said: *"communication concerning the abandoning of a legacy IS is important, because it will impact the work of stakeholders, this message has to be repeated regularly".* This was agreed upon by the Senior user (1) who said: *"it is important to communicate the processes around the abandoning of legacy IS".* The A&M programme had its own communication staff and all employees were regularly informed regarding the migration and abandoning strategies in Case A. The Business operations manager (1) said: *"despite it was known and communicated that the system was to be abandoned, people still are emotional when it really happens, it hurts and communication did not prevented that*". An example of the impact on work and life was that when Case A announced that the ABC IS (Trail 1) was abandoned, an expert of this system immediately left the organization, because he could quickly find another job. This left Case A with a knowledge gap regarding the details of this system. Case A, therefore, also illustrates that abandoning IS is also abandoning people's work and changing working processes. Communication regarding these changes and the implications for individual staff members is important.

5.8 Other results

Besides the legacy IS abandonment triggers, also other factors influencing legacy IS decision making were found in Case A. These other results are derived from the interviewee transcripts and additional desk research:

1. There is a difference between the abandonment decision of primary business applications and secondary (supportive) applications (e.g. a Finance or HRM application). This is because the business impact might be lower if something goes wrong when a secondary supportive legacy IS is abandoned compared to a primary business legacy IS. Secondary supportive applications were often bought in the past as commercial off-the-shelf (COTS) systems. Stakeholders were less attached to these IS, because they were not built and designed by "us". There were more opportunities for replacement due to the fact that secondary IS were used by more organizations, resulting in competition of suppliers and scale efficiencies. Other organizations might already have experience with abandoning similar secondary IS. According to the Finance director (1): *"all IS*

supporting primary business processes were custom-made, in contrast to IS supporting secondary processes, we are not unique in our HRM system or our procurement system, there is a market for these systems.... If an IS supporting a primary business process goes down, it is much worse than when our procurement IS goes down".

2. Large legacy IS have a higher survival rate compared to smaller legacy IS in Case A and were not easily abandoned, due to the fact that:
   - According to Business information manager (1): *"they support more products or services (e.g. sometimes up to 25 products or services); consequently, if one product or service is abandoned there will be many left (e.g. 24)".*
   - They have more users and often more powerful stakeholders.
   - In case of a merger, it is expensive to train more users for a small IS than a few users for a large IS.
   - According to project manager consolidation (1): *"large systems are in the heart of the organization and more risky to abandon than a small system, compare it with an open heart surgery".*
   - Specific functionality of a small legacy IS is relatively easy to implement in a large IS, compared to implementing large functionality of a large IS into a small IS.
   - A large legacy IS can easily handle additional small volumes, while a small legacy IS might have difficulties to handle large amounts of extra volume (scalability).
   - According to the Finance director (1): *"because they are large and therefore difficult to rebuild and implement (take a lot of time, money and resources) compared to a small legacy IS which is easy built and implemented".*
   - According to the CIO/ICT director (1): *"it is simply not possible to rebuild a large legacy IS due to the fact that years of development (e.g. five years) and maintenance (e.g. 10 years) are invested in the legacy IS and organizations often do not have the skills to replace this large legacy IS".*
3. The more activities around the legacy IS were outsourced, the easier the legacy IS abandonment decision is taken and the legacy IS practical abandoning is realized, due to the fact that:
   - There were less personal connections with such legacy IS.
   - No internal personnel was fired.
   - There were contracts and it is a business transaction (terminating a contract).

   According to Abandonment and migration project manager (1): *"it is easier to abandon outsourced IS, … of course the supplier will complain, as the supplier is losing revenue. However, it is for an organization easier to tell a supplier to abandon a legacy IS, than to tell to your own staff that they are not needed anymore, … social aspects are important, as the internal staff delivered on time and was reliable, the internal staff was very dedicated even helped you on Friday night, ... it has to do with names, with affection, abandoning IS supported by internal staff is harder, compared to abandoning outsourced legacy IS by terminating a contract with suppliers"*.
4. The stakes of stakeholders in a legacy IS can be high: (e.g. revenues for suppliers, management roles, and span of control). Abandoning high stakes is risky. Legacy IS were occasionally referred to as 'friends' and suppliers earn money with them. For staff managing these systems (e.g. Business information manager or, Business information administrator), these systems form their jobs. Existing suppliers want you to stay and new suppliers always promise much better systems.
5. The legacy IS is trying to "survive", it influences the demand (politics) up front to make sure that the politicians will not implement laws that the legacy IS could not handle.

Finally in the Appendix E, (p. 202) an evaluation of the merger of the information functions (Trail 1) and an evaluation and results "Abandonment and Migration" (Trail 2) are provided.

5.9 Summary and conclusions

In this chapter three subsequent life cycle stages were elaborated upon: (1) *the aging process of legacy IS*, (2) *the abandonment decision making process of legacy IS*, and (3) *the practical abandoning process of legacy IS*. In each stage there was a research question (see, p. 53) and each stage was further refined by sub-research questions.

Concerning: "*the aging process of a legacy IS*". The term "legacy" in the word "legacy IS" was not formally defined in Case A. According to most interviewees (75%) legacy had a negative connotation. Interviewees in Case A mentioned legacy characteristics of aging legacy IS. By means of first cycle coding in Case A, 635 legacy characteristics of aging legacy IS, or its constituents (application, hardware, data sets, procedures and people) were identified. By means of second cycle coding (axial coding) this was reduced to 105 unique aging legacy characteristics. The legacy IS aging characteristics from Case A combined with the legacy IS aging characteristics from the literature resulted in 140 unique legacy characteristics of aging legacy IS (42 IS-related legacy characteristics, 46 application-related legacy characteristics, 19 hardware-related legacy characteristics, 16 data-related legacy characteristics, six procedure-related legacy characteristics and 11 people-related legacy characteristics). These legacy characteristics are described in Appendix C.1 (p. 185). It is concluded that there were many legacy characteristics. This is due to the fact that the legacy characteristics were not categorized into a higher level. This research chooses not to do so, but to distinguish them at an elementary level. Other research might categorize them into a higher level and relabel them.

These legacy IS aging characteristics were further categorized into internal and external aging factors. Internal aging factors were within the boundaries of the system and could be influenced by organizations; external aging factors were outside the boundaries of the system and could not be influenced by an organization. These are described in Appendix C.1 (p. 185). Legacy IS were evaluated at the inception of Case A by architects with help of a multicriteria model. Interviewees suggested to identify legacy IS by scoring all applications on functional, technical and business value and make this a yearly process.

Concerning: "*the abandonment decision making process of legacy IS*". Mandatory for decision making was an overview of the existing IS. In Case A legacy IS abandonment decision making was found at three different functional abstraction levels. These levels were information function, IS and application level. Three interacting decision making levels were identified and object of this research. Information functions were decided at a higher level (e.g. the highest board decided) than the IS level (the IS owner, e.g. director, business unit director) and the lower application level (e.g. Business information manager). In Case A, by means of multicriteria analyses and evaluating scenarios it was decided which legacy IS to abandon. Many stakeholders participated in the decision to abandon legacy IS (see Table 20, p. 76). The IS owner (the director, business unit director) was accountable for the legacy IS abandonment decision.

Interviewees in Case A mentioned legacy IS abandonment triggers. By means of coding, 25 unique legacy IS abandonment triggers were identified (see Table 18, p. 75). One of the legacy IS abandonment triggers "complexity" is an equivocal trigger. Interviewees mentioned complexity a reason to abandon the legacy IS, but also a reason not to abandon the legacy IS. *"If a legacy IS is very complex, this might be an argument to abandon a legacy IS, due to the fact that implementing changes in a complex legacy IS is risky. On the other hand abandoning a complex legacy IS is more difficult and risky than a less complex legacy IS. If a legacy IS is very complex, this might also be an argument not to abandon or to postpone abandoning"* (Business information manager [5], Architect [2], Business information manager [4], CIO/ICT director [1], Abandonment and migration project manager [1], Business information manager [8]).

The equivocality of trigger "complexity" suggests that this trigger could be dissected into subsequent triggers. For example a complexity discontinuance trigger to abandon the legacy IS, and a complexity continuance trigger to maintain or re-engineer the system. By means of re-engineering (see Figure 13, p. 35) the complexity of the legacy IS can be reduced, this should fit a legacy IS continuance perspective. The concept "complexity" probably

has different implications for various interviewees. Later on research on the equivocality of trigger “complexity” is advised.

The IS abandonment triggers from Case A combined with the legacy IS abandonment triggers as found in the literature result into 32 unique legacy IS abandonment triggers. These were further categorized into internal and external legacy IS abandonment triggers. These triggers are described in Appendix C.3 (p. 193). Finally 16 good practices were identified regarding legacy IS abandonment decision making (see Table 21, p. 77).

Concerning “*the practical abandoning process of legacy IS*”. An overview of the existing IS was a prerequisite for practical abandoning. Furthermore, 40 good practices regarding practical abandoning were identified see Table 22 (p. 80). Although the practical abandoning was performed in a programme, separated from the line organization, the IS owner was always accountable.

Furthermore, a number of additional results were identified. A difference was identified between abandoning legacy IS supporting primary processes and abandoning legacy IS supporting secondary processes. If something goes wrong the business impact might be lower when a secondary supportive legacy IS is abandoned compared to a primary business legacy IS. Furthermore, it seemed that large legacy IS survived more easily compared to smaller legacy IS. Commercial off-the-shelf (COTS) systems were more easily abandoned compared to proprietary IS. Finally, outsourced legacy IS were more easily abandoned compared to insourced legacy IS. In Section 5.8 (p. 80) these causes were explained.

This research follows the three predominant life cycle stages (first, aging, then, abandonment decision making and, finally, the practical abandoning process of legacy IS). For each of the life cycle stages a research question and supported SRQ were identified. The answers to these SRQ provide the input for the research objective, which is to propose a method to abandon legacy IS (see Section 5.2, p. 53). In the following chapter the explanation phase will be described. In this explanation phase the conceptual models related to aging, abandonment decision making and practical abandoning of legacy IS are conceptualized.

PART 2: EXPLANATION PHASE

6 AGING AND ABANDONING PROCESS

"Old age has its infirmities"[10]
---Halma, F. (1710)---

6.1 Introduction

As described in Section 2.8 (p. 17) this study follows the research outline described by Van Engelen and Van der Zwaan (1994). This chapter describes the explanation phase. The explanation phase is divided into the following sections: first the aging process of legacy IS is conceptualized (Section 6.2), then abandonment triggers, decision making process (Section 6.3) and the practical abandoning process of legacy IS (Section 6.4) are conceptualized. Finally design dilemmas on legacy IS abandonment decision making methods (Section 6.5) are described. This chapter ends with a summary and conclusion (Section 6.6).

6.2 The aging process of legacy information systems

In this section the first life cycle stage: *"the aging process of legacy IS"* is conceptualized. The complexity of legacy IS aging was illustrated by the identification of 140 unique legacy IS characteristics from the literature and Case A research. These 140 characteristics referred to the IS and all of its constituents being: hardware, software, data sets, people and procedures (Brussaard & Tas, 1980). IS, like other systems, age. Although the aging factors are dynamic and change over time, IS gradually contain more legacy characteristics. In the literature no agreed accepted yardstick for measuring IS aging was found (Brooke & Ramage, 2001; Commela-Dorda, 2000). Events can occur anywhere within the environment or company and can be divided into internal events, those over which an organization has control (for instance, product launch) and external events, outside the control of the organization (for instance, regulatory change) (Gold, 1998). All legacy characteristics are further categorized into internal and external legacy IS aging factors (see Appendix C, p. 185).

The life cycle of an IS starts with planning followed by developing. During development there might already be some external aging, for example, due to the fact that the technology development in the real world proceeds. After implementation of the IS, the IS is operated and maintained. During this maintenance the IS will age due to accumulation of internal and external aging factors and becomes a legacy IS (a legacy IS is defined as any IS that significantly resists meeting organizations' requirements).

During maintenance and operation of the legacy IS, there are constantly opportunities to decide to continue or discontinue the legacy IS (Sommerville, 2007). This is illustrated in (Figure 13, p. 35). A decision to continue the legacy IS means continuing regular legacy IS maintenance in which functionality is added, or re-engineering the legacy IS in which the structure of the system is improved to make maintenance in the future easier and cheaper. A decision to abandon a legacy IS may result in scrapping the legacy IS, referred to as decommissioning by Warren (1999), or replacing all or parts of the legacy IS with new systems developed from scratch.

At some point in time, the aging crosses a certain legacy IS aging threshold, which is a trigger for decision makers to decide not to maintain a legacy IS anymore, but to re-engineer the legacy IS. Re-engineering is often incorrectly treated as being synonymous with software engineering. Its true scope is broader, although the software is often the focal point of re-engineering efforts and it alone is only one of the components of the complete system that must be considered when re-engineering (Tilley & Smith, 1995). The salient difference between engineering (development) and re-engineering is that the legacy system already exists compared to engineering a new system (development), which has to be incepted first (Tilley & Smith, 1995). Re-engineering is considered to again lower the internal aging factors.

[10] Dutch French dictionary by Halma (1710), translation of old Dutch saying to French: "*De ouderdom koomt met veele gebreken. La vieillesse vient accompagnée de bien des incommodités*".

After re-engineering the legacy IS, the IS is maintained again and the internal and external legacy IS aging factors accumulate again. This process continues until the moment that it is no longer beneficial to maintain or re-engineer the legacy IS. The legacy IS should be abandoned, possibly be replaced by a new IS. If the decision has been made to abandon a legacy IS, it takes some time to practically abandon a legacy IS. During this time no maintenance should be performed anymore and the legacy IS aging still continues (this occurred in Case A and is described in Appendix E, p. 202). This life cycle IS aging process is illustrated in Figure 23.

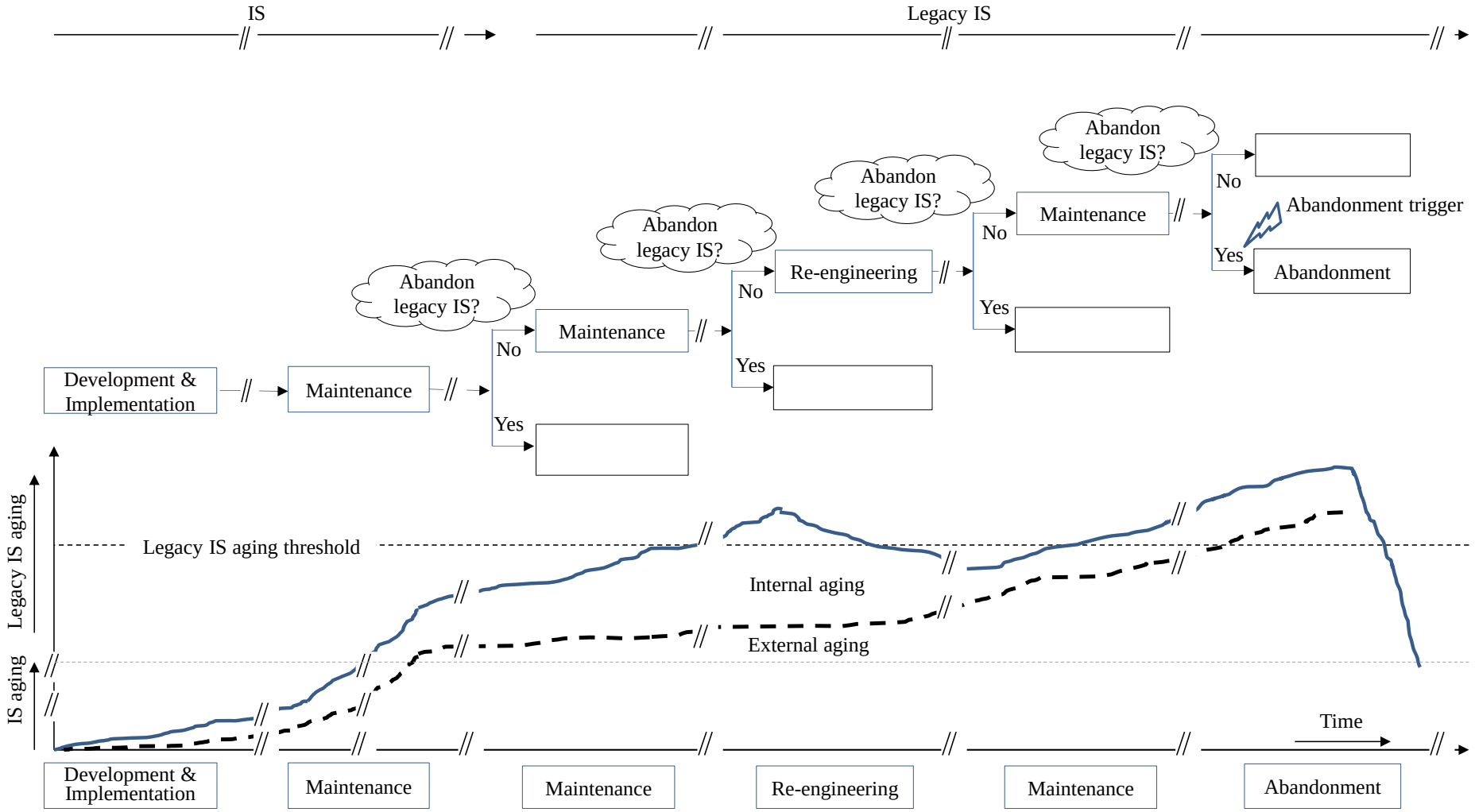


Figure 23. The aging process of legacy IS, the abandonment decision making process and the practical abandoning process of legacy IS

6.3 The abandonment decision making process of legacy information systems

In this section the second life cycle stage: *"the abandonment decision making process of legacy IS"* is conceptualized. In the former section the aging of legacy IS was described. The identified model suggested a gradual aging of legacy IS, through an increasing number of legacy IS aging factors. Besides the aging factors also arguments were identified that make organizations decide to actually abandon the legacy IS. These were referred to as abandonment triggers. Overall, 32 triggers were identified from which 25 in Case A and 11 from the literature; four triggers were found in both Case A and the literature (see Appendix C.3, p. 193). These triggers were also categorized in internally or externally rooted. All legacy IS abandonment triggers are illustrated in Figure 24.

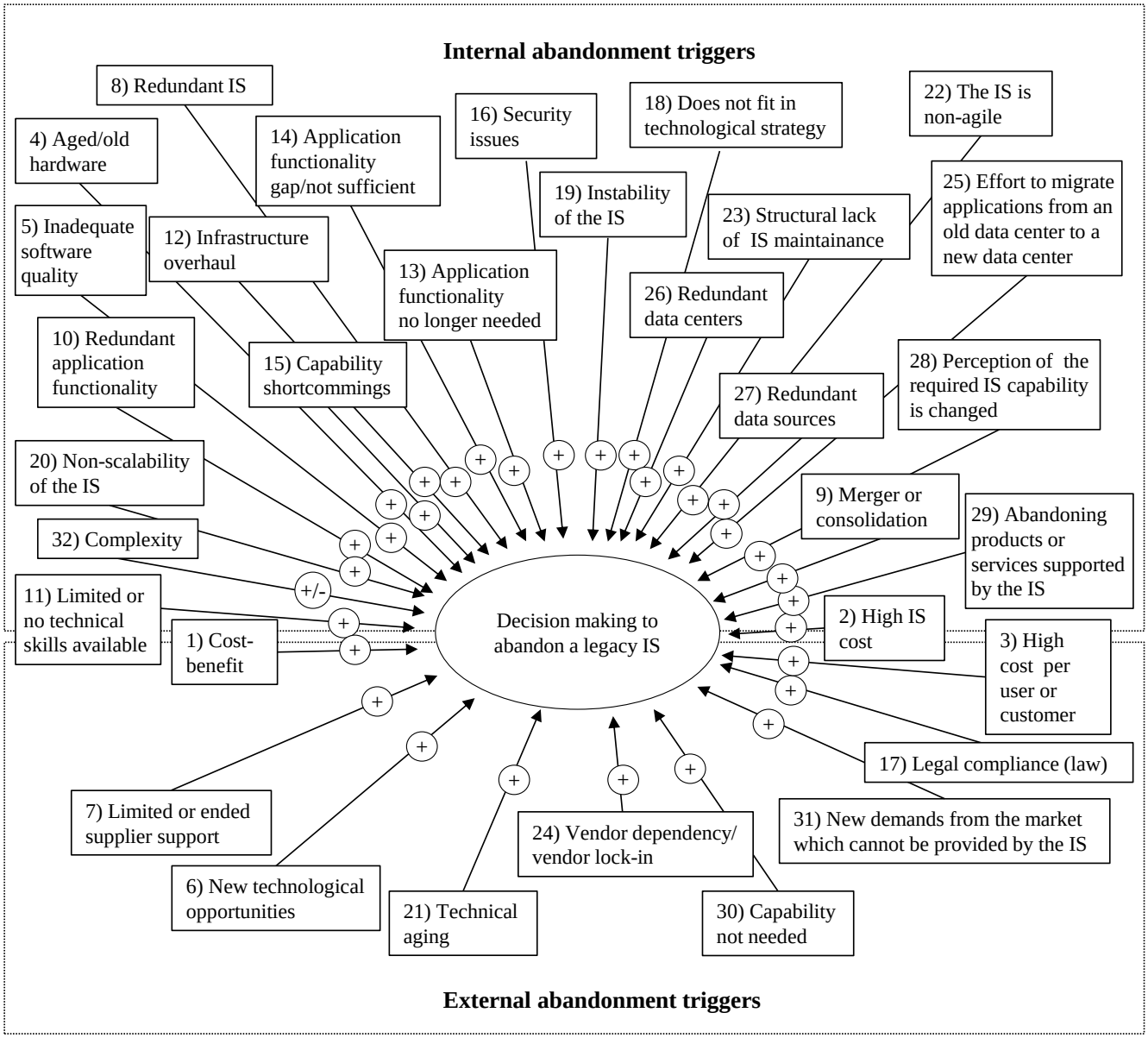


Figure 24. Overview of legacy IS abandonment triggers

Besides the categorization in internally- and externally- rooted, other categorizations are also possible. Categorization could be by means of its components (legacy application, legacy hardware, legacy data, legacy people and legacy procedures). Various triggers could apply to multiple components. Another categorization could be based on the eclectic decision making approaches as illustrated in Figure 14 (p. 36), especially the categorization in technical, functional and economical can be used for selecting decision makers, this categorization will be provided in Table 31 (p. 125).

In Appendix C.3 (p. 193) the legacy IS abandonment triggers are presented. Because an IS consists of applications, hardware, data sets, people and procedures in this section a further refinement is presented. One of the components of the IS might be a trigger to abandon a legacy IS (e.g. inadequate software quality). Concerning the abandonment triggers, the focus on this dissertation is on the abandoning of legacy IS, whilst others (Brodie & Stonebraker, 1995; Sellars, 2004; Ulrich, 2002; Van den Heuvel, 2007) focus on extension, modification or on migration strategies. There was one equivocal abandonment trigger mentioned by several interviewees which had both a positive and negative relationship. This concerns "complexity". *"If a legacy IS is very complex, this might be an argument to abandon a legacy IS, due to the fact that implementing changes in a complex legacy IS is risky. On the other hand abandoning a complex legacy IS is more difficult and risky than a less complex legacy IS. If a legacy IS is very complex, this might also be an argument not to abandon or to postpone abandoning of legacy IS"* (Business information manager [5], Architect [2], Business information manager [4], CIO/ICT director [1], Abandonment and migration project manager [1], Business information manager [8]). There is an overlap between aging factors and abandonment triggers. For example, "limited or ended supplier support" (e.g. a supplier that went bankrupt) was mentioned by some interviewees as an abandonment trigger, while others mentioned this both as a legacy IS characteristic (categorized external aging factor) and as abandonment trigger. This overlap concerned 13 abandonment triggers. It is concluded that the IS aging factors and legacy IS abandonment triggers both increase the aging of legacy IS. After the decision has been made to abandon the legacy IS, it will take additional time to practically abandon the legacy IS. During this period maintenance should be minimalized, resulting in additional aging factors. The above concepts are illustrated in Figure 25.

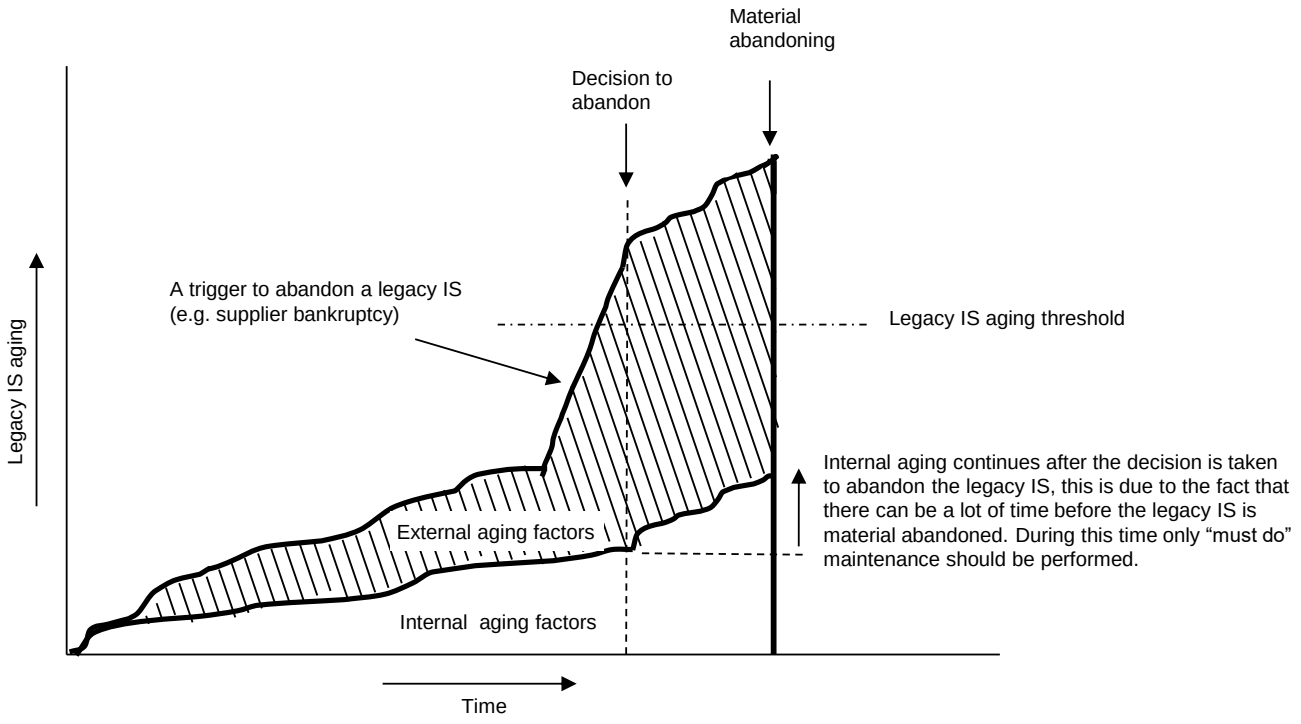


Figure 25. Legacy IS aging

Also other factors were identified which influenced the decision to abandon legacy IS (see Section 5.8, p. 80). Smaller IS were more easily abandoned compared to larger IS. Secondary (supportive) applications (e.g. a Finance or HRM application) were more easily abandoned compared to primary business applications. Because of their lower business impact. Also, secondary applications were often bought as commercial off-the-shelf (COTS) systems. Stakeholders were therefore also less committed to these IS. Normally, there were also more opportunities to replace secondary systems, due to the fact that secondary IS were used by more organizations, resulting in competition of suppliers and scale efficiencies. Furthermore, other organizations might already have

experience with abandoning similar secondary IS. Finally, outsourced legacy IS were also more easily abandoned due to the fact the accompanying contracts could easily be terminated.

Furthermore, good practices concerning the abandonment decision making of legacy IS were identified in both Case A (see Section 5.6, Table 21, p. 77) and the literature. These practices include an IS database which provides supporting information for legacy IS abandonment decision making (e.g. including all applications, interfaces, hardware, costs). Decision making should be preferably done by different disciplines (e.g. business, finance and ICT) to provide more knowledge. The IS owner (e.g. director, business unit director), however, should always be accountable for the final decision to abandon a legacy IS. Pressure on business budgets will help the IS owner to decide to abandon legacy IS, because every abandoned IS will no longer consume resources. It is preferable that decision makers do not have a personal attachments to the IS (e.g. personally purchased or built the IS). Decision making should be done by means of in-depth analyses and methods (e.g. a scoring model, including criteria and weighting factors or plotting application functionality on business processes). Stakeholders with different viewpoints (technical, functional and economical) should participate in the decision making. It is suggested to perform regular (e.g. one or two times a year) evaluations of all IS. Prohibition to add new IS is also a good practice, otherwise the number of IS will grow. Finally, clear goals and strong sponsorship from senior management concerning the IS decision making related to the abandonment of legacy IS is important. These legacy IS abandonment decision making good practices (see Table 21, p. 77) are further investigated in Chapter 7 (p. 101).

6.4 The practical abandoning process of legacy information systems

In this section the third life cycle stage: "*the practical abandoning process of legacy IS"* is conceptualized. In the former section the abandonment triggers of legacy IS were described. The identified model suggests a legacy IS abandonment trigger which make organizations decide to abandon legacy IS, the IS owner (e.g. director, business unit director) should always be accountable for the decision to abandon a legacy IS. When the decision is taken to abandon the legacy IS, the legacy IS have to be practically abandoned. Between the decision to abandon a legacy IS and the practical abandoning of the legacy IS will be additional time.

The practical abandoning process based on Case A is conceptualized as follows. A comprehensive IS database is necessary to support the practical abandoning of legacy IS. The IS database should include indicators such as: application names, costs, number of users, interfaces, hardware, contracts and suppliers. Interfaces between the abandoned legacy IS and other IS should be identified, to know the impact of the abandoned legacy IS on the recurring IS. Untangling these interfaces cost a lot a time. A strategy (delete, or store) to handle the data should be in place. When the legacy IS is abandoned a fall back scenario should be in place.

Due to the fact that the business priority is the continuity of business operation and not the abandoning of legacy IS, the practical abandoning of legacy IS can be done by a project or programme with specialized skills. In this case the responsibility to abandon a legacy IS is delegated to a project or programme manager, however, the IS owner (e.g. director, business unit director) is always accountable for the practical abandoning. When the decision is made to abandon a legacy IS, stop the maintenance immediately; more maintenance makes it more difficult to abandon. If stakeholders of the abandoned IS are given the prospect of another job, they are more willing to help in the abandoning process.

Lumpsum contracts with suppliers should preferably be avoided due to the fact that it is not clear what cost is recovered when the legacy IS is practically abandoned. For instance, 10 applications including maintenance and operation were on one server. When a legacy IS was abandoned, it was not clear which part of the costs were recovered. After the legacy IS is practically abandoned, terminate the contracts with suppliers and cash the benefits by not paying anymore for the IS.

Furthermore, good practices concerning the practical abandoning of legacy IS were identified in both Case A (see Section 5.7, Table 22, p. 80), and the literature these are further investigated in Chapter 7 (p. 101).

6.5 Design dilemmas for a method to abandon legacy information systems

From the literature and Case A, a method to abandon legacy IS decision making emerged. This section elaborates upon requirements for a legacy IS abandonment decision making method. It will do so by means of six design dilemmas. Regarding some dilemmas a position is taken based on the research presented so far. These design dilemmas are tested in the validating case research of Chapter 7. The design dilemmas are:

1. Enhancing existing IS decision making methods or designing a new legacy IS abandonment decision making method?
2. Decision making at information function, IS or application level?
3. Financial versus non-financial evaluation aspects?
4. Many or few attributes in the abandonment decision making method?
5. Life cycle or portfolio oriented legacy IS assessment?
6. Many stakeholders versus few stakeholders involved in legacy IS abandonment decision making?

1. Enhancing existing IS decision making methods or designing a new legacy IS abandonment decision making method?

This section elaborates upon the question whether a new legacy IS abandonment decision making method should be developed or whether available inception or maintenance decision making methods could be enhanced. Three types of decision making methods are discerned, all addressing a particular life cycle phase (Swinkels, 1997):

- Decision making regarding the inception of IS (e.g. Berghout, 1997; Renkema & Berghout, 1997).
- Decision making regarding maintenance of IS (e.g. Looijen, 1995; Pols & Backer, 2006; 2007).
- Decision making regarding abandoning legacy IS (e.g. Furneaux & Wade, 2011; Sakthivel, 1994; Sellars, 2004; Swanson & Dans, 2000).

These three decision making methods in the life cycle phase are compared using the following characteristics: (1) management level of the involved decision maker, (2) organizational risks involved, (3) financial implications of the decision, (4) effect on operations, (5) technological implications, (6) operations or project, (7) documentation quality and (8) individual prestige (image or status effects) of being involved in this type of decision making.

Ad (1). The decision to invest in new IS (the inception decision) concerns strategic investments, which are taken by higher level managers compared to maintenance decisions. New IS have to compete with other scarce resources and in Case A the board of directors, would have to confirm these decisions, whilst for maintenance decisions this board would provide a lumpsum for maintenance. Strategical abandonment decisions (e.g. legacy IS abandon programmes) are taken by the highest management, whilst operational abandonment decisions are delegated to middle management.

Ad (2). Concerning the organizational risks involved, it can be argued that these are high when investing in new IS. Concerning the maintenance of systems, these organizational risks are considered to be lower. An IS merger expert (1) said: "*if eight new functional upgrades have been performed and went well, the ninth functional upgrade will probably also go well*". However, abandoning an IS can be risky. Certainly, when capabilities of a legacy IS have to be transferred to a new IS and an important business process is supported, or when important databases could be corrupted. Evidently, abandoning could also concern less risky systems (e.g. supporting secondary processes, or only affecting few customers). Also the risk of abandoning a legacy IS with important data files might be more serious than introducing a new system where limited data history exists.

Ad (3). Concerning the financial implications of the decision, the investments in new IS are higher than the investments in the maintenance of an operational IS, although the operational cost of an IS are during the lifetime much higher than the initial investment (Berghout & Nijland, 2002; Sommerville, 2007). In Case A

abandoning legacy IS cost between 15k - 150k Euro per application, depending on the complexity of the IS. It can be argued that it is most difficult to forecast the future cost of newly built IS, due to the fact that there is limited cost history at that stage. For maintenance cost of existing IS there is always cost history available and there is often experience regarding cost needed to implement additional functionality. Concerning the financial implication of the decision to abandon a legacy IS, it is argued that cost history is available.

Ad (4). Concerning the effect on operations, it can be argued that the inception of a new IS means something extra: it might considered to be a major change, while maintaining current IS means something more of the same and is considered to be a minor change, and abandoning a legacy IS means something less.

Ad (5). From a technological implication perspective, it is argued that "new" investments probably involve "newer" or even "latest" technology, whilst maintenance has to do with "current" technology and abandoning legacy IS is associated with "older" technology.

Ad (6). Other differences regarding the investment concern its organization (project or operations). Inception of an IS was primarily organized as a project in Case A, whilst maintenance is done as operations. Abandoning legacy IS in Case A was also performed in a project.

Ad (7). Another difference concerns the quality of documentation of IS within the three different stages. During inception Case A had ample opportunities to develop proficient documentation. However, documentation of the operational IS appeared too often incomplete or inadequate. Documentation of the abandoned IS is often absent (Bennet, 1995; Bisbal et al., 1999; Brodie & Stonebraker, 1995; Daga et al., 2005; Zvegintzov, 1984).

Ad (8). It is also noticed that the "individual prestige" (image or status effects) are different within the three discerned stages. Working with something new is suggested to have a high personal image or status effect, while working with something similar is considered to have a lower personal image or status effect. In Table 23 these decision making characteristics are summarized.

	Inception decision of a new IS	Maintenance decision of an operational IS	Abandonment decision of a legacy IS
(1) Management level of the involved decision maker	Higher management decides, it is a strategical decision	Middle management decides (decision is delegated to middle management), it is an operational decision	Higher management decides if it is a strategical decision (large legacy IS), middle management decides if it is an operational decision (small legacy IS)
(2) Organizational risks involved	High risks when it is a large system	Low risks for functional updates	High/Low risks
(3) Financial implications of the decision	Zero-based budgeting (increasing budget) No financial history of the maintainability of the system available (looking forward and predicting cost)	Incremental budgeting/ Annually budget (incremental or decremental) Some financial history of the maintainability of the system available (looking backward and forward estimating cost)	Decreasing budget Financial records available (looking backward, based on history determining cost savings due to abandonment)
(4) Effect on operations	Something new and extra, major change	Something more of similar, minor change	Something less
(5) Technological implications	New technology	Same technology	Old technology
(6) Operational process or project-based	Programme/project	Operations/local project	Programme/project
(7) Documentation quality	Up-to-date	Inadequate	Inadequate or even absent
(8) Individual prestige (image or status effects)	High	Lower	Not known

Table 23. Decision making characteristics throughout the IS life

In summary, based on the life cycle literature and the different decision making characteristics in the three phases of the IS life cycle, it seems logical to design separate decision making methods for each stage. This assumption will be validated in the next Chapter 7 (p. 101).

In the following section a decision making method is conceptualized. First step is to conceptualize a method to identify the legacy IS, subsequently dilemmas on designing a method to abandon legacy IS are discussed.

2. Decision making at information function, IS or application level?

In Case A three hierarchical levels of decision making were described (see Section 5.6, Table 17, p. 72): the information function level, the IS level and the application level. These levels have an interaction with permeable boundaries (Harrison, 1999). The dilemma is: should there be one overall decision making method for all three levels or three separate decision making methods for each level? In the literature methods are found on the lowest most detailed level. For example Benson et al. (2004) distinguish decision making at application level and hardware level. The IS level decision making is described by for example Comella-Dorda et al., 2000; Warren, 1999. Concerning information function level no references were found in the literature. For practical reasons, due to the fact that there are permeable boundaries (Harrison, 1999), when the IS level is selected, the higher level (information function) and the lower level (e.g. application) are taken into account. Therefore this level of IS decision making is suggested. An example is provided by Warren (1999) who describes the assessment of legacy IS. Warren (1999) distinguishes a technical viewpoint in which hardware, support software and application software attributes of legacy IS are scored versus business value attributes (including business importance and expected lifetime) of the legacy IS.

3. Financial versus non-financial evaluation aspects?

This section elaborates upon the dilemma: should evaluating legacy IS decision making focus on financial aspects, non-financial aspects or a combination? Financial evaluation has the advantage that money is the

common dominator of all attributes. By calculating net present value, decision making is a well-accepted method (Brealey & Myers, 1991; Sakthivel, 1994). For example, when there is an abandonment trigger due to the fact that there is an opportunity which gives similar functionality for a much lower price, then the instrument of a financial evaluation is the right instrument. The decision is then more or less a comparison of the current cost, with the future cost including the project cost to switch from the current situation to the future situation. An example could also be that the current client base is diminishing; this will lead to similar IT costs divided by fewer clients, making the IT cost per client in relation to the delivered services not beneficial anymore. In Case A this was suggested by Business information manager (6).

It can be argued that an abandonment decision (a dis-investment) is more easily evaluated than an investment decision in a new IS, due to the fact that there is cost history of a legacy IS compared to unknown estimated future cash flows of a new IS. IT investments have bad reputations for the investment part and the maintenance part, and Berghout and Nijland (2002) suggest that the maintenance cost is often overlooked. This suggests that the financial part of a legacy IS abandonment decision is relatively easy to elaborate, especially when an IS is abandoned and is not replaced.

Besides financial aspects also non-financial aspects exist. For example in Case A, due to a merger five IS were abandoned. Financial criteria were only one part of the decision. For instance, how do you value a new system with new technology, against an old system that worked for 30 years and can handle large volumes? How do you value the technical quality or the availability of documentation or IS knowledge by support staff? In legacy IS abandonment decisions, organizations were confronted with knock out criteria, for instance year 2k readiness. These were non-financial arguments which were so important that they were mandatory. These were often non-financial criteria which should be weighted in a decision to abandon a legacy IS. It is suggested that a legacy IS abandonment decision making should include financial and non-financial elements.

4. Many or few attributes in the abandonment decision making method?

The design dilemma for a method to abandon a legacy IS, is to include all (more than a hundred) legacy IS aging factors and legacy IS abandonment triggers and be accurate (which will make it very complex due to the large number of IS aging factors [see Appendix C.1, p. 185]), versus including a few legacy IS aging factors and abandonment triggers, being less accurate but better to handle? According to Harrison (1999) different interdisciplinary approaches to decision making may be viewed as decision making models because they represent a particular segment of the real world at a given time and place under varying conditions. Because of the almost infinite number of variables in decision making and because these variables have varying degrees of complexity, a model with a small number of causal variables that are both significant and understandable must be developed (Harrison, 1999). If insufficient or incorrect variables are included, the model will not function as the real world phenomenon does; if too many variables are included, even when they are the right ones, the models complexity will work against understanding (Harrison, 1999). So for practical purposes, concerning the decision making to abandon legacy IS, it is suggested that there is an IS attributes database with generic information and that added to this generic information are business value attributes and the legacy IS abandonment triggers, which are identified in the literature and the explorative research (see Appendix C.3, p. 193).

5. Life cycle or portfolio oriented legacy IS assessment?

In this section the design dilemma is elaborated whether to assess a single legacy IS based on an individual IS life cycle or a portfolio of legacy IS? In Section 3.2, (p. 21) the difference between individual life cycle assessment and portfolio assessment is described. Every IS asset, consuming scarce resources should have a life cycle plan, including a plan to retire an IS. An asset IS management plan should be available (Kyte, 2008). These individual IS life cycle plans are written from the IS perspective; a portfolio approach is from a more holistic perspective and makes it possible to compare assets in an organization by ignoring details. Sabbagh (2008) concludes that the concept of analyzing applications in a portfolio could be interesting as it provides a level of analysis one step up from the individual application analysis as in application life cycle management. Both IS management methods provide valuable information. Therefore, instead of choosing between the two methods, it is argued

that these methods should be used complementary to each other. Input of an IS portfolio analysis can be the individual IS life cycle information and vice versa. Both methods benefit from the IS attributes database and can be used to fill in the IS attributes database. Part of an IS attributes database concerns the IT asset management repository (ITAM). This ITAM repository includes (and cross references): procurement, contract, inventory, maintenance, entitlement management and retirement information for the software and the hardware that a company owns (Hunter & Aron, 2006). It is further argued that IS life cycle management is done by relative lower management, having a more restricted view, sometimes of one IS only, whilst IS portfolio management is handled by relative higher management, who have more overview, and the ability to compare different IS on different IS attributes. In general, the attributes database gives organizations the possibility to manage the IS (examples are provided by Benson et al., 2004; Maizlish & Handler, 2005; Gottling & Torgnysdotter, 2002; Van den Heuvel, 2007; Warren, 1999). A comprehensive list of legacy IS characteristics is provided in Appendix C (p. 185). For IS abandonment decision making an IS portfolio approach is suggested.

6. Many stakeholders versus few stakeholders involved in legacy IS abandonment decision making?
More and different stakeholders (e.g. functionality, economics and technology) will bring additional knowledge, and possibly additional solutions to a problem. Participation also increases the general acceptance of the final choice and better comprehension of the decision (Harrison, 1999). However, involving more stakeholders also introduces social pressure, acceptance of solutions, individual domination and winning the decision (Harrison, 1999). Within the explorative case eleven different types of stakeholders participated in the abandonment decision making of legacy IS (see Table 20, p. 76). In Case A, a medium sized group of 5-15 members was preferred. From the explorative case organization is also conceptualized that the IS owner (e.g. director, business unit director) is always accountable for the decision to abandon a legacy IS.

6.6 Summary and conclusions

Based on the literature as presented in Chapter 3 and Case A (Chapters 4 and 5), this chapter describes a legacy IS aging model and contains a discussion of the possible outline of a legacy IS abandonment decision making method. First the complexity of legacy IS aging is investigated, 140 legacy IS aging characteristics are distinguished, which may refer to the IS or its constituents and are further categorized into internal or external aging factors. When the accumulation of aging factors passes a legacy IS aging threshold, the legacy IS needs re-engineering or should be abandoned. When the legacy IS is re-engineered, it is assumed that the overall age is reduced and the life of the legacy IS is extended.

Furthermore, 32 legacy IS abandonment triggers were identified. These triggers were mentioned by interviewees as triggers to abandon legacy IS. Due to the fact that interviewees mentioned 13 similar legacy IS aging factors and legacy IS abandonment triggers, it was concluded that both characteristics overlap and complement. It was further suggested that abandonment triggers pass a legacy IS aging threshold which make management of organizations decide to abandon legacy IS. Subsequently, good practices concerning abandonment decision making and the practical abandoning of legacy IS were described. It was suggested to have an IS asset database, which consisted of the IS and their IS legacy attributes (legacy IS abandonment triggers). Periodically an assessment should be done and all the IS are scored on the legacy IS abandonment triggers. Subsequently, six design dilemmas for a method to abandon legacy IS were addressed. These are:

1. Enhancing existing IS decision making methods or designing a new legacy IS abandonment decision making method? Based on the life cycle literature and on all the different decision making characteristics in the three phases of the IS life cycle, it seems logical to design separate decision making methods for each stage, because current IS investment or maintenance decision making methods hardly address predominant abandonment triggers. Further a distinct decision making method for each life cycle phase is suggested due to the fact that the decision making characteristics are different within: (1) management level of the involved decision maker, (2) types of risks involved, (3) financial implications of the decision, (4), effect on operations, (5) operations or project, (6) technological implications, (7) documentation quality and (8) individual prestige (image or status effects) of being involved in this type of decision making.

2. Decision making at information function, IS or application level? It was argued that a method at IS level is to be preferred.
3. Financial versus non-financial evaluation aspects? It was argued that the evaluation of legacy IS should also include non-financial aspects.
4. Many or few attributes in the abandonment decision making method? It was argued that a substantial amount of attributes need to be identified and maintained. Besides generic IS attributes, this includes business value attributes and the legacy IS abandonment triggers.
5. Life cycle or portfolio oriented legacy IS? It was argued that decision making can be based on an individual life cycle base or by means of a portfolio approach. Both methods are valuable for organizations and fulfill requirements for different levels of management. Within the explorative case organization a portfolio approach was used.
6. Many stakeholders versus few stakeholders involved in legacy IS abandonment decision making? It was argued that a formalized decision to abandon a legacy IS should be made by 5-15 stakeholders with different viewpoints (e.g. business, technology and finance stakeholders) should apply.

These dilemmas are further addressed in the additional case studies, described in Chapter 7.

The legacy IS abandonment decision making is formalized by a multiple discipline group of stakeholders (e.g. functional, technical and economical). The group size of decision making (the number of members) is between five and 15 members (an unequal number is preferred). The IS owner (e.g. director, business unit director) is always accountable for the legacy IS abandonment decision and the practical abandoning. Sponsorship of the highest board is suggested as there are a lot of opposing forces not to abandon legacy IS. A programme for practically abandoning legacy IS with specialized skilled staff members is suggested.

In the following chapter, the aging process, the abandonment decision making process, the practical abandoning process of legacy IS and design dilemmas for a method to abandon legacy IS are validated.

PART 3: TESTING PHASE

7 VALIDATING CASE STUDY RESEARCH

"There are two possible outcomes: if the result confirms the hypothesis, then you've made a measurement. If the result is contrary to the hypothesis, then you've made a discovery".
---Fermi, E. as quoted by Jevremovic, T. (2005)---

7.1 Introduction

As described in Section 2.8 (p. 17) this study follows the research outline described by Van Engelen and Van der Zwaan (1994). This chapter describes the testing phase in which the aging process, the abandonment decision making process and the practical abandoning process of legacy IS are validated. Subsequently design dilemmas for a method to abandon legacy IS are validated. First the research model for validating cases is described, this includes the validation research questions and sub-research questions (Section 7.2). Subsequently, the four case organizations are introduced (Section 7.3) and the analysis process is described (Section 7.4). Then the aging process (Section 7.5), the abandonment decision making process (Section 7.6) and the practical abandoning process (Section 7.7) of legacy IS are described. Finally, the validation of the design dilemmas (Section 7.8) are described and this chapter ends with a summary and conclusions (Section 7.9).

7.2 Research model for validating cases

In this chapter the concepts as presented in the explanation phase (see Chapter 6, p. 87) are validated in four subsequent cases. These four cases were selected by snowballing. Overall, this research consist of one explorative longitudinal research at a public organization and four validating cross-sectional researches (at one private organization and three public organizations).

The validating research model is divided in two sections and subsequently follows an inductive and a deductive approach (see Section 2.3, p. 13). Integrating these two approaches is encouraged by Lee (1991) and Robey (1996). The first section of the research model is similar to the research model of the exploratory case in Chapter 5. The first section again researches the three predominant life cycle stages, each predominant life cycle includes a research question. Researching these life cycle stages will built further on the results of the former explorative case research and is mainly deemed inductive (the results might confirm, contradict or enhance earlier results of the explorative case). Similar to Chapter 5 the same research questions are answered, the SRQ answering these research questions are however further refined. Some SRQ are added based on the results of the explanatory case. Some SRQ, approved to be less useful. To distinguish between the SRQ of the explorative research and the validation research, continued SRQ numbers are used. The numbering in this chapter, therefore, starts at SRQ 14. The following adjuvant life cycle stages, research questions and sub-research questions (SRQ) are derived.

The first life cycle stage identified was: (1) *the aging process of legacy IS*, the first research question was:

Research question 1:
How do legacy IS age?

In order to answer this research question, in the explanation phase (see Section 6.2, p. 87), based on the literature (see Section 3.4, p. 30) and Case A (Chapter 4 and 5), legacy IS aging was conceptualized. This means that in this dissertation legacy IS characteristics were categorized into internal aging factors (within the boundaries of the system and could be influenced by organizations), and external aging factors (outside the boundaries of the system and could not be influenced by an organization) (see Section 6.2, p. 87). In order to, further investigate the internal and external aging factors the following SRQ were defined (the SRQ were supported by a questionnaire, see Appendix B.1, p. 178):

- SRQ 14: What are internal legacy IS aging factors?
- SRQ 15: What are external legacy IS aging factors?

The next logical life cycle stage concerned the decision to abandon legacy IS (Berghout & Nijland, 2002). This second logical process concerned: (2) *the abandonment decision making process of legacy IS*, the second research question was:

Research question 2:
How do organizations decide to abandon legacy IS?

In order to answer this research question, in the explanation phase based on the literature (Chapter 3) and Case A (Chapter 4 and 5), legacy IS abandonment decision making within organizations was conceptualized (Chapter 6) and validating SRQ were defined. From Case A (Section 5.6, p. 68) and the literature (Section 3.5, p. 33) different reasons were found why organizations abandon legacy IS. These reasons were conceptualized in the explanation phase (Figure 24, p. 90). In order to further enhance and validate the legacy IS abandonment triggers the following SRQ 16 was defined:

- SRQ 16: What are triggers to abandon legacy IS?

From Case A (Section 5.8, p. 80) and the literature different legacy IS abandonment decision making perspectives were found, these were conceptualized in the explanation phase (Section 6.3, p. 90), in order to further enhance and validate the legacy IS abandonment decision making perspective the following SRQ 17 was defined:

- SRQ 17: What legacy IS abandonment decision making perspective (e.g. technical, functional, economical) is predominant?

From Case A (Section 5.8, p. 80) it was found that there were differences in abandoning commercial off-the-shelf (COTS) compared to a proprietary (own build) IS. These differences were conceptualized (Section 6.3, p. 90) and were validated by the following validation SRQ 18:

- SRQ 18: To what extent does a commercial off-the-shelf (COTS) or a proprietary (own build) IS influence the characteristics of abandonment decision making?

From Case A (Section 5.8, p. 80) it was found that there were differences in abandoning IS supporting primary business processes versus IS supporting secondary processes. These differences were conceptualized (Section 6.3, p. 90) and were validated by the following validation SRQ 19:

- SRQ 19: To what extent do IS supporting primary business processes versus IS supporting secondary processes influence the characteristics of abandonment decision making?

From Case A (Section 5.8, p. 80) it was found that the size of an IS (small or large) influence the abandonment decision making. These differences were conceptualized in Section 6.3, (p. 90) and were validated by the following validation SRQ 20:

- SRQ 20: To what extent does the size of an IS (small or large) influence the characteristics of abandonment decision making?

In Case A, (Table 21, p. 77) good practices in legacy IS abandonment decision making were identified (by means of repeated experiments by the organization). These good practices were conceptualized in the explanation phase in Chapter 6 (see Section 6.3, p. 90). In order to confirm, and further enhance the good practices the following SRQ 21 is defined:

- SRQ 21: What are good practices in legacy IS abandonment decision making?

The next logical life cycle stage concerned the practical abandoning process of legacy IS (Berghout & Nijland, 2002) (see Figure 11, p. 24). This third logical process was named therefore: (3) *the practical abandoning process of legacy IS*, the third research question was:

Research question 3:
How do organizations practically abandon legacy IS?

In order to answer this research question, in the explanation phase (Section 6.4, p. 92) based on the literature and Case A (Chapter 3, 4 and 5), practically abandoning legacy IS including good practices were conceptualized and were validated by the following validation SRQ 22:

- SRQ 22: What are good practices in practically abandoning legacy IS?

These three life cycle stages together will provide input (such as good practices) for the second section of this validation research model which is to (4) *propose a method to abandon legacy IS*. In this section design dilemmas for a method to abandon legacy IS are validated, this is considered to be a deductive approach. The research objective was:

Research objective:
Propose a method to abandon legacy IS

In order to achieve this research objective in the explanation phase design dilemmas for a method to abandon legacy IS were conceptualized (Section 6.5, p. 93). The following validating SRQ were defined:

In Section 6.5, (p. 93) it was argued that a new method to abandon legacy IS should be developed. This was based on the life cycle literature and on all the different decision making characteristics in the three phases of the IS life cycle. Decision making characteristics were different within: (1) management level of the involved decision maker, (2) types of risks involved, (3) financial implications of the decision, (4) effect on operations, (5) technological implications, (6) operations or project, (7) documentation quality and (8) individual prestige (image or status effects) of being involved in this type of decision making). This resulted in the following SRQ 23:

- SRQ 23: Enhancing existing IS decision making methods or designing a new legacy IS abandonment decision making method?

In Section 6.5, (p. 93) it was argued and conceptualized that a method at IS level is to be preferred, this resulted in the following SRQ 24:

- SRQ 24: Decision making at information function, IS or application level?

In Section 6.5, (p. 93) it was argued and conceptualized that the evaluation of legacy IS should include financial and non-financial elements, this resulted in the following SRQ 25:

- SRQ 25: Financial versus non-financial evaluation aspects?

In Section 6.5, (p. 93) it was argued and conceptualized that there should be an IS attributes database with generic information and that added to this generic information are business value attributes and the legacy IS abandonment triggers, as identified in the literature and the explorative research to abandon legacy IS (see Appendix C.3, p. 193). This resulted in the following SRQ 26:

- SRQ 26: Many or few attributes in the abandonment decision making method?

In Section 6.5, (p. 93) it was argued and conceptualized that the decision making concerning legacy IS can be based on an individual life cycle approach or by means of a portfolio approach. Both methods were valuable for organizations and fulfill requirements for different management levels. For IS abandonment decision making an IS portfolio approach is suggested. This resulted in the following SRQ 27:

- SRQ 27: Life cycle or portfolio oriented legacy IS assessment?

In Section 6.5, (p. 93) it was argued and conceptualized that the formalized decision to abandon a legacy IS should be made by 5-15 stakeholders with different viewpoints (e.g. business, technology and finance stakeholders). This resulted in the final SRQ 28:

- SRQ 28: Many stakeholders versus few stakeholders involved in legacy IS abandonment decision making?

In this testing phase the validation is done by means of empirical case research in four organizations. The SRQ are transferred into case study questions, described in Appendix B, p. 178 (see Table 50, p. 183). Also in this appendix the validating analysis model is presented (see Table 51, p. 184). The validating analysis model illustrates the relation between the elaborated life cycle stages, the research question, the SRQ and the questions in the questionnaire. The Case Study Protocol (CSP) for the validating cases follows a similar design as the explorative CSP (see Section 5.3, p. 56).

7.3 Overview of the validating case organizations

Four different case organizations were investigated during the period ultimo 2010, until 2012. Overall fifteen stakeholders were interviewed. An overview of the key characteristics of the four different case organizations including the interviewed stakeholders is provided in Table 24.

Organization	**Case B**	**Case C**	**Case D**	**Case E**
Number of employees. ICT staff employed.	3,000 350	9,000 600-800	1,600 180	1,200 70
Total revenue.	€ 1,500m	€ 5,200m	€ 1,500m	€ 500m
Total capital spend in Euro.	€ 280m	Not provided	€ 150m	€ 3,000m
Yearly ICT spend.	€ 20-60m	€ 250m	€ 75m	€ 24m
Stability of ICT spending in last 3 years.	No	Yes	-2%	Yes
% of ICT spend outsourced.	10-20%	70-90%	80%	>90%
No. of IS in the organization.	250	2000	100	90
Interviewed stakeholders.	CIO/ICT director (2)	IS owner (2) (director)	Senior user (3)	Business information manager (9)
	Consolidation project manager (3)	Business information manager (11)	Functional maintenance manager (3)	
	Application manager (1)	Business information manager (12)	Business information manager (10)	
	Rationalization project manager (2)	Functional maintenance manager (4)		
	Architect (6)	Rationalization programme manager (1)		
		Rationalization project manager (1)		

Table 24. Overview of the four validating case organizations including interviewed stakeholders, between parentheses their unique number which will be used for quotations

In the next sections, the four validating case organizations are described. For each case study, first an introduction of the case organization is provided and then processes concerning, inception, maintenance, abandonment and IS management are described.

7.3.1 Case study B

Case B was a provider of cable television, broadband internet and telephony services to 3.3 million households and businesses. Case B emerged from three former organizations (X, Y and Z). These former organizations had their background in the public utility sector (energy suppliers). The former three organizations were bought by two equity firms. They envisioned a single more efficient and effective new organization, and were prepared to invest in this new organization and subsequently consolidated these three organizations within a time frame of two years.

Part of this consolidation included merging the three information provisionings of the organizations containing information functions supporting the individual business units, each information function consisting of many IS. The three initial organizations were quite different in scale, structure, processes, systems and customer satisfaction. Smallest was organization Z with 300k clients, then there was organization X with 1.2m clients and organization Y with 1.7m clients.

At the start of the consolidation, in November 2006, quartermasters from the former organizations made an initial scan of the core systems. Core systems of any telecommunications company are Customer Relationship Management (CRM) systems and Billing systems. The CRM and Billing IS from each former organization were assessed on aspects such as scalability, costs, risks, functionality and technical quality. Evaluation of these core IS, from these three former organizations was performed by means of a detailed multicriteria method. The IS from the smallest organization (organization Z) were unable to handle the consolidated business volumes (and were also not scalable). The IS of organization Y were still in the inception/developing phase and were not yet fully implemented and operational and were also lacking business functionality. Implementing a completely new IS would not be possible in the given short time frame. Organization X core IS had been developed only 1.5 year earlier and were fully operational. These core IS were subsequently preferred and new instances of the organization X core IS were created. First organization X, then organization Z and finally organization Y were migrated to these new core IS and the former IS were successively abandoned.

In the new Case B organization business processes supported by information functions were consolidated in streams by business project management teams. Each consolidation focused on the consolidation of business processes (business streams) and supporting processes (e.g. HRM stream, Finance stream and ICT stream). All business project management teams reported to Integration Programme Management (IPM). Part of this consolidation process was the abandoning of hundreds of IS.

In 2007 a new CIO was hired and also an architecture board was initiated. The architecture board consists of architects and was headed by the CIO. The architects formulated architecture principles and guideline standards. A generic telecom business process model was used being the “Enhanced Telecom Operations Map” (eTOM model). The architecture board subsequently made an inventory of all IS and plotted these applications on the generic eTOM business process model. The architecture board identified approximately 200 larger applications within the three former organizations. These 200 large applications were inquired and decisions were made to abandon 133 applications and to retain 67 applications.

Decision criteria included the functionality of the IS, underlying technology and whether it was able to support the larger volumes of the Case B organization. These decisions were handed over to the business project management teams (these decisions by the architecture board were not made in isolation, due to the fact that the architects were also involved in the business project management teams as an important advisor to the stream manager). If the business project management team did not agree with the proposal of the architectural board, there was a possibility to submit a decision paper with an alternative plan to the IPM. The whole process was described by the Consolidation project manager (3) as a pragmatic approach rather than a scientific approach and included a lot of communication and discussion. The streams were responsible for implementing the retained IS and abandoning the old IS.

Subsequently in 2007 an Application Rationalization Programme (ARP) was incepted by the CIO. By digging and inventorying the ARP found another 1,050 applications[11]. These were very small applications, including applications nobody thought of or knew of (ARP found that every major application was related to some smaller applications like tooling). With the help of the architect the ARP facilitated the decision to abandon 783 applications with business cases per logical application group. Again the decision was made that the business streams were responsible for implementing the retained IS and abandoning the old applications. The ARP formally only facilitated the process of practical abandoning.

The chronological overview of the consolidation of the three former organizations into the new Case B organization is illustrated in Figure 26. In the consolidation phase of Case B several hierarchies of decision making in time were discerned. First at the highest level there was the acquisition of the three former organizations (organization Z, organization Y and organization X) by the two equity firms and the decision to consolidate organization Z, organization Y and organization X. Secondly in November 2006 a high-level pre-consolidation research was performed by quartermasters. These quartermasters did initial scouting and advised unanimously to use the organization X core IS as a platform for the new consolidated organization. Thirdly, after the decision to choose for the core IS, the architecture board proposed a rationalization regarding the larger applications of which 67 were continued and 133 were abandoned. Finally there was the ARP to practically abandon the IS (remove applications, procedures and the hardware) and by doing so this programme found another 1,050 small applications of which about 783 applications were abandoned. The streams were responsible for implementing these decisions and reported to IPM. The decision making process is illustrated in Figure 26. The figure illustrates a timeline and the hierarchical levels of decision making.

[11] Besides this number of applications, also much higher numbers of applications were mentioned by several interviewees, e.g. 1,600, 2,000 and 2,400 applications in total. These numbers include all very small personal productivity applications, spread sheets, like version A, B and C of Excel, Access databases and so on. By combining these local versions to one final version and abandoning applications nobody wanted to have, the number of applications was reduced to about 1,200 applications. The personal productivity applications are not in the scope of this research.

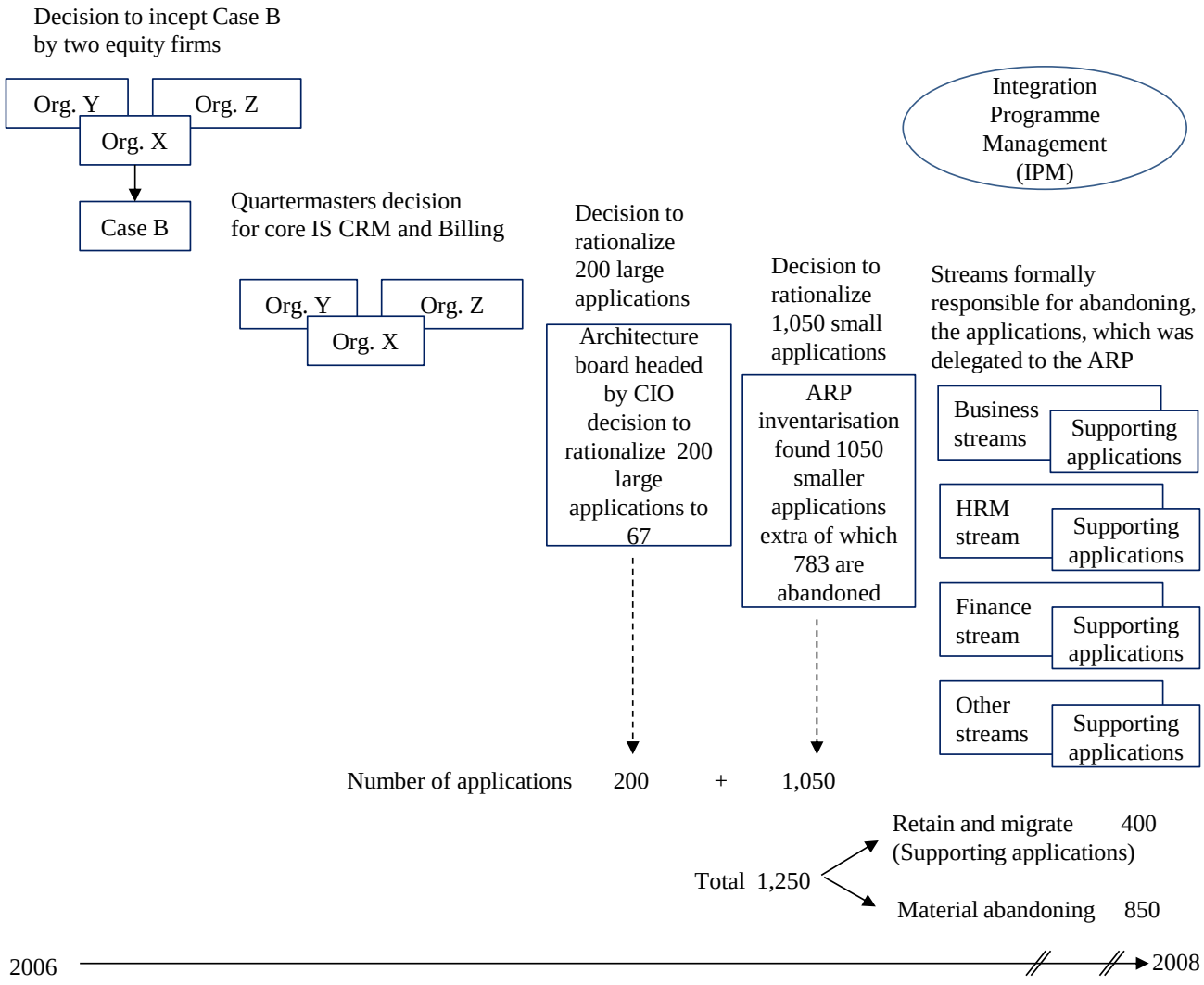


Figure 26. Timeline and decision making

Inception at Case B

Initially Case B's most senior managers and the strategy department planned their predominant overall objectives for the period up to 2016. Based on these objectives, the Development director created the initial IT architecture, which was approved by the board. This IT architecture included high-level descriptions of predominant IT components and most important IS. Investments in IS were subsequently managed through the Prince 2 method, where for each project a *project mandate*, *project proposal* and *project initiation document* (PID) were established. All projects were steered by the project board, which reported to the board. The board was responsible for inception decisions. The total budget for investments was fixed, which implied that priorities were continuously monitored. The other stakeholders in the IS inception process include the Strategy Department, CIO, Finance Department, business managers, Development director, Application management director, external advisors and several senior users.

Maintenance at Case B

For the maintenance process a roadmap and a yearly fixed budget were available. This budget was split over several budgets, for example customer demand, business continuity and sales. Business cases needed to be approved in order to utilize this funding. There was a planning period of six quarters and each quarter included

an evaluation by the portfolio board. This portfolio board included the board members of Case B and Vice Presidents.

Abandoning at Case B

Concerning the abandoning of IS during the consolidation of Case B the following sequence of events took place. As described before, first a decision was made concerning the core IS (CRM and Billing) for the consolidation of Case B. Subsequently the architecture board decided to rationalize 200 applications, which were handed over to the consolidation streams responsible for incepting the new organization. Then the ARP was incepted. The ARP started with the inventorization of the complete information provisioning of the three former organizations; they identified another 1,050 applications. This inventorization was made by means of desk research, interviews, contracts and tooling that researched the network. The information was put into an application database. The database contained the following information on the IS: application name, hardware, hardware location, data included, number of users or clients, business value, supplier, contracts, licenses, costs, number of users, functionality and application management cost.

Based on this information a proposition was made by the ARP with the help of the architects on which applications could be continued and which applications could be abandoned. These decisions were discussed with the streams. The stream managers became responsible for the abandoning of the applications.

Due to the fact that the focus within the business streams was on developing the new organization, including migrating data to the new system, there was less attention for abandoning old applications. According to Architect (6): *"it was celebrated when the new system was operational and fully working, not when the last application was abandoned".* There was less necessity due to the fact that the budgets for operating these old systems were with the ICT department and not with business departments themselves.

To keep the abandoning of applications weekly on the agenda, a team member of the ARP was added to each business stream. ARP also constantly reported about the status of abandoning applications to higher management and asked for continuous pressure on the business streams to abandon applications. This process cost a lot of time and energy. According to Application manager (1): *"It seemed like an IT party and the business did not want to join this party ... the business should have shown more initiative to users, stating not to use old systems anymore, establish real ownership of abandoning, a suggestion would have been to include budget incentives to abandon applications for the business".*

ARP facilitated a working process for abandoning applications. ARP used the following procedure to practical abandon old systems: first, they informed business managers, subsequently the application was disconnected and there was a waiting period of four weeks. During this period users were unable to access the application. Subsequently the hardware was disconnected. In case ARP missed something, for instance, an interface, ARP would be able to restart within one hour. ARP always provided plans for the data: abandon, store, or keep accessible (occasionally, Case B had to keep data because of compliancy requirements; therefore, some applications were kept running on low performing platforms in case the governmental tax and revenue representatives would come controlling).

Due to the accuracy of this process ARP earned trust in the organization. Sometimes there were discussions about the decision to abandon an IS. By means of the information in the extensive application database, discussions were made transparent. For example if the IS costs € 150k and there are only 20 users, why should you keep this IS? The ARP application database required substantial up-front effort, but this paid itself back later. The ARP also maintained a change procedure in case an IS (application and hardware) labeled for abandoning could not be completely abandoned. This change procedure implied that the IS owner (e.g. director, business unit director) of the IS had to write a business case, or another decision paper, including goal, scope, approach, project and budget, planning and risk management. ARP also provided support in writing this document.

IS management at Case B

Concerning IS management, Case B first decided between the available funding for IS and non-IS investments. After the consolidation, Case B made "buying" market share their first priority. This implied that investment in new IS were minimized and preferably IS of any of the three consolidation partners were selected. After this consolidation in 2008, Case B went into a post consolidation phase. In this phase business cases could be made for new IS investments and there was also a roadmap for the existing IS. Case B initiated a form of IS portfolio management, in which certain business domain managers were given authority to decide which maintenance to perform. In Appendix F (p. 205) an interview with a CIO who was responsible for the abandoning of legacy IS is provided.

7.3.2 Case study C

Case C was a public organization and was divided into 10 regional departments, five centers of excellence and one project organization. As of January 1, 2006, Case C was an independent executive agency. Until 2007 the responsibility for ICT was decentralized and a lot of (regional) business departments had their own ICT department. In 2006, there were more than 15 independent (regional) business departments. And all these independent (regional) business departments had their own hardware, data center and software applications. This resulted into a lot of redundant technology, such as hardware, software, licenses and applications.

In 2007 Case C centralized into six information functions (domains), supporting the five centers of excellence (one center had two information functions). Case C applied the BISL framework (Pols & Backer, 2007) for demand specification and ITIL (OCG, 2007) and ASL (Pols & Backer, 2006) for supply specification. In 2007 the newly founded center for Data and ICT named Data and ICT Department (DID) was made responsible for the reduction of decentralized data centers and applications.

Case C management envisioned to first reduce the overall number of 2,200 applications to 1,500 applications. This step was called Technical Application Consolidation (TAC). In TAC all different application versions of all independent (regional) business departments were upgraded to one single version, migrated to the centralized data center and old versions of applications and local data centers of (regional) business departments were abandoned. Business departments hardly noticed this step.

The second step was to go from 1,500 applications to 1,200 applications. In this step redundant functionality of formerly decentralized applications was eliminated (called Functional Application Consolidation, FAC). This second step also included involvement of business managers. The differences between regional departments were substantial. In one case a particular regional department was using 10 applications for similar business processes, where another was using one. As a baseline an application list had to be initiated. This list was hosted centrally in an application management database named COMAS.

The third step, which took longer, was to further rationalize the application portfolio and included outsourcing and the rebuilding of applications. The number of applications was reduced from 1,200 to 800. The application consolidation process of Case C is illustrated in Figure 27.

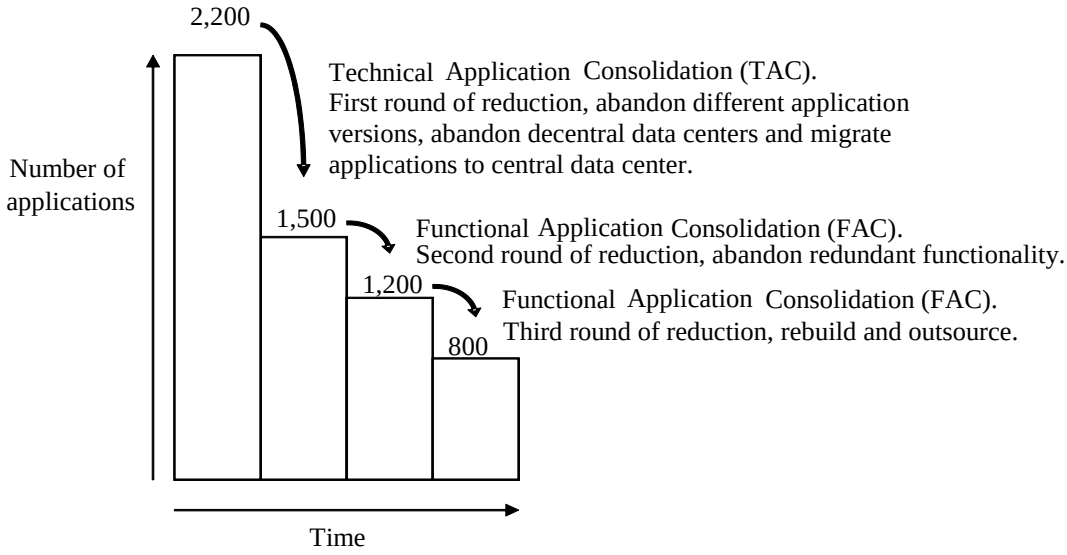


Figure 27. Case C application consolidation process

Inception of IS at Case C

Concerning the inception of IS, there were five centers of excellence: Center for Corporate Services, Data and ICT Department (DID), Center for Infrastructure, Center for Transport and Navigation and Center for Water Management. DID was responsible for all the hardware, data center, network, personal computing and telephony. The centers of excellence were responsible for the inception of new IS. This task was effectuated by advisory groups for each center of excellence.

A typical advisory group consisted of four regional departmental directors, one representative from the executive board, two representatives of process and systems and one representative of ICT. The inception of new IS started with a request (e.g. from a user, a director). From these requests, business cases were developed. These business cases were evaluated by the advisory group. Part of this evaluation concerned whether the projects fit with the long term strategic plans and short term availability of resources. There was no competition with non-IS investments and decision making was based on a majority of votes.

Besides these advisory groups in the centers of excellence there was also a higher hierarchy advisory board: "advisory group information provisioning". This advisory board evaluated all business cases received form individual advisory groups of excellence centers. They evaluated from an ICT perspective and, for instance, evaluated fit with the overall ICT architecture. They advised the CIO/CFO (a combined function within Case C). Final decision making took place in a meeting with all the directors and was headed by the CFO/CIO. These decisions required consensus and were also compared with non-IS investments.

Maintenance of IS at Case C

The Business information administrator would notice that certain functionality was needed for the business. Then the Application manager would be responsible for making a release plan. A yearly release plan and a yearly fixed budget for maintaining the IS were available. During the year on the basis of new insights the release plan was updated. Several databases were used to support the maintenance (e.g. business information management of the IS was performed in a database named Troux; the application management was supported by an application named COMAS and the technical infrastructure management was supported by a package named Topdesk).

All these databases had their own perspective. However, the application name was the key within all three databases. ARIS was used as a tool for process modeling and made it possible to illustrate business processes with functional, application and technical information. In case the maintenance of an IS was substantial, a business case had to be written and it followed a process similar to inception.

The managing director of a center of excellence was responsible for the decision on which IS to maintain. The DID was responsible for technical upgrades at hardware level. There was no prioritization with non-IS investments. In the maintenance phase the role of the CIO/CFO was minimal, except in the case of substantial maintenance when the business case process for inception was followed.

Abandoning of IS at Case C

Concerning abandoning of IS, in Case C, three different abandonment initiatives were identified: (1) the normal abandoning process, (2) the process in one specific center of excellence by an advisory group and (3) the IS Rationalization Programme.

Abandonment initiative (1): the normal process to abandon an IS was as follows. The Application manager would take the initiative that an IS was to go “end of live”. Reasons could be twofold: the Application manager knew via the Business information manager that the IS was no longer needed, or the Application manager noticed that there was no demand for the IS anymore. In this case he would report back to the Business information manager. The trigger for abandoning, was for, example aging hardware or older software versions. This normal abandonment process was not yet in use during this study (year 2012).

Abandonment initiative (2): the process within one specific center of excellence. In the past (2005) each regional business unit had its own applications. The advisory group of the center of excellence reduced about 500 applications to circa 180 applications. This rationalization was described as follows: first they identified which applications existed. The next step was to find out who was using the system and what the problem would be if the system was disconnected. It took a lot of effort to find all the applications. It was also noticed that abandoning applications was lower on the priority list of the stakeholders compared to inception. Finally, 320 applications were abandoned, which were supporting systems. Also three operational systems were abandoned. This was a complex task, because due to legal requirements, Case C had to keep parts of the data. Case C also stored data about physical roads, roads that might be 100 years and older. Therefore, Case C needs to store the data for very long periods of time. It was not always efficient to abandon the entire application and to burn the data on a CD. It could also be beneficial to have the application running on a bare minimum, just for maintaining access to the data.

Abandonment initiative (3): The DID was incepted in 2007 and started in 2010 with an IS Rationalization Programme (ISRP). At the start, a high-level top down inventory was made, in which it was estimated that Case C could go back from 2,200 to 800 applications. This inventory was performed by network tooling. This resulted in technical information like the number of applications. Other information (for example the functionality of the application or the local contracts with suppliers or the name of the users or the owners), however, was often not known. Aim was to go from 2,200, to 1,500 to 1,200 to 800.

The strategy of DID was as follows. First, a Technical Application Consolidation (TAC) was performed: different versions of applications within local data centers were migrated to one version on a centralized data center. This enabled a reduction from circa 2,200 applications to 1,500 applications; also local data centers and contracts with suppliers were abandoned. Then a Functional Application Consolidation (FAC) was performed. Although ISRP was a DID initiative, it seemed to be a success factor that it was the business who wanted to have fewer applications and not DID. DID only offered help and did the coordination.

DID asked the board to assign the task to reduce these 1,500 to business managers. DID would then provide generic methods to establish this reduction and help them. In Q1 2011 the CIO/CFO sent a letter to the business,

in which the business was made responsible for reducing the number of applications and per information function a Business information manager was made responsible. The responsible Business information manager subsequently formed a project team. The project leader was somebody from ICT, and this person added expertise by means of linking architects and product managers and by knowing what relevant information about the application was needed. Information about applications was collected and stored in an application repository. By collecting this data more applications were found and stored in the application repository. In advance selection criteria for applications had been formulated from an ICT perspective, such as: cost, maintainability, continuity, technology perspective and also criteria from a business perspective, such as: user friendliness and number of users. These criteria were approved by the architects.

Subsequently, the applications and the functionalities were plotted on the work processes, which were modeled in ARIS. This method clearly indicated that there were more redundant applications with similar functionality. Based on this information and the selection criteria a target application portfolio was created, which was plotted on the business processes. This target application portfolio was ratified by the board. This was important and considered a success factor as nobody within the organization could restart a discussion. Based on this information first the TAC and then the FAC was followed. This was the plan in Q1 2011. In Q4 2012 the TAC (from 2,200 to 1,500 applications) was almost realized; the second step of the FAC, going from 1,500 to 1,200 applications was realized before 31 December 2012.

With regards to the effort to establish a repository of all applications, the following problems occurred. Some applications were claimed by several owners, but there were also applications that nobody wanted to own. The ISRP used the weighted importance principle; the owner that used the application most was allotted the application. At the start of the project the application managers in the business were able to add new applications on the network of Case C. However, ISRP wanted to reduce costs by abandoning former hardware, centralizing the data center and by abandoning applications. Subsequently, the ISRP technically restricted that new local applications could enter the network of Case C, unless these applications were approved in a centralized process. This was considered a success factor.

The practical abandoning took place as follows. From the application repository the ISRP knew which applications to abandon. The ISRP checked all for consistency reasons, double checked with information management and product management. Then the ISRP communicated the abandonment proposal to the business unit directors (representing the business process) and to the SLA managers (representing the users). Then the ISRP removed the rights to access the application and waited for two or three weeks. If nobody complained during this period the ISRP de-installed the application and archived the data (some of the data had to be kept by law compliancy for some period). And finally, which was very important, the ISRP abandoned the applications administratively, including the contracts with the suppliers, because the goal was to save money.

The workload for the business for these activities was estimated at 10 hours per application. Activities included finding out which users were on the system, administration, finding documentation and abandon or store the application. It was noticed that there was some reluctance on the business side; abandoning was not part of their culture. It was also low on their priority list. For the ISRP, however, it was important as ISRP wanted to dismantle the contracts with suppliers and reap the benefits of abandoning an application.

IS management at Case C

Concerning IS management, the centers of excellence applied a portfolio approach and compared all business cases by means of pre-set criteria. The portfolio management process was as follows. Portfolio managers assessed products and services of their business unit a number of times per year. Based on their prospects, maintenance budgets were allocated. A Life cycle approach at application level was not yet available (year 2012).

7.3.3 Case study D

Case D was a public organization and part of a ministry. There was a demand/supply relationship with another public organization, an agency X which was responsible for ICT within the ministry. Functional requirements were specified by Case D. Agency X was responsible for supplying infrastructure services, application maintenance and application development.

This research focuses on the abandoning of a single legacy IS named “BRS”. BRS contains factual data of companies (e.g. name, address) and their operational relationships (e.g. payment of subsidies). BRS included about 400,000 relationships.

The COBOL based BRS was incepted in 1991 and used to be an important IS. Many other IS were connected and merged within BRS. BRS was regularly improved and updated. For instance, there was “Rbs Cnnct”, which interfaced with the “Basic Administration System” of municipalities and the “Chamber of Commerce” database. In 2006, there were discussions concerning the complexity of the IS, including the interfaces and the costs. Besides these aspects, European law required that there should be only one master registry for European subsidy payments and it was decided to build a new system. This system was named EBS.

Inception at Case D

In general, Case D required business cases for board level inception decision making. Both IS and other investments were also prioritized at board level in Case D. More specifically, the decision in favor of EBS was also based on a business case and ample discussions concerning the complexity of the BRS, the interfaces and the costs. Besides this, European law required that there should be only one master registry for subsidy payments and, therefore, it was decided to build a new system. The business case/benefits document was written by a programme manager and the board of directors. The decision to build EBS was taken at board level. According to the interviewees no important stakeholders were missing in this inception decision.

Maintenance at Case D

The Business information manager divided his application landscape into three phases: (1) continuity, which means keeping the current functionality working; (2) improvement, which means adding additional functionality and (3) totally new functionality. The first category required most of the budget, the second was minimized in order to leave funding for the third category. The third category required business cases and included a form of application portfolio management and application life cycle management.

Abandoning at Case D

The board decided to abandon BRS because of cost and complexity considerations. When EBS became operational, the CFO asked whether it would be possible to abandon BRS. This was uncertain because of interfaces with other systems. Therefore, Case D preferred a process where the functionality of BRS was gradually reduced in favor of EBS. At the time of this research, only 3 of 40 tables remained in use. Furthermore, BRS data was stored separately for compliance reasons. The IS abandoning process in general was not formally described.

IS management at Case D

Concerning IS management, each year a budget was given for IS inception. By means of business cases the E-counsel would decide which projects were approved and rejected. It can be argued that there was a form of project portfolio management. Concerning application portfolio management, Agency X had a central application repository in which they kept track of life cycle items. Application portfolio management and application life cycle management were marginally available, and had to be developed further.

7.3.4 Case study E

Case E was a large public organization. The investments in ICT were relatively small (€ 3.5m), also compared with the other investments in this organization. Case E continuously evaluated legacy IS as part of their portfolio management process in which all application were regularly reviewed. During this research no specific legacy IS that was abandoned was researched, only the processes of inception, maintenance, abandoning and IS management were researched.

Inception at Case E

The process concerning the inception of IS was as follows. The business information managers yearly prepared information plans for their business unit, including new projects. All projects were included in the initial project calendar. The projects in the project calendar were subsequently prioritized through a multicriteria method. Criteria were: is the suggested project in line with the goals of Case E, is there enough staff capacity and what are the costs? The Corporate information manager assessed each project in terms of "Yes" or "No", including arguments. This advice was subsequently discussed in the IS steering group, in which all business unit information managers, the CFO, COO and CIO participated. This IS steering group made the final decision on which projects were to be executed. Subsequently, the business cases were updated and project leaders were assigned. The IS steering group subsequently received progress reports of all projects on a monthly basis. The CIO had an important role in this process, but the CIO reported to the CFO and formally the CFO was accountable.

Maintenance at Case E

Concerning the maintenance of IS, there were two streams. The first stream concerned small changes, for which there was a budget of € 250k. This budget was divided over the business units (each business unit had between € 30 - 40k); the business units were free to spend this on small changes.

The second stream was the stream of "major" changes. The budget was about € 1.2m in total for all the applications. The prioritizing and decision making was done at a lower level, the ICT management team, together with the Corporate information manager and the business information managers. At this stage it was decided which changes to prefer. Decision criteria were: are the resources available, can the supplier handle this, is there enough test capacity available. For the 30 biggest applications there was also a release calendar (these 30 applications were chosen arbitrarily). There was no prioritization between IS and non-IS at this maintenance level. Triggers for maintenance could be business, IT, or suppliers. Business information manager (9) argued that the maturity of the maintenance process was lower than the inception process (project portfolio). In the end the business was responsible for the maintenance decision. In case the maintenance was relatively expensive, more stakeholders were involved, such as architects.

Abandoning at Case E

Concerning the abandoning of IS, there was no separate abandoning process. However, when new projects were started, abandoning old systems was always a part of this. The overall maintenance and operation budget for the Case E was fixed, so in case new systems were added, preferably others were abandoned.

IS management at Case E

Concerning IS management, there was project portfolio management; the governance was done by the IS steering group in which the highest management team was involved in decision making (e.g. CFO and COO). There was a form of application portfolio management, as a result of the fact that there was a release calendar for the 30 most important applications. Applications were plotted on the main business processes. No formal application life cycle management practice was in place.

7.4 Analyzing the validating cases

In the former section, the validating case study organizations were described. In the following sections the accompanying research questions are addressed.

Conform the CSP, all interviews were recorded and transcribed. During the interviews (see Q 52, Table 50, p. 183) more stakeholders were identified and, if possible, interviewed. This resulted in 15 interviewees who were interviewed[12]. The interviews were transcribed and the transcriptions were sent back to the interviewees. The interviewees were able to add or change information, or withdraw the interview. The recorded and transcribed audio files were time stamped and saved in the case study database. This also applied to any additional documents provided by the interviewees. To avoid seeking conformance, the CSP described that the interviewer had to have a critical reflection and had to do further probing of issues. The questionnaire was semi-structured, Appendix B.2 (Table 51, p. 184) describes the relation between the subsequent legacy IS life cycle stages, the research questions, the sub-research questions, the design dilemmas and the "questions" in the questionnaire. The analyses were performed using qualitative research software ATLAS.ti. All the transcribed interviews and relevant documentation provided by interviewees were loaded into ATLAS.ti and were coded. In Table 25 the coding scheme is presented.

Codes	**Corresponding questionnaire questions.**
SRQ 14 What are internal legacy IS aging factors?	Q18, Q35, Q39
SRQ 15 What are external legacy IS aging factors?	
SRQ 16 What are triggers to abandon legacy IS?	
SRQ 17 What legacy IS abandonment decision making perspective (e.g. technical, functional, economical) is predominant?	Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q19, Q20, Q21, Q22, Q23, Q24, Q25, Q26, Q27, Q28, Q29, Q30, Q31, Q32, Q36, Q37, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q45, Q46, 47, Q48, Q49
SRQ 18 To what extent does a commercial off-the-shelf (COTS) or a proprietary (own build) IS influence the characteristics of abandonment decision making?	
SRQ 19 To what extent do IS supporting primary business processes versus IS supporting secondary processes influence the characteristics of abandonment decision making?	
SRQ 20 To what extent does the size of an IS (small or large) influence the characteristics of abandonment decision making?	
SRQ 21 What are good practices in legacy IS abandonment decision making?	
SRQ 22 What are good practices in practically abandoning legacy IS?	Q33, Q34, Q39
SRQ 23 Enhancing existing IS decision making methods or designing a new legacy IS abandonment decision making method?	Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q19, Q20, Q21, Q22, Q23, Q24, Q25, Q26, Q27, Q28, Q29, Q30, Q31, Q32, Q36, Q37, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q45, Q46, 47, Q48, Q49
SRQ 24 Decision making at information function, IS or application level?	Q39
SRQ 25 Financial versus non-financial evaluation aspects?	Q39
SRQ 26 Many or few attributes in the abandonment decision making method?	Q24
SRQ 27 Life cycle or portfolio oriented legacy IS assessment?	Q19, Q20
SRQ 28 Many stakeholders versus few stakeholders involved in legacy IS abandonment decision making?	Q27

Table 25. Codes based on answering the questions in the questionnaire and the sub-research questions

By means of queries on the coded material, the three subsequent legacy IS processes: (1) *the aging process of legacy IS*, (2) *the abandonment decision making process of legac*y IS and (3) *the practical abandoning process of*

[12] One interview included two interviewees, making a total of 14 interviews and 15 interviewees.

legacy IS and also (4) *design dilemmas for a method to abandon legacy IS* are validated and described in the following sections.

7.5 Validating the aging process of legacy information systems

From the literature and the first exploratory Case 140 unique legacy IS aging factors were derived. In upcoming validating case studies, the generalizability of the earlier findings will be explored. The results might confirm, or contradict and enhance earlier results of the explorative case. Aging factors were discerned and the accompanying sub-research questions were:

- SRQ 14: What are internal legacy IS aging factors?
- SRQ 15: What are external legacy IS aging factors?

The aim of this research was to identify as many unique legacy IS-related characteristics at an elementary level as possible. Because this research was based on case studies, relative frequency or relative importance did not play a role.

SRQ 14 What are internal legacy IS aging factors?
By means of first cycle coding in ATLAS.ti, all internal legacy IS aging factors were structurally coded. Subsequently, a query was performed on the code: "internal legacy IS aging", which resulted in all internal legacy IS aging factors with references to the interviewees and the organizations. Subsequently, in second cycle coding all similar IS internal legacy IS aging factors were axial coded resulting in 32 internal legacy IS aging factors. Next, the axial codes were compared with the results from the explorative research and the literature research and are presented in Appendix C (p. 185). When the axial coded internal legacy IS aging factors were identical to the explorative research or the literature research, they were labeled "C" (Confirmed). If the internal legacy IS aging factors had not been identified in the exploratory research or the literature research, they were labeled "N" (New). Overall, the four case studies confirmed 23 aging factors and identified nine new aging factors. The result is illustrated in Table 26. This table includes all internal legacy IS aging factors (rows), all interviewees (columns) and case organizations. An "X" means that this internal legacy IS aging factor was mentioned by a particular interviewee in a particular organization.

Internal legacy IS aging factor.	Case organization	B					C						D		E	
	Role	Consolidation project manager (3)	Application manager (1)	Rationalization project manager (2)	CIO/ICT director (2)	Architect (6)	Rationalization programme manager (1)	Rationalization project manager (1)	Business information manager (12)	IS owner (2) (director)	Business information manager (11)	Functional maintenance manager (4)	Functional maintenance manager (3)/Senior User (3)	Business information manager (10)	Business information manager (9)	(C)onfirmed or (N)ew
	Interviewee No.	1	2	3	4	5	6	7	8	9	10	11	12, 13	14	15	
Lots of manual interventions.															X	C
System skills hard to find, insufficient internal support anymore/knowledge of IS left organization.				X											X	C
Different versions of single software package, lower versions of the software, not using latest release.							X	X								C
Old hardware type, hardware is labeled old.										X					X	C
Software is no longer able to run on the old hardware. Legacy IS run on obsolete hardware.										X						C
COTS software which is changed according to the working process instead of using the standardized process.					X					X						N
Inadequate architectural decisions in the past, limited architectural fit.			X													C
A lot of unnecessary functionality has been added to the IS.						X										N
Unintended use of the IS (using the IS for which it is not designed and then adding functionality).						X										C
Inadequate documentation.		X	X	X	X			X	X			X				C
The integrity of the data may be poor and contain information it should not contain. Inadequate data integrity/corrupted data integrity.		X	X			X										C
Functionality does not meet requirements (wrong functionality).						X										C
Limited agility (changes take longer), long lead times required to implement changes to the software.					X	X										C
Inadequate stability of the IS.		X						X								C
Inadequate usability.								X								N
Inadequate maintainability.								X								N
IS includes workarounds.						X										C
Similar or redundant IS capabilities (e.g. due to a merger).		X		X	X	X				X	X		X		X	C
Products or services supported by the IS are abandoned, capabilities provided by the IS are less used, less internal support for the IS will be available.				X	X				X		X			X		C
Outdated infrastructure (including old platform).					X	X			X		X	X	X	X		C
Inadequate software quality, dead code in the software.			X			X			X			X			X	C
Growing functionality gap (additional functionality required).					X								X	X	X	C
Company policy changed to COTS, which means that custom-made software is no longer preferred.									X						X	N
Outsourcing is preferred and all applications are reassessed (first clean your application landscape before you outsource).										X						N
Standardization is preferred.										X	X					N

Redundant data centers (being not fail over) e.g. seven decentralized data centers instead of one central.		X					X			X	X				C
Redundant (double or similar) functionality.	X	X	X	X		X	X				X			X	C
Non-scalability of existing IS.	X	X		X	X										C
Enormous proliferation of (unknown) IS in unknown (different) places.	X			X		X			X		X			X	N
Data is no longer needed.									X						N
High cost per user.			X	X											C
High IS cost.		X	X	X	X		X	X	X		X	X		X	C

Table 26. Internal legacy IS aging factors

The most frequently mentioned internal legacy IS aging factors provided by the interviewees were: "high IS cost[13]" (10 times), "similar or redundant IS capabilities" (e.g. due to a merger) (eight times), "redundant (double or similar) functionality" (eight times), "outdated infrastructure (including old platform)" (seven times), "inadequate documentation" (seven times). It should be noticed that Case B had recently merged and Case C had recently consolidated. A merger or consolidation of similar organizations also results in a lot of redundancy, which was causing high IS cost. When the results per case study organization are compared (every individual "X" within an organization is compared across four organizations), the legacy IS aging factor: "high IS cost" and "similar or redundant IS capabilities" were mentioned in all four organizations. Three organizations identified: "product or service abandoned", "outdated infrastructure" (including old platform), "inadequate software quality", "growing functionality gap" (additional functionality required), "redundant (double or similar) functionality" and an "enormous proliferation of (unknown) IS in unknown (different) places".

Nine new internal legacy IS aging factors were identified. All the new IS legacy characteristics are added to the overall list of internal legacy IS aging factors in Appendix C.2 (p. 192; Table 58). Furthermore, Table 26 illustrates that different interviewees within the same organization sometimes identified different internal legacy IS aging factors for similar situations, emphasizing that no uniform viewpoint on internal aging factors of legacy IS exists and that different stakeholders have different internal aging viewpoints (e.g. technical, functional and economical).

SRQ 15 What are external legacy IS aging factors?

Similar to the internal legacy IS aging factors, the external legacy IS aging factors were also structurally coded throughout all interviews and documentation. Subsequently, a query was performed within ATLAS.ti on the code: "external legacy IS aging", resulting in all the external legacy IS aging factors with references to the interviewees and the organization. Then within second cycle coding all the similar external legacy IS aging factors provided by the interviewees were axial coded, resulting in four external legacy IS aging factors. The axial codes were compared with the results from the explorative research and the literature research as presented in Appendix C (p. 185). When the axial coded legacy IS aging factors were identical to explorative research, they were labeled "C" (Confirmed) and four were labeled confirmed. If the legacy IS aging factors had not been identified in the exploratory research, they were labeled "N" (New). No new external legacy IS aging factors were identified. The results are illustrated in Table 27. This table includes all external legacy IS aging factors (rows), all interviewees (columns) and case organizations. An "X" means that this external legacy IS aging factor was mentioned by a particular interviewee within a particular organization.

[13] High cost was not defined in absolute numbers, for a large organization cost might be small, whilst for a small organization the same cost might be high.

External legacy IS aging factor. \ Case organization	B					C						D		E	(C)onfirmed or (N)ew
Interviewee No.	1	2	3	4	5	6	7	8	9	10	11	12, 13	14	15	
New capability required which could not be implemented in the current IS (no longer meeting the needs of the environment, due to external demands from the market).				X	X							X			C
High IS cost.				X	X		X	X	X		X	X		X	C
New technological opportunities (e.g. superior new technology).					X				X	X	X	X	X	X	C
Limited or ended supplier support (e.g. service regarding hardware, software or databases).	X			X	X			X	X		X		X	X	C

Table 27. External legacy IS aging factors

The most frequently mentioned external legacy IS aging factors provided by the interviewees were: "high IS cost" (eight times), "limited or ended supplier support" (eight times), and "new technological opportunities (e.g. superior new technology)" (seven times). When comparing the external legacy IS aging factors at organization level (every individual "X" within an organization is compared across four organizations) then: "high IS cost", "new technological opportunities" and "limited or ended supplier support" were mentioned by interviewees in all four organizations. No new external legacy IS aging factors were identified. Table 27 illustrates that different interviewees within the same organization distinguished external legacy IS aging factors for similar situations, emphasizing that uniform viewpoints on external aging factors of legacy IS hardly existed and that different stakeholders had different external aging viewpoints (e.g. technical, functional and economical).

Concerning internal and external legacy IS aging factors, the complexity is illustrated through the identified 149 legacy IS aging factors (see Appendix C, p. 185). From these 149 aging factors, 14 aging factors were external and beyond the control of the organization, 20 were scored both internal and external and 115 were scored internal, meaning within the control of the organization.

7.6 Validating the abandonment decision making process of legacy information systems

In this section the legacy IS abandonment decision making process as identified in the literature and the explorative case organization are validated in the four validating cases. Validating the IS abandonment decision making process is based on the answers provided by the interviewees on the following sub-research questions (SRQ). The results might confirm, contradict or enhance earlier results of the explorative case. The accompanying sub-research questions were:

- SRQ 16: What are triggers to abandon legacy IS?
- SRQ 17: What IS abandonment decision making perspective (e.g. technical, functional, and economical) is predominant?
- SRQ 18: To what extent does a commercial off-the-shelf (COTS) or a proprietary (own build) IS influence the characteristics of abandonment decision making?
- SRQ 19: To what extent do IS supporting primary business processes versus IS supporting secondary processes influence the characteristics of abandonment decision making?
- SRQ 20: To what extent does the size of an IS (small or large) influence the characteristics of abandonment decision making?
- SRQ 21: What are good practices in legacy IS abandonment decision making?

SRQ 16: What are triggers to abandon legacy IS?

Interviewees provided the legacy IS abandonment triggers of their organizations (see Q35 questionnaire, Table 50, p. 183). By means of first cycle coding, all legacy IS abandonment triggers were structurally coded. Subsequently a query was performed within ATLAS.ti on the code: "abandonment trigger", resulting in all the legacy IS abandonment triggers with references to the interviewee and the organization. In second cycle coding

all the similar legacy IS abandonment triggers provided by the interviewees were axial coded and valued as internal or external abandonment triggers. These axial codes were compared with the result from the explorative research and the literature research as presented in Appendix C (p. 185). When the axial coded abandonment triggers were identical to the explorative research or the literature research, they were labeled "Confirmed" (C). If the abandonment triggers had not been identified in the exploratory research or the literature research, they were labeled "New" (N). The results are illustrated in Table 28. This table includes all abandonment triggers (rows), all interviewees (columns) and case organizations. An "X" refers to the fact that this abandonment trigger was mentioned by a particular interviewee in a particular organization.

Case organization / **Interviewee No.** / **Legacy IS abandonment triggers.**	B					C						D		E	(I)nternal/ (E)xternal	(C)onfirmed or (N)ew
	1	2	3	4	5	6	7	8	9	10	11	12, 13	14	15		
Merger or consolidation.	X		X	X	X				X	X		X		X	I	C
Abandoning products or services supported by the IS.			X	X				X		X			X		I	C
Aged/old hardware.				X	X			X		X	X	X	X		I	C
Inadequate software quality.								X			X				I	C
Application functionality gap/not sufficient.				X									X		I	C
Internal directive COTS. Business information manager (9): *"our goal is to have the least possible own build".*														X	I	N
Outsourcing, first clean your IS landscape. IS owner (2): *"outsourcing was a trigger to clean the IS landscape and to abandon many systems".*									X						I	N
Standardization. Business information manager (11): *"we standardized our processes and systems and abandoned systems not according to this standard".*									X	X					I	N
Redundant data centers.		X		X			X			X	X				I	C
Redundant application functionality.	X	X	X	X			X				X			X	I	C
Redundant IS.			X								X				I	C
Non-scalability of the IS.	X	X		X	X										I	C
Enormous proliferation of (unknown) information systems in unknown (different) places. Rationalization programme manager (1): "*we had 17 measurement laboratories on different premises in the country, all with their own IS, and we rationalized all these premises, by selecting one IS and abandoning the rest".*				X		X									I	N
Data is no longer needed. IS owner (2): *"we stored data, because we always stored it, however, we are not going to use it, ... management decided to abandon this data, therefore the IS was abandoned".*									X						I	N
Momentum, opportunity for a new organization. CIO/ICT director (2): *"when I became CIO, of this consolidation organization, I inherited legacy IS from the former organizations, which was almost useless and had low value, ... the equity firm was prepared to invest and I saw it as a perfect opportunity to get rid of all legacy IS in one time".*		X		X											I	N
High cost per user or customer.			X	X											I/E	C
High IS cost.		X	X	X	X			X	X		X	X		X	I/E	C
New technological opportunities.									X	X	X	X	X	X	E	C
New demands from the market which cannot be provided by the current IS.				X											E	C
Limited or ended supplier support.				X	X			X	X		X		X	X	E	C

Table 28. Triggers to decide to abandon a legacy IS

Table 28 illustrates that different interviewees within the same organization distinguished different triggers to decide to abandon legacy IS. This illustrates that stakeholders may vary in their perception of abandonment

triggers. All these different viewpoints, however, are valuable for an organization in deciding when to abandon a legacy IS. Furthermore, most triggers reoccur in other organizations. For example, if Case B was not included in this research only one legacy IS abandonment trigger was missed, which was: “momentum, opportunity for a new organization”. If Case C had not been included, the maximum of three triggers would have been missed. If Case D had not been included, this would have had no implications for the identified triggers. If case E had not been included, one trigger would have been missed.

An overview of the number of legacy IS abandonment triggers as found in the literature, Case A (see Appendix C.3, p. 193) and Case B, C, D and E (Appendix C.4, p. 197) is provided in Table 29, illustrating the occurrence of the legacy IS abandonment triggers in the literature, Case A and Case B, C, D and E. If abandonment triggers found in Case B, C, D and E were already found in the literature or Case A, they were labeled confirmed. If abandonment triggers earlier discerned in the literature or Case A were not found in Case B, C, D and E, they were labeled non-confirmed. If abandonment triggers found in Case B, C, D and E were not found in the literature or Case A, they were labeled new and added as an abandonment trigger.

Legacy IS abandonment triggers.	**Literature sources**	**Case A**	**Case B, C, D and E**	**Total**	**No. in Figure 28 (p. 123)**	
High IS costs.	4	1	4	9	2	Confirmed
Limited or ended supplier support.	4	1	4	9	7	
New technological opportunities.	4	1	3	8	6	
Inadequate software quality.	6		1	7	5	
Aged/old hardware.	3		3	6	4	
Merger or consolidation.		1	4	5	9	
Abandoning products or services supported by the IS.		1	3	4	29	
Redundant application functionality.		1	3	4	10	
Application functionality gap/not sufficient.		1	2	3	14	
Redundant data centers.		1	2	3	26	
Redundant IS.	1		2	3	8	
High cost per user or customer.		1	1	2	3	
Non-scalability of the IS.		1	1	2	20	
New demands from the market which cannot be provided by the IS.		1	1	2	31	
Capability shortcomings.	5			5	15	Non-Confirmed
Limited or no technical skills available.	2			2	11	
Infrastructure overhaul.	1			1	12	
Application functionality no longer needed.	1			1	13	
Legal compliance (law).		1		1	17	
Does not fit in technological strategy.		1		1	18	
Instability of the IS.		1		1	19	
Technical aging.		1		1	21	
The IS is non-agile.		1		1	22	
Structural lack of IS maintenance.		1		1	23	
Vendor dependency/vendor lock-in.		1		1	24	
Effort to migrate applications from an old data center to a new data center.		1		1	25	
Redundant data sources.		1		1	27	
Perception of the required IS capability is changed.		1		1	28	
Capability not needed.		1		1	30	
Cost-benefit.		1		1	1	
Security issues.		1		1	16	
Complexity (+/-).	1	1		1	32	
Enormous proliferation of (unknown) information systems in unknown (different) places.			2	2	33	New
Outsourcing, first clean your IS landscape.			1	1	34	
Standardization.			1	1	35	
Internal directive COTS.			1	1	36	
Data is no longer needed.			1	1	37	
Momentum, opportunity for a new organization.			1	1	38	

Table 29. Legacy IS abandonment triggers, sorted on total appearance in the literature and the organizations

Based on Table 29 and the conceptual model (see Figure 24, p. 90), the model is further enhanced with six extra legacy IS abandonment triggers. In total 38 legacy IS abandonment triggers were identified, see Figure 28.

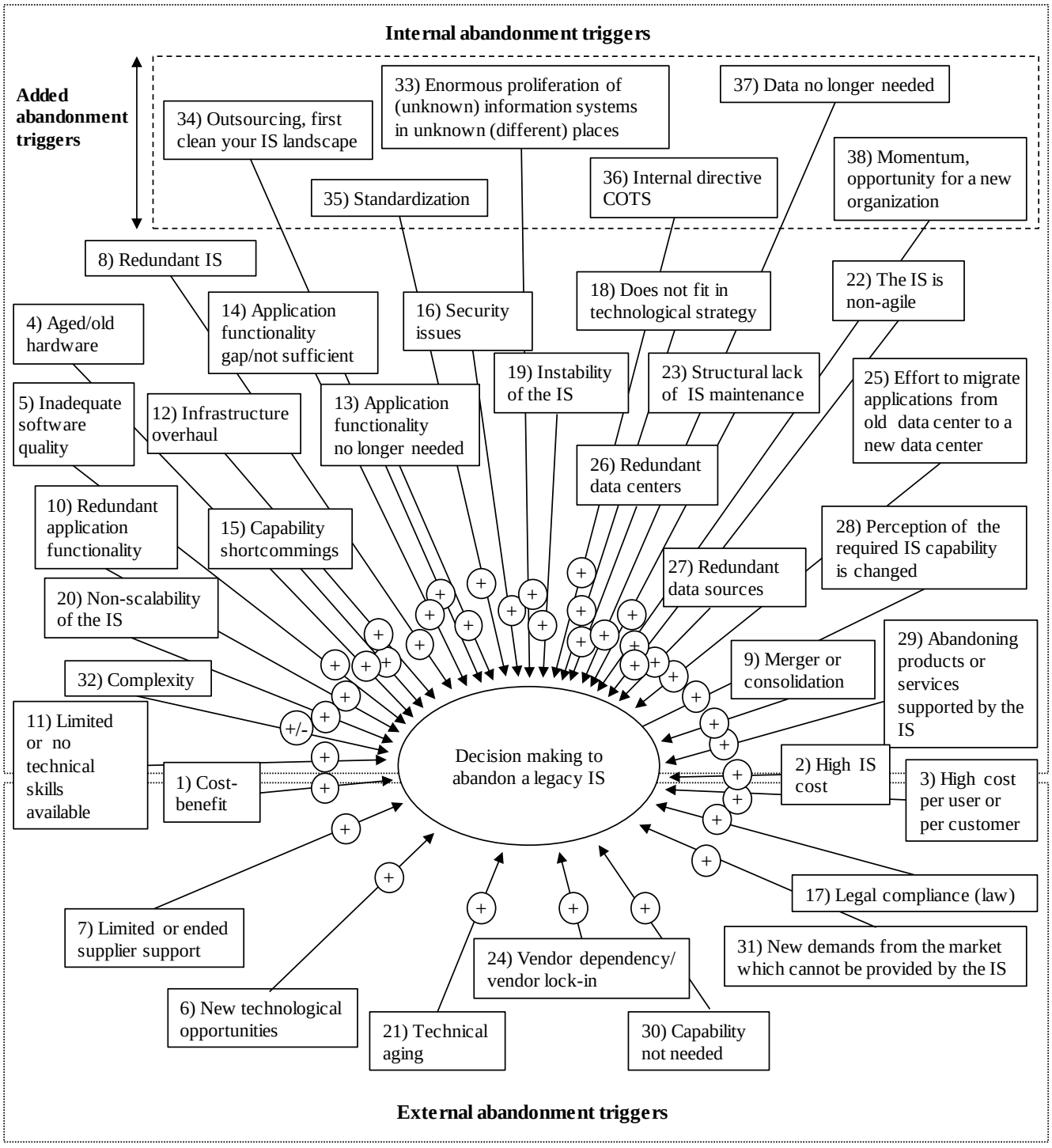


Figure 28. Identified legacy IS abandonment triggers

In the validating cases most of the abandonment triggers as found in the literature and the exploratory case return and only six new triggers could be identified. Overall, this research identified 38 abandonment triggers. This illustrates the complexity of abandonment decisions. Because most applications databases will hardly include 38 items, proactive management of IS aging will be difficult. SRQ 17 investigates whether any of the abandonment decision making perspectives is predominant in abandonment decisions.

Due to the fact that again there is an overlap between legacy IS aging factors and legacy IS abandonment triggers (13 factors were mentioned by several interviewees, as both a legacy IS aging factor and a legacy IS abandonment trigger), it is concluded that an increasing number of legacy IS aging factors and legacy IS abandonment triggers accumulate and made the legacy IS "older" (see Figure 25, p. 91). The legacy IS

abandonment trigger passes a legacy IS aging threshold which makes organizations decide to abandon a legacy IS. There can be a long time between the decision to abandon a legacy IS and the practical abandoning (e.g. years in Case B). During this period only "must do" maintenance should be performed and during this period the aging of the legacy IS continues.

SRQ 17: What legacy IS abandonment decision making perspective (e.g. technical, functional, and economical) is predominant?

Interviewees were asked to score (see Q38, Table 50, p. 183) the most important discipline in decision making concerning the abandoning of legacy IS (lowest is most important); importance is illustrated in Table 30.

Organization	Case B	Case C	Case D	Case E	Average score
Political	3	4			3.5
Mathematical					
Statistical					
Economical	2	2.25	1	2	1.8
Sociological		3			3
Technological		1.75	2.5	2	2.1
Anthropological					
Functional	1	1	2.5	2	1.6
Security					
Philosophical					
Psychological					
Internal politics				4	4

Table 30. Decision making discipline

According to interviewees, the following decision making disciplines were predominant, in order of predominance: "functional, economical, technological, sociological, political and internal politics". Different interdisciplinary approaches to decision making may be viewed as decision making models because they represent a particular segment of the real world Harrison (1999). Decision making models are not mutually exclusive, different models will share some of the same components (Harrison, 1999). Therefore the identified abandonment triggers (see Figure 28, p. 123) were categorized into the predominant abandonment decision making perspectives as mentioned by the interviewees. This is illustrated in Table 31.

Legacy IS abandonment triggers.	No. in Figure 28 (p. 123).	(F)unctional, (E)conomical, (T)echnical.
Merger or consolidation.	9	F
Redundant application functionality.	10	F
Application functionality no longer needed.	13	F
Application functionality gap/not sufficient.	14	F
Capability shortcomings.	15	F
Legal compliance (law).	17	F
Perception of the required IS capability is changed.	28	F
Abandoning products or services supported by the IS.	29	F
Capability not needed.	30	F
New demands from the market, which cannot be provided by the IS.	31	F
Data is no longer needed.	37	F
Momentum, opportunity for a new organization.	38	F
Cost-benefit.	1	E
High IS cost.	2	E
High cost per user or customer.	3	E
Aged/old hardware.	4	T
Inadequate software quality.	5	T
New technological opportunities.	6	T
Limited or ended supplier support.	7	T
Redundant IS.	8	T
Limited or no technical skills available.	11	T
Infrastructure overhaul.	12	T
Security issues.	16	T
Does not fit in technological strategy.	18	T
Instability of the IS.	19	T
Non-scalability of the IS.	20	T
Technical aging.	21	T
The IS is non-agile.	22	T
Structural lack of IS maintenance.	23	T
Vendor dependency/vendor lock-in.	24	T
Effort to migrate applications from an old data center to a new data center.	25	T
Redundant data centers.	26	T
Redundant data sources.	27	T
Complexity (+/-).	32	T
Enormous proliferation of (unknown) information systems in unknown (different) places.	33	T
Outsourcing, first clean your IS landscape.	34	T
Standardization.	35	T
Internal directive COTS.	36	T

Table 31. The abandonment triggers categorized into considering functional, economical, technical perspectives

Table 31 illustrates the identified legacy IS abandonment triggers that were categorized into eclectic approaches of decision making. Three decision making disciplines prevail, functional, economical and technical. In this categorizing it should be emphasized that occasionally abandonment triggers address multiple perspectives. For example, vendor dependency/vendor lock-in, which was scored as a technological abandonment trigger, might also result in high IS cost, which was scored as an economical abandonment trigger. Another example was that a structural lack of IS maintenance, which was also scored as a technical legacy IS abandonment trigger, will probably also result into capability shortcomings which was scored as a functional legacy IS abandonment trigger.

Concerning the other decision making eclectic perspectives on legacy IS decision making mentioned by interviewees, namely: political, internal politics and sociological, one abandonment trigger "Legal compliance (law)" was categorized functional, however, might also be categorized as political. This study did not identify mathematical, psychological, sociological or philosophical perspectives on legacy IS abandonment decision making perspectives.

SRQ 18: To what extent does a commercial off-the-shelf (COTS) or a proprietary (own build) IS influence the characteristics of abandonment decision making?
The assumption was that proprietary legacy IS would be more difficult to abandon than COTS IS for all sorts of reasons such as: more personal emotions or personal stakes would be involved and more specific knowledge would be required. Interviewees were asked (see Q37, Table 50, p. 183) if the decision making to abandon a proprietary (own build) IS was easier than one which was bought COTS. In all cases it was confirmed that personal emotions and stakes were higher with proprietary IS. The difficulty of the decision to abandon a custom-made IS was quoted by Business information manager (11) as follows: *"proprietary IS are considered to be babies and you are not abandoning your baby".* Consolidation project manager (3): *"the risks of abandoning proprietary IS are also higher due to the fact that you often need specialist staff, paradoxically this is also a reason why you want to abandon the IS"*.

SRQ 19: To what extent do IS supporting primary business processes versus IS supporting secondary processes influence the characteristics of abandonment decision making?
The assumption was that continuity issues of primary IS would include more risk compared to secondary IS and therefore, could influence decision making characteristics. Interviewees were asked if the decision making to abandon a primary IS is easier than a secondary IS (see Q36, Table 50, p. 183). In all cases it was confirmed that there is a difference in abandoning primary IS compared to supporting IS. According to Business information manager (12): *"I consider a primary IS as leading and a secondary IS as following; it is easier to abandon following IS compared to leading IS".* According to IS owner (2): *"if something goes wrong with abandoning a primary IS, it will influence the primary task of the organization*".

SRQ 20: To what extent does the size of an IS (small or large) influence the characteristics of abandonment decision making?
The assumption for this question was that there is a difference between abandoning small and large IS. Interviewees were asked (see Q23, Table 50, p. 183) if there are differences between the life expectancy of relatively small IS and large IS? According to the interviewees, larger systems evidently receive more attention simply because they require more funding or have a bigger impact on the organization. The life expectancy of larger systems was also longer compared to smaller systems, because it is more difficult to abandon larger IS, or to replace it by a new IS. Occasionally abandoning will also lead to disinvestments (operational IS do not have any trade in value and the capital destruction of a large IS is much higher compared to a small IS).

In two case studies it was observed that larger IS "eat" smaller IS in case of mergers. Larger IS were capable of handling volumes that smaller competing IS could not and also had lower switching costs. Larger IS also had more and more powerful stakeholders. Larger IS might support more products and services. If a product or service was abandoned, there were still more products left justifying the support of an IS. The organizational risks of abandoning large IS were experienced as higher compared to a smaller IS. Business information manager (9) said: *"I expect that larger IS are there for a longer period, say 10-15 years; the investment of a million in SAP is not intended for only three years".* Business information manager (11) said: "*In our organization SAP has an almost unlimited life expectancy, compared to 3-5 years for smaller IS".* According to IS owner (2)*: "there is always a lot of money invested in larger IS, in case of abandoning, a disinvestment might be higher, therefore they are more difficult to abandon. The alternative is often also very expensive*". Therefore, larger IS do have longer life expectancies compared to smaller IS and decision making is often different in terms of management attention and number of stakeholders, organizational risks, capital destruction and options for new succeeding IS.

SRQ 21: What are good practices in legacy IS abandonment decision making?
In Section 6.3 (p. 90) good practices were conceptualized based on the explorative case organization. These good practices have been validated in the four case organizations. Interviewees provided good practices (success criteria) related to the IS abandonment decision making of legacy IS (see Q32, Table 50, p. 183). These good practices were identified by *"accumulating and applying knowledge about what is working and not working in different situations and contexts, including lessons learned and the continuing process of learning, feedback, reflection and analysis"* (Alberti & Bertucci, 2006). Due to limited number of cases in this research, these practices will be labeled "good practices" and not "best practices" (Alberti & Bertucci, 2006). The organizations report these good practices based on multiple practices.

By means of first cycle coding, all good practices were structurally coded. Subsequently, they were axial coded. The axial codes were compared with the conceptualized good practices presented in Section 6.3 (p. 90). The good practices were categorized in "C" (Confirm) when the additional conceptualized good practices were similar to earlier results. When there was no confirmation, it was categorized "E" (Explorative), referring to the explorative case or "N" (New) referring to the fact that it was found in one of the validating case organizations. Overall, 37 good practices were identified of which eight were confirmed, eight were non-confirmed and 21 were new good practices. The results are presented in Table 32.

No.	Good practices in legacy IS abandonment decision making.	C/ E/ N
1	The decision to abandon a legacy IS should always be driven by business economics (even though the organization was a public organization).	C
2	To support the legacy IS abandonment decision making in-depth analyses of this decision was necessary.	C
3	Plotting applications on the business processes helped in visualizing overlap of functionality; this was used for decision making.	C
4	A success factor for abandoning legacy IS decision making was to actually know the existence of all applications (have the application base organized, e.g. know their names, functionalities and who were using them).	C
5	Commitment by board management was a success factor.	C
6	Success factors in decision making were that goals were made clear to everybody and management acted in accordance with these goals.	C
7	In case of mergers, it was an advantage to have decision makers who did not personally purchase or built the IS.	C
8	IS owners should paid for system maintenance.	C
9	Pressure on the expense budgets helped to make decisions to abandon legacy IS (every abandoned IS would no longer consume resources, e.g. money and staff).	E
10	In the decision making to abandon legacy IS, the programme that realized the abandoning of the legacy IS operated as a team, the client of Case A was the focus, not the internal powers in Case A.	E
11	It was suggested to have regular meetings (e.g. once or twice a year), as part of information planning and to evaluate all IS and to decide which IS could possibly be abandoned.	E
12	Three stakeholders should always be part of the decision making process: business, ICT and finance.	E
13	Prohibiting adding new IS, otherwise growth might still outnumber abandoning.	E
14	Scoring models including criteria and weighing factors supported legacy IS abandonment decision making.	E
15	Sometimes the system was redundant, but its data needed to remain accessible (for instance compliance). A solution offered to store this data (e.g. on CD) helped the abandonment decision making.	E
16	Adequate information concerning the legacy IS (interfaces, impact, costs and contracts) was very important.	E
17	A success factor was to tune the abandonment decision in advance with stakeholders.	N
18	A good and thorough plan (including dependencies and time plan) was important.	N
19	Take the time for good asset management (for instance, where is the IS located practically, identify which contracts with suppliers were associated with the IS).	N
20	In case it was a technical matter, abandoning a legacy IS and replacing it by better and cheaper hardware resulted in the fact that no customers complained. When it was a functional matter, however, like an application rationalization programme, then it was a pitfall - customers complained. It was, however, a success if functionality was combined and applications were abandoned if customers were provided extra functionality.	N
21	A success factor was that there was only one owner. So if there was an application which covered more domains (business units) it was our policy to appoint one accountable owner, the rest was user.	N
22	Start with a list of application names, including functionalities, of who were using them and of how much the damage was in case of abandoning.	N
23	Sometimes data had to be kept for a long time; roads are there for more than 100 years and data might be stored for 100 years or longer. A solution had to be found for storing this data.	N
24	The expectations of the business concerning the benefits of decision making to abandon a legacy IS were often too high. Business cases were used to communicate and present a reduction of 10 FTE, whilst the management was expecting a reduction of 50 FTE.	N
25	Dependency on the situation of the organization. If the organization was in a consolidation situation it was different compared to a normal situation in which there was a stable organization. From the perspective of a consolidation situation a success factor was that there were a few smart people who received a lot of freedom to prepare and make decisions. Because the success of the inception of the new organization was a ticket for a new position this was also a success factor; it forced individuals to detach themselves from their former organization and to think of the new organization.	N
26	Three stage decisions comprised of: first the easiest abandonment decisions immediately, the more difficult with decision papers and applications which were decided to be later.	N
27	Bringing in an external partner was a good decision because they fitted in with the culture of the organization and their pragmatic approach.	N
28	The timing of the new CIO for doing this ARP was the right timing. We submitted a proposal to the business streams; if the business streams did not agree, we drew up a decision paper to the higher management. There was a dynamic pragmatic approach and it was allowed to make mistakes.	N
29	By abandoning old legacy IS, staff was forced to work with the new IS. That was a success factor in building a new organization.	N

30	There was enough money to consolidate the new and abandon the legacy IS. A success factor was ownership of the business to abandon the IS. When the business had to decide to practically abandon the IS, there was a big silence. It looks like an IT activity and it was a good decision to incept ARP and to let ARP help the business in practically abandoning applications. According to Application manager (1): "*a success factor could be that the business has a financial incentive; the business did not have much incentive to abandon, because they were not paying for the IS".*	N
31	A change process (to reverse the decision to abandon a legacy IS) was available for stakeholders who had a problem with the abandoning of an IS. It was a possibility to redo the assessment.	N
32	Part of the decision making was that the IS costs were transparent. That made decision making a lot easier, e.g. these 20 customers cost € 50k. Costs outweighed the benefits of using the legacy IS.	N
33	Before the decision to abandon legacy IS it was explained by the CIO to the board what would happen if the organization decided not to abandon these systems; it was an insurance when things went wrong, because legacy IS were not abandoned. When things went wrong, the board members didn't blame the CIO because he had advised to abandon the legacy IS and already had predicted the problems.	N
34	Suppliers (consultants) were abandoned that did not behave as partners in the ARP. According to CIO/ICT director (2): *"you do not want to have suppliers who will immediately show a contract when things get difficult, you want to have suppliers who first solve problems and then look at the contracts".*	N
35	There were problems with contracts which were not abandoned before the end of the year. As a result on 2 January a new invoice was sent for the following year.	N
36	Part of the abandonment decisions and guidance for the decisions was an architecture framework with principles, guidelines and standards.	N
37	According to IS owner (2): *"in the past we never abandoned due to the fact that the activity was never initiated; now that the activity has been initiated, it also received ample awareness, which was also a success factor in itself".*	N

Table 32. Legacy IS abandonment decision making good practices

7.7 Validating the practical abandoning process of legacy information systems

In this section the good practices of the practical abandoning process of legacy IS are validated. In the exploratory case study was identified that organizations should have an overview of all legacy IS by means of a database. Interfaces to other IS should be known. There should be a fall back scenario. The business (IS owner, e.g. director, business unit director) should always be accountable for the practical abandoning of a legacy IS. In case many legacy IS were abandoned, there should be a programme with standardized abandoning processes. In the following section the legacy IS practical abandoning good practices are validated based on the answers provided by the interviewees on the following sub-research question (SRQ):

SRQ 22: What are good practices in practically abandoning legacy IS?

In Section 6.4 (p. 92) good practices were conceptualized based on the explorative case organization. Subsequently these good practices were validated in the four validating case organizations. Interviewees provided their preferred good practices (success criteria) related to the practical abandoning of legacy IS (see Q33, Table 50, p. 183). These good practices were found by: "accumulating and applying knowledge about what is working and not working in different situations and contexts, including lessons learned and the continuing process of learning, feedback, reflection and analysis" (Alberti & Bertucci, 2006). The interviewees identified these good practices by repeated experiments. Practices are labeled "good" practices, because this research is limited to overall five case studies. Further research could validate whether these good practices could also be labeled "best" practices.

By means of first cycle coding, all good practices were structurally coded. Subsequently, they were axial coded. The axial codes were compared with the conceptualized good practices as presented in Section 6.4 (p. 92). The good practices were categorized "C" (Confirmed) when the conceptualized good practices were similar to the results from the validation case. When there was no confirmation the practice was categorized "E" (Explorative) referring to the explorative case. A practice was labeled "N" (New), when it was found in the case organizations testing phase. In total 60 good practices were identified, of which 12 were confirmed, 28 were identified in the explorative case organizations and 20 were identified in the testing phase. The results are presented in Table 33.

No.	Good practices for the practical abandoning of legacy IS.	C/E/N
1	Have a fall back scenario.	C
2	Have a strategy to back up your data, specific database (named: "*Historical Data Storage"* in the case organization), data on a cd, or throw away (dispose) media.	C
3	Identify the interfaces to other IS. It was a success factor if organizations knew how a system interfaced with other systems and how this would affect the other systems.	C
4	Start with an IS database (a baseline) that includes indicators such as: application names, costs, number of users, contracts and suppliers. The completeness of such a database was called a success factor.	C
5	In case people had to abandon their own legacy IS and also their own job (they were dismissed after the abandoning of the legacy IS) two things were important: an explanation about why the IS was abandoned and an adequate exit compensation.	C
6	In case an IS was abandoned due to the fact that there was a merger and another IS was preferred, avoid detailed discussion with stakeholders about the arguments.	C
7	In case an IS is succeeded by another system, make sure that there is adequate training on the new system.	C
8	We worked with the beep mechanism: we turned an IS off and when somebody was complaining we turned it on. (Make a difference between critical and non-critical IS to the operation. In case it is non-critical IS, you can just put the system down and see what is happening).	C
9	It was a success factor to make the business (IS owner, e.g. director, business unit director) accountable for the practical abandoning of an IS (ownership of the business) and to offer help from a decentralized organizational unit.	C
10	In case several IS were abandoned, creating a programme and making a standardized process was a success factor. Also put this programme outside of regular business operations. Otherwise problems in primary processes might prevail.	C
11	It cost a lot of time and resources to untangle all the IS which were to be abandoned (spaghetti).	C
12	Sometimes applications were found that were not listed in the company's application database.	C
13	When IS are practically abandoned, all data should be removed from these IS.	E
14	Inform stakeholders about the plans to abandon a legacy IS.	E
15	Give stakeholders from an abandoned IS a new job (in the new IS).	E
16	Test alternative IS extensively before abandoning the legacy IS.	E
17	The pressure to migrate to a new data center was useful in abandoning the legacy IS.	E
18	Avoid lumpsum contracts with suppliers. For instance, 10 applications including maintenance and operation were on one server. When a legacy IS was abandoned, it was not clear which part of the costs were recovered.	E
19	Avoid dispersion of the IS contracts and make them transparent. For instance the IS was split amongst (different) suppliers into test environment contracts, data center contracts, software contracts, database administrator contracts, maintenance contracts and were sometimes part of a lumpsum.	E
20	When IS were outsourced, professional contract management had to be supported by an adequate database.	E
21	IT staff should talk business language.	E
22	In case it was a decision of the board of directors, a clear decision of the board of directors to abandon or not to abandon a legacy IS has to be made.	E
23	Between the decisions to abandon a legacy IS and the practical abandoning of the legacy IS there was some leap time. In that case the maintenance was reduced to a bare minimum; only "must have" functionality was maintained. This implied that maintenance budgets were reduced. The bare minimum maintenance which was allowed was documented very well, to avoid additional abandoning work when the legacy IS was finally abandoned.	E
24	Technological complexity or the availability of support was a critical factor. In one case the legacy IS included reporting functionality and it was a lot of work to transfer this functionality to another system, also because the person responsible for this functionality left the organization almost immediately after hearing that the IS was going to be abandoned.	E
25	If an IS service provider that provided data center services and application maintenance services was abandoned, success was higher if a representative from the application department rather than a representative from the hardware department was approached. In outsourcing situations it was better to approach the applications people than the data center people. It seems that there was a hierarchy and that the application staff is higher in hierarchy than the hardware staff.	E
26	Communicate the progress of your abandon programme throughout the organization; in Case A the "abandonment thermometer" visualized the successes of the A&M programme.	E
27	If data stored at suppliers was abandoned, the suppliers should also practically abandon this data by contract.	E

28	Abandoning an IS was very painful for some stakeholders because these stakeholders identified themselves with a system and the importance of these systems. Pay attention to this from a management perspective.	E
29	Bring an IS to the grave. People forget that a system was often built with a lot of affection. Abandoning this system was therefore not always easy, make sure to have a party afterwards.	E
30	Consistently and consequently follow the goal to abandon the legacy IS.	E
31	Adequate project management skills in the abandoning of a legacy IS.	E
32	A sponsor on the highest board of the organization was absolutely necessary.	E
33	Between the decision to abandon and the actual practical abandoning there could be a long time lapse (e.g. in case of a merger, years). To ensure that the abandoning of a legacy IS was still accepted by the IS owner (e.g. director, business unit director), a few weeks before the actual practical abandoning of the IS, the owner of the IS was asked to sign a document stating that he agreed to abandon the IS.	E
34	Abandon the contracts with suppliers and cash the benefits by not paying anymore for the IS.	E
35	It was suggested to start with the simple low risk IS and to cluster similar IS.	E
36	Apply continuous improvement, "plan-do-check-act" cycles to evaluate and learn abandoning activities.	E
37	Adequate project or programme management to practically abandon legacy IS.	E
38	Pressure on the expense budgets of the business helped IS owners (e.g. director, business unit director) to decide to practically abandon legacy IS (every abandoned IS would not consume resources, e.g. money and staff).	E
39	After the decision was taken to abandon the legacy IS, it took some time before the legacy IS was practically abandoned. At the practically abandoning, there were emotions involved of people attached to the legacy IS. From a managerial perspective, pay attention to these emotions.	E
40	Select the right project members for the abandoning project.	E
41	It was difficult to find enthusiastic staff for this kind of work (practically abandoning legacy IS); staff had to dig and search for information and within the organization hardly anybody was really interested any longer.	N
42	The supply organization gave insight into the abandoning, the demand organization anticipated on the fact that IS were abandoned and created a basis (buy-in) within the user organization. It was very important that the demand organization was paying for the maintenance of systems; they then had an incentive to abandon IS and not to pay for them anymore. They were able to prioritize within their TCO.	N
43	We preferred that the Business information manager communicated the practical abandoning of legacy IS to the end users and not the centralized ICT organization or the programme or project which was responsible for the practical abandoning of the legacy IS.	N
44	It was important that the business unit directors kept questioning when the legacy IS was practically abandoned and that there was a possibility to return to the old system.	N
45	In case there was a need to keep the data by law, it was sometimes easier and cheaper to have the IS running somewhere on a very low performance server.	N
46	ICT personnel had their own priorities; abandoning legacy IS was low on their priority list.	N
47	Make sure that the data model is clear. According to Project manager consolidation (3): *"other organizations have to realize that it is the data which is most important, the data model should be clear… what is the value of the data, the data needs to be clean and clear if it is going to migrate to a new IS".*	N
48	The knowledge and commitment of the personnel was important; support within the organization was very important. Don't focus too much on the applications, also focus on the data and the data model.	N
49	In case of practical abandoning stick to the goal. According to Project manager consolidation (3): "*an application was a safety belt of staff within the organization; when this application was abandoned, the safety belt was abandoned. Stick to the goal, but pay attention to staff affections to an IS. Communication was very important".*	N
50	In case external consultants with scientific approaches were hired, the distance to the personnel was getting too big.	N
51	Concerning the migration of data, we did small batches; the mutation degree was quite high, so after some time it was impossible to go back, to the old system. We had pilot migrations, pilot batches; after these were more successful our batches became larger.	N
52	Do not act on rumors.	N
53	Changes for people were hard, they will bring their principal in doubts.	N
54	Follow a good practical abandoning process. According to Rationalization project manager (2): *"we did not have real problems due to the fact that we followed a good process. We did not unplug systems on the flight, we had a plan and by following this, we were trusted in the organization. The process was: first close the access to the application, (however, let the application run in the background) then there was a cooling off period and after 4 weeks we dismantled the hardware. If we missed something, like an interface, or some specific data that should be secured, we were able to start up within one hour".*	N

55	We also had plan how to handle data, e.g. for tax, or law purposes. In case we kept a certain financial application running on low performance for tax purposes, we made new contracts for licenses with suppliers. We made a change process in which stakeholders who had a problem with the abandoning of an IS had a chance to do an assessment again.	N
56	There was data in systems, sometimes the application was "abused" (work around); data was put in systems which should not be there and which was not known. Sometimes also other users were using the IS; the owner agreed with abandoning and then another user from another department appeared. So you have to do thorough analyses in advance.	N
57	An archive matrix, which describes how long data should be kept, for which purposes and if it is online or offline.	N
58	Because this organization was a consolidation to a new organization, it was not decided in advance who should get a management position. This position could be earned by successfully incepting the new Case B and "abandoning" the former organization, although practically the systems were not actively abandoned.	N
59	It was a success factor, that this was a greenfield, something new and not a merger.	N
60	In case some rest functionality should be kept, abandon some parts of the legacy IS. According to Functional maintenance manager (3): *"there was moment, a situation in which the legacy IS was trimmed from redundant functionality from the past. So if there were 40 tables that were not used anymore, these tables were discarded. Only three tables which were necessary for the moment were kept. This lowered the maintenance effort …. so we did not abandon the entire IS at that moment, we were just making it smaller. That also helped in the cost savings, as we had to pay our supplier for things like objects".*	N

Table 33. Good practices associated with practical abandoning

7.8 Validating the design dilemmas for a method to abandon legacy information systems

In this section, the six design dilemmas (which are derived from the explorative case in Section 6.5, p. 93) for a method to abandon legacy IS are validated. These design dilemmas will provide guidelines for designing a method to abandon legacy IS (Chapter 8). Each dilemma is validated based on the answers provided by the interviewees on the following sub-research questions (SRQ):

- SRQ 23: Enhancing existing IS decision making methods or designing a new legacy IS abandonment decision making method?
- SRQ 24: Decision making at information function, IS or application level?
- SRQ 25: Financial versus non-financial evaluation aspects?
- SRQ 26: Many or few attributes in the abandonment decision making method?
- SRQ 27: Life cycle or portfolio oriented legacy IS assessment?
- SRQ 28: Many stakeholders versus few stakeholders involved in legacy IS abandonment decision making?

SRQ 23: Enhancing existing IS decision making methods or designing a new legacy IS abandonment decision making method?

The question that needed to be answered was: "*what makes a decision to abandon a legacy IS different compared to a decision to invest into the inception of a new IS or a decision to invest in the maintenance of an operational IS?"* Table 23 (p. 95) describes the differences in the life cycle decision making concerning: (1) management level of the decision maker involved; (2) types of organizational risks involved; (3) financial implications of the decision; (4) effect on operations; (5) technological implications (6) operations or project; (7); documentation quality; (8) individual prestige (image or status effects). The decision making characteristics will be subsequently discussed.

Ad (1). Management level of the decision maker involved.

In all cases higher management decided upon inception. Furthermore, the interviewees indicated that relatively most higher management time was dedicated to inception decisions and lower management spent less time on inception. However, for maintenance this was the other way around: higher management was less involved in maintenance and decision making was delegated to lower management. All stakeholder spent least time on abandonment decisions (in the four cases of this study).

In the inception phase members of the board (e.g. CEO, CFO or CIO) were responsible for the inception decision. In the maintenance decision, the responsibility was delegated to lower management. Concerning the abandoning of legacy IS, the IS owner was always responsible. According to all interviewees within all three phases (inception, maintenance and abandonment) of decision making, the decision making could be characterized by negotiation resulting in a compromise and consensus in which a state of mutual agreement was reached. According to all interviewees, in inception and maintenance decision making no important stakeholders were missing. However, in the legacy IS abandonment decision making process, in three organizations four interviewees noted that some stakeholders were missing, but that this had little impact on the decision.

According to the interviewees higher management spent most of their time on inception decision making (74%) compared to maintenance (14%) and abandoning (12%) decision making. Lower management spent most of their time on maintenance decision making (66%) compared to inception (20%) and abandonment (14%) decision making as illustrated in Table 34.

Management level	Life cycle phase	Case B	Case C	Case D	Case E	Average organizations	Standard deviation
Higher management	Inception	56.7	67.5	80	90	74	12.6
	Maintenance	30	7	15	5	14	9.8
	Abandonment	13.3	25.5	5	5	12	8.4
Lower management	Inception	16.7	25	10	30	20	7.7
	Maintenance	65	60	77.5	60	66	7.2
	Abandonment	18.3	15	12.5	10	14	3.1

Table 34. Time involvement of management over the life cycle (total is 100%)

It can be concluded that in the case organizations, there was more time involvement of higher management in inception decision making, than in maintenance and abandonment decision making. Furthermore, there was more time involvement of lower management in maintenance decision making, than in inception and abandonment decision making. Abandonment decision making had the lowest time involvement of both higher and lower management. Inception decisions were taken by higher management, maintenance decisions were delegated to lower management levels. Abandonment decisions were taken by higher management in case of strategically important programs, or large IS. Abandonment decisions of smaller IS were always delegated to lower management levels.

According to the interviewees, the time distribution of working staff along the life cycle was on average 47% in inception, 47% in maintenance and 6% in the abandonment phase. This is illustrated in the Table 35.

Life cycle phase	Case B	Case C	Case D	Case E	Average organizations	Standard deviation
Inception	60	25	49	Includes abandonment 60	47	13.2
Maintenance	35	65	49	40	47	11.4
Abandonment	5	10	2	Includes abandonment 60	6	2.9

Table 35. Overall relative staff effort in FTE over the IS life cycle (inception, maintenance and abandoning, total 100%)

Ad (2). Organizational risks involved.

Interviewees in the organizations were asked to score anticipated organizational risks regarding the different life cycle phases. It is concluded that highest risks were perceived in the inception phase. The maintenance phase was scored medium risk. Abandoning scored medium to low organizational risk. This research confirms the assumption that inception decisions were considered to be riskier than maintenance decisions and abandonment decisions. These opinions are illustrated in Table 36.

Life cycle phase	Case B	Case C	Case D	Case E
Inception	H	H	H	M
Maintenance	M	M	M	M
Abandonment	M/L	M	M/L	M

Table 36. High (H), Medium (M) and Low (L) risks levels in the different life cycle stages

Ad (3). Financial implications of the decision.

In all organizations IS inception investment decision making was prioritized with other non IS investments. IS maintenance decision making being prioritized in competition with non IS investment only occurred in Case B. In all cases inception decisions making were perceived as investments, whereas maintenance decisions were perceived as operational costs. Abandonment decision was perceived as a disinvestment (most of the time there was no rest value in the abandoned IS asset).

Concerning financial implications, it was argued that IS inception investments are forward looking, there was limited financial project history and IS investments are considered to be relatively large. It was difficult to estimate the benefits. Maintenance investments were based on some financial history; most of the maintenance investment decisions concerned smaller investments, and it was easier to make an estimation of the benefits. For abandoning it was argued that there was a lot of financial history; the benefits of abandoning an IS could therefore be estimated relatively easy. According to the interviewees in the organizations answering the question: *"is the financial impact known of an inception, maintenance, abandonment decision of an IS?",* the financial impact was best known in the case of inception decision making, followed by the maintenance decision making. The financial impact of abandonment decision making was least well known. This is illustrated in Table 37[14].

Financial impact in life cycle phase	Case B	Case C	Case D
Is the financial impact known of inception decisions?	5	4.25	5
Is the financial impact known of maintenance decisions?	3	3	4
Is the financial impact known of abandonment decisions?	2	4	2

Table 37. The knowledge of the financial impact of decisions (ranging completely unknown 1 - 2 - 3 - 4 - 5 known)

These results hardly correspond with the proposed assumptions, where it was argued that due to the financial history in maintenance the impact should be known better. An explanation could be that the process of inception was well established in the case organizations and also had a separate financial section. The maintenance process did not have a well-established financial evaluation. Further research is advised here. According to the interviewees, however, most money was spent in the maintenance phase. Resource spending is described in Table 38.

Life cycle phase	Case B	Case C	Case D	Case E
Inception	25	29	44	27
Maintenance	70	65	55	68
Abandonment	5	6	1	5

Table 38. Percentage of money spent over the IS life cycle (inception, maintenance and abandoning total is 100%)

It can be concluded that according to the interviewees the financial impact of inception decision making was best known, followed by the maintenance decision making and that the financial impact of abandonment decision making was least well known.

[14] No information concerning Case E was available regarding this item.

Furthermore, several characteristics of abandoning activities were compared with inception and maintenance activities.

Ad (4). Effect on operations.

For all case organizations inception decisions meant something new, whilst maintenance primarily meant something more of the same and abandonment decisions meant something less.

Ad (5). Technological implications of the decision.

In all case organizations, inception primarily meant applying newer technology, whilst maintenance implied similar technology and abandonment implied older technology.

Ad (6). Operational process or project-based.

In all organizations, inception decisions were organized in programmes or projects, whilst maintenance decisions were primarily organized in operational processes. Abandoning activities never concerned regular activities and were always organized in a project or a programme.

Ad (7). Documentation quality.

There were also interview questions concerning the differences in documentation quality within distinct phases of the life cycle. According to the interviewees, the documentation in the inception phase was normally adequate. However, most interviewees mentioned that documentation during maintenance often deteriorates (e.g. when there was urgent patching or maintenance). In the abandonment phase, the documentation was also often limited. The interview results are illustrated in Table 39.

Life cycle phase	Case B	Case C	Case D	Case E
Inception	S	N	U	S
Maintenance	N	U	N	N
Abandonment	U	U	U	U

Table 39. IS documentation quality (Satisfactory [S], Neutral [N] or Unsatisfactory [U])

Overall is concluded that IS documentation quality deteriorates in time. An explanation was given by several interviewees. At the time of inception, the IS was delivered including adequate documentation. As time passes, functional maintenance was performed and this maintenance was often inadequately documented. During the abandonment phase, all interviewees regarded the documentation as inadequate.

Ad (8). Individual prestige (image or status effects).

According to the interviewees inception was considered to provide a higher status in the organization than maintenance. Project manager consolidation (3) said: *"inception has the highest status, you are the hero who solved a problem"*. Abandoning had the lowest status (unless it concerns a major and successful programme with high financial returns). According to the interviewees the job satisfaction was also rated highest within inception, lower within maintenance and lowest within abandoning. Project manager consolidation (3) said: "*inception is a new, novel, challenging and a dynamic environment, ... maintenance is a more stable and predictable environment*". Rationalization programme manager (1) said: "*abandoning is a frustrating activity, that nobody finds really interesting and for, which it was hard to find suitable staff*". According to Architect (6): *"it was celebrated when the new system was operational and fully working, not when the last application was abandoned".* The scores on job status and on job satisfaction are illustrated in Table 40**.**

Life cycle phase	Case B	Case C	Case D	Case E
Status Inception	1	1	1	1
Status Maintenance	2.5	2	2	2
Status Abandonment	2.5	3	3	3
Job satisfaction Inception	1	1	1	1
Job satisfaction Maintenance	1	1	2	2
Job satisfaction Abandonment	1	2	2	3

Table 40. Job status and satisfaction (1 is perceived highest, 3 is perceived lowest)

Furthermore, the activities in the various life cycles phases also required different competencies. Application manager (1) said: *"the art is to put the right person on the right job ... some people like to have new impulses and will fit into the inception phase, others like to have more stable endeavors and will fit in maintenance phase".* Concerning the abandonment phase, these employees should typically possess endurance and the analytical skills to understand all interfaces. Application manager (1) said: "*organizations that abandons legacy IS require staff, which are analytical, like to do archival research and are really dedicated".* An overview of differences between the various life cycle activities that were identified in this research is given in Table 41.

	Inception decision of a new IS	Maintenance decision of an operational IS	Abandonment decision of a legacy IS
(1) Management level of the involved decision maker	Higher management	Lower management	Higher management decides to initiate a programme, lower management decides on single systems
(2) Organizational risks involved.	High risks	Medium risks	Medium to low risks
(3) Financial implications of the decision and responsibility	Financial impact is high. Board is responsible. There is also trade-off between IS and non IS investments	Financial impact is less. In three out of four organizations lower management is responsible and this concerns operational cost	Financial impact is hardly known. The IS owner (e.g. director, business unit director) is always responsible and this concerns disinvestments
(4) Effect on operations	Functionality added	Minor change	Functionality removed
(5) Technological implications	New technology	Same technology	Old technology
(6) Operational process or project-based	Programme/project	Operations	Operations or Programme/project
(7) Documentation quality	Good	Poor	Inadequate or even absent
(8) Individual prestige (image or status effects)	High status Often concerns people that prefer new developments	Mediocre status Often concerns people that prefer routine-like activities	High status when there is a high impact programme, otherwise low status. Often concerns people that do not prefer new and routine-like activities

Table 41. Different decision making characteristics within the IS life

The various distinct characteristics of abandonment decision making compared to inception and maintenance decision making indicate that a separate decision making method seems required. This method could be used in addition to methods within the inception and maintenance decision making. The final four SRQ concern the characteristics of the decision making method.

SRQ 24: Decision making at information function, IS or application level?
In all cases decision making concerned the single IS level. In Case B, there were also decisions made at the level of the information function, due to the mergers abandonment decisions. Methods addressing abandonment decision making should, therefore, always address the individual IS level.

SRQ 25: Financial versus non-financial evaluation aspects?
All the case organizations considered both financial and non-financial aspects important in evaluating legacy IS. This result is in conformance with SRQ 17, where it was concluded that different viewpoints apply, (functional, technical and economical), and which was described in Table 31, (p. 125).

SRQ 26: Many or few attributes in the abandonment decision making method?
Organizations Case B and Case C that abandoned more than thousand legacy IS, had a lot of work to articulate/inventories their IS landscape. They both used an IS attributes database. This information was used in the decision making. In these organizations a pragmatic number of attributes was considered.

SRQ 27: Life cycle or portfolio oriented legacy IS assessment?
Decision making can be based on an individual life cycle base or on a portfolio approach. Both methods are valuable for organizations and fulfill requirements for different management levels. In Case A and Case B which had a merger situation, and Case C, which had an application rationalization approach, portfolio analyses were used. Case D and Case E, which had a less dynamic environment, also had a life cycle approach. It is concluded that in mergers or application rationalization programmes, a portfolio oriented approach seems most suitable. In a more stable environment, a life cycle approach for individual legacy IS might also be suitable.

SRQ 28: Many stakeholders versus few stakeholders involved in legacy IS abandonment decision making?
It is concluded that a formalized decision to abandon a legacy IS should be made by 5-15 stakeholders. In all cases legacy IS decision making stakeholders would always cover the three major viewpoints (business, technology and finance). Preferably an odd number of stakeholders should be used. In Case A and Case B which had a merger situation and Case C, which had an application rationalization approach, more stakeholders were involved in the legacy IS abandonment decision making, compared to Case D and Case E, which had a more stable environment. Finally, it was always the IS owner, who was accountable for the legacy IS abandonment decision.

7.9 Summary and conclusions

This chapter describes the legacy IS abandonment decision making experiences of four validating case study organizations. This chapter is divided in two sections. The first section follows the three predominant life cycle stages (aging process, abandonment decision making process and the practical abandoning process of legacy IS). Researching these life cycle stages built further on the results of the former explorative case research and is mainly deemed inductive. For each predominant life cycle stage, the main research question were answered by SRQ and research questions (see Appendix B.1, p. 178). Subsequently, assumptions regarding the three predominant legacy IS life cycle stages have been validated.

These three life cycle stages together provided input (such as good practices) for the second section of this validation research model which was the research objective: *propose a method to abandon legacy IS*. In this section dilemmas for a method to abandon legacy IS were validated, this is considered to be a deductive approach.

The first life cycle stage identified was: (1) "*the aging process of legacy IS".* The pertaining research question was, *"how do legacy IS age?"* Due to the fact that the existing literature lacks a comprehensive overview of legacy IS aging factors, the aim of this research was to identify all elementary legacy IS aging factors. In the exploratory research (based on the literature and Case A) 140 legacy IS aging factors had already been identified (see Appendix C.1, p. 185). In the validating cases, 27 similar aging factors were found, furthermore, nine new

aging factors were identified and added (see Section 7.5, p. 116). In total 149 unique legacy IS aging factors were identified illustrating the complexity of legacy IS aging (see Appendix C, p. 185). From these 149 aging factors, 14 aging factors were beyond the control of the organization and labeled external, 20 were scored both internal and external and 115 were labeled internal aging factors.

In order to improve the understanding of legacy IS aging, an elementary level of categorization of legacy IS aging factors was chosen. Relative frequency or relative importance did not play a role here, some of the aging factors might be overlapping. These legacy IS aging factors can be further categorized in subsequent research.

It is concluded that internal legacy IS aging factors are more important for business management compared to external legacy IS aging factors, due to the fact that they were within the control of an organization and could have been avoided by means of adequate maintaining the IS.

Swanson and Dans (2000) suggested that management should equilibrate its allocation of system maintenance effort with its estimate of a systems remaining life. An increase in the maintenance effort may extend the systems estimated remaining life. However, the estimate of a shortened remaining life may also call for a reduction on maintenance effort. Because the two variables are related, management should consider each with respect to its decision or estimate of the other (Swanson & Dans, 2000).

The second life cycle stage identified was: (2) *"the abandonment decision making process of legacy IS*". The pertaining research question was "*how do organizations decide to abandon legacy IS?"* Due to the fact that the existing literature lacked a comprehensive overview of legacy IS abandonment triggers, the aim of this research was to identify all elementary abandonment triggers. In the exploratory research (based on the literature and Case A), 32 legacy IS abandonment triggers had already been identified. In the validating cases, 14 similar legacy abandonment triggers were identified, furthermore, six new abandonment triggers were identified and added. In total 38 legacy IS abandonment triggers were identified illustrating the complexity of legacy IS abandonment decision making (see Appendix C, p. 185). From these 38 legacy IS abandonment triggers, seven abandonment triggers were labeled external and beyond the control of the organization, 27 were labeled internal abandonment triggers and four were labeled both internal and external. It is concluded that internal legacy IS abandonment triggers were more important for organizational management compared to external legacy IS abandonment triggers, due to the fact that they were within the control of an organization and could have been avoided by means of adequate maintaining the IS. Due to the fact that there was an overlap between legacy IS aging factors and legacy IS abandonment triggers (13 were mentioned as a legacy IS aging factor and a legacy IS abandonment trigger, by several interviewees), it is concluded that the legacy IS aging factors and the legacy IS aging triggers together cause an IS to age and to be abandoned. Supporting the conceptual legacy IS aging and legacy IS abandonment triggers as illustrated in Figure 25 (p. 91).

Not all types of IS were equally difficult to abandon. Standard purchased packages (COTS) were easier to abandon compared to proprietary IS. IS supporting secondary/supporting business processes were also easier to abandon compared to IS supporting primary business processes. Smaller IS were easier to abandon compared to larger IS.

Furthermore, in total 37 good practices for legacy IS abandonment decision making were identified. In the exploratory research 16 good practices for legacy IS abandonment decision making were identified (see Table 21, p. 77). In the validating cases, eight similar good practices for legacy IS abandonment decision making were identified, furthermore, 21 new good practices for legacy IS abandonment decision making were identified and added (see Table 32, p. 129).

The third life cycle stage identified was: (3) *"the practical abandoning process of legacy IS*". The pertaining research question was "*how do organizations practically abandon legacy IS*?" In total 60 good practices for practical abandoning of legacy IS were identified. In the exploratory research 40 good practices for practical abandoning of legacy IS were identified (see Table 22, p. 80). In the validating cases, 12 similar good practices

for the practical abandoning of legacy IS were identified, furthermore, 20 new good practices for the practical abandoning of legacy IS were identified and added (see Table 32, p. 129).

The research objective, was to "*propose a method to abandon legacy IS*". Design principles were validated by means of dilemmas. Subsequently, differences between the inception, maintenance and abandonment decision making were described and these differences were substantial. There were different stakeholders involved, the information underlying the decision making differed and, also, the decision making processes were dissimilar. Based on these differences, it was concluded that the abandoning of legacy IS includes many distinct characteristics and therefore requires a distinct method.

The method should be at IS level, include financial and non-financial evaluation aspects, use a pragmatic number of attributes, should be used on a portfolio approach and should involve between 5-15 stakeholders. These stakeholders should have different viewpoints (e.g. business, technology and finance stakeholders). The IS owner (e.g. director, business unit director) is always accountable for the decision making to abandon a legacy IS and for the practical abandoning of a legacy IS. If more legacy IS are abandoned (e.g. merger Case A, Case B or an application rationalization Case C) it is proficient to have a dedicated programme which realizes the legacy IS abandoning on behalf of the IS owner.

This research is not aware of the existence of such a method. In the next chapter, such a method will be elaborated upon.

PART 4: DESIGN PHASE

8 PROPOSING A METHOD FOR ABANDONING LEGACY INFORMATION SYSTEMS

8.1 Introduction

Based on the literature and the case organizations, this chapter describes a method to abandon legacy IS based on the design science approach. There are apparent business needs for such a method, because of financial implications of legacy IS (Brodie & Stonebraker, 1995) and the absence of robust methods (Swanson & Dans, 2000; Sellars, 2004; Furneaux & Wade, 2011; Sakthivel, 1994). This chapter first describes design considerations (Section 8.2), then design guidelines (Section 8.3) are presented, subsequently constituent elements of the legacy IS abandonment decision making method (Section 8.4) are proposed. Then roles and responsibilities of associated stakeholders (Section 8.5) are proposed. Subsequently, limitations and recommendations concerning the method are addressed (Section 8.6) and finally this chapter ends with a summary and conclusions (Section 8.7).

8.2 Design considerations

In this section a design science approach has been applied to design a legacy IS abandoning method. According to Hevner, March, Park and Ram (2004): *"design science is fundamentally a problem solving paradigm, it has his roots in engineering and the sciences of the artificial (Simon, 1996)".* The goal of design science is utility, whilst the goals of behavioral science research is truth (Hevner et al., 2004). Design science seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, management, and use of IS can be effectively and efficiently accomplished (Hevner et al., 2004).

Hevner et al. (2004) introduce a framework for design science-based research in the IS discipline. This framework describes that IS research (the development or building of IS theory and artifacts) should be rigorous and relevant. Researchers aim to combine the highest standards of rigor with a high level of relevance Winter et al. (2007), and according to Davenport and Markus (1999) the impact frontier is maximal when both aspects (rigor and relevance) are met. Relevance (Hevner et al., 2004) is based on the environment, which is categorized into people, organizations and technology (this categorization again is further sub-categorized). This environment (Hevner et al., 2004) has business needs for the development or building of a theory or an artifact. Analogous rigor (Hevner et al., 2004) is based on the existing knowledge base, which is categorized into foundations and methodologies (this categorization again is further sub-categorized). This knowledge base has applicable knowledge which should be applied for the development or building of a theory or an artifact (Hevner et al., 2004). Subsequently, a developed theory or artifact should be justified or evaluated. Hevner et al. (2004) provide five design evaluation methods (analytical, case study, experimental, field study and simulation). Finally, IS research (the development or building of IS theory and artifacts) should be applicable in an appropriate environment and should add knowledge to the knowledge base (Hevner et al., 2004). The IS research framework is presented in Figure 29.

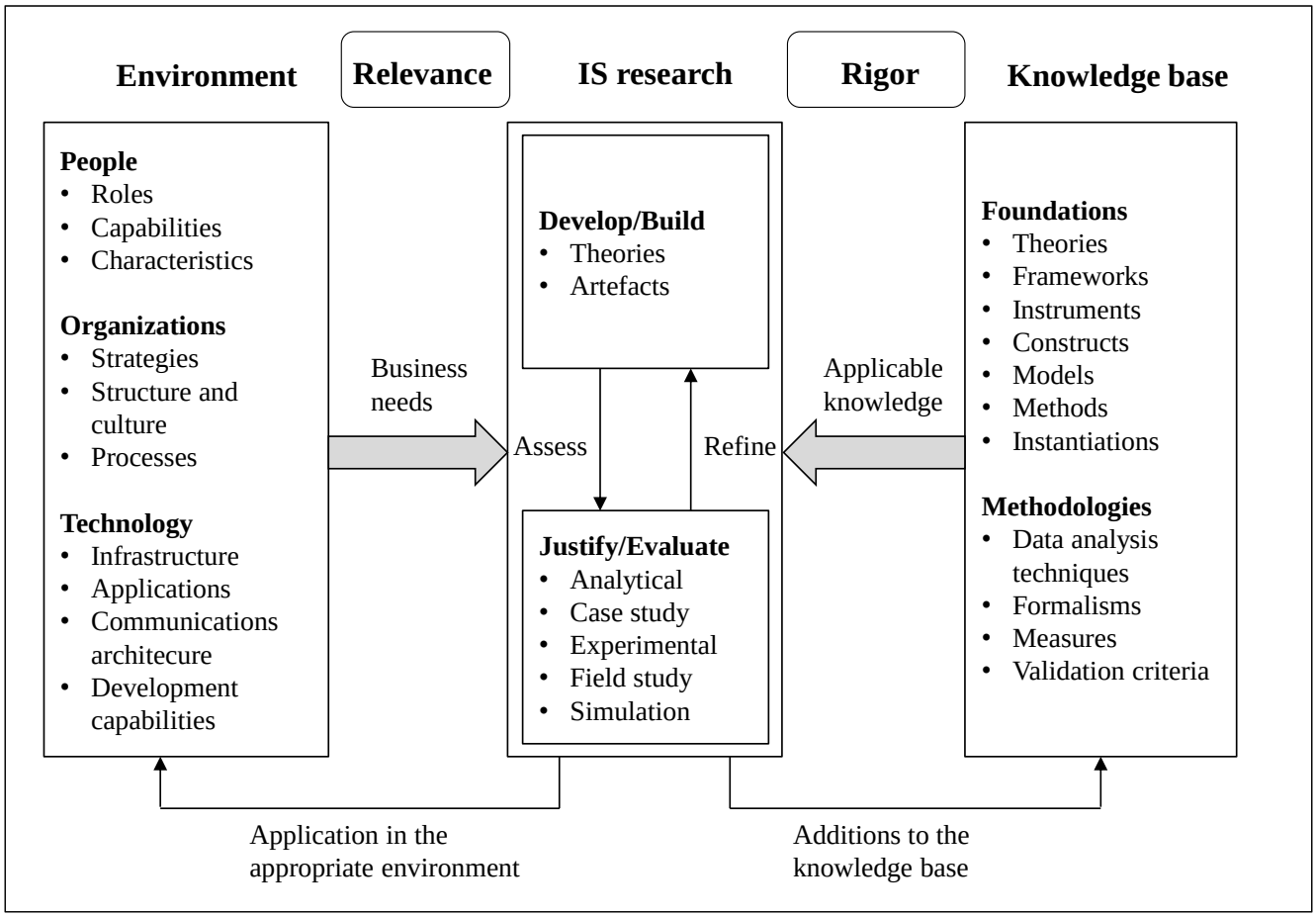


Figure 29. Information systems research framework (Hevner et al., 2004)

8.3 Design guidelines of a legacy information systems abandoning method

In the former section the IS research framework (Hevner et al., 2004) was presented. Hevner et al. (2004) propose seven guidelines for IS design science research; these guidelines are presented in Table 42.

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact design foundations and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Table 42. Design science guidelines (Hevner et al., 2004)

Based on the research of the case organizations, the IS research framework and design science guidelines (Hevner et al., 2004) a method to abandon legacy IS is proposed. Below the design guidelines are discussed.

Guideline 1: Design as an artifact
Design science research emphasizes the production of a viable artifact in the form of a construct, a model, a method or an instantiation (Hevner et al., 2004). This study designs a method for the abandoning of legacy IS. The IS research artifact that is built is: "a method to abandon legacy IS".

Guideline 2: Problem relevance
A lot of organizations have legacy IS. IS designed and developed during the 1960's and 1970's continue to operate and perform in a large number of organizations (Daga et al., 2005). These systems sometimes run on obsolete hardware and software capabilities, are designed in stovepipe fashion, have rigid work processes and their maintenance budgets consume 60-80% of the software related budgets of the organization (Brodie & Stonebraker, 1995). Large organizations nowadays have hundreds of operational IS. There are few indications of the actual number of legacy IS and their impact on society. It is estimated that there are 200 billion lines of COBOL software, accounting for roughly 60% of the total software deployed worldwide (Ulrich, 2002) and that more than 30-billion COBOL-based transactions are processed daily (Lawrence, 2007). Gartner (Hunter & Aron, 2006) states to have indications that legacy portfolios are proportionately 30-50% more expensive to run than their packaged systems or other newer technologies. The social relevance of abandoning legacy IS seems paramount, as legacy IS require large amounts of money, according to the United States Government Accountability Office, GAO (2015, p. 47), federal agencies plan to spend about $58 billion on legacy systems.

In the case organizations the relevance was expressed by several interviewees. For example, according to the Abandonment and migration project manager (1) in Case A: *"there was a business need for a method to abandon legacy IS; due to the fact that there was no legacy IS abandoning method available, we had to learn it ourselves".* In Case B the CIO/ICT director (2) said: *"I have been in the IS field for decades, I always see a growing IS landscape, legacy IS are hardly abandoned, a method to abandon a legacy IS and reduce the size of an IS landscape should be very beneficial for organizations".*

Relevance of business needs is assured by researching the environment and the business needs of this environment for a method to abandon legacy IS. In Case A six information provisionings were consolidated; in Case B three information provisionings were consolidated; in Case C IS were rationalized. Each organization abandoned hundreds of legacy IS and therefore had a need for a method to abandon legacy IS. The need for a method was further expressed by the 38 identified legacy IS abandonment triggers (see Table 29, p. 122) from the case organizations.

Guideline 3: Design evaluation
To evaluate a design, several evaluation methods are available (Hevner et al., 2004); these are described in Table 43.

1. Observational	Case Study: Study artifact in-depth in business environment.
	Field Study: Monitor use of artifact in multiple projects.
2. Analytical	Static Analysis: Examine structure of artifact for static qualities (e.g., complexity).
	Architecture Analysis: Study fit of artifact into technical IS architecture.
	Optimization: Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior.
	Dynamic Analysis: Study artifact in use for dynamic qualities (e.g., performance).
3. Experimental	Controlled Experiment: Study artifact in controlled environment for qualities (e.g., usability).
	Simulation: Execute artifact with artificial data.
4. Testing	Functional (Black Box) Testing: Execute artifact interfaces to discover failures and identify defects.
	Structural (White Box) Testing: Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation.
5. Descriptive	Informed Argument: Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact utility.
	Scenarios: Construct detailed scenarios around the artifact to demonstrate its utility.

Table 43. Design evaluation methods (Hevner et al., 2004)

The proposed legacy IS abandoning method is designed by case study research of five organizations, which summarized abandoned around 2,020 IS. With informed arguments a method to abandon legacy IS is proposed (no experiments, field study or simulation is performed). As a check on the proposed method, all good practices as found in the cases are plotted in one of the elements of the proposed method. If good practices are missed they should be added to the method. These justification/evaluation elements are put into the IS research framework of Hevner et al. (2004) and are illustrated in Figure 30 (p. 147).

Guideline 4: Research contributions
The literature on legacy IS abandonment decision making remains scant, see Section 3.5 (p. 33). A method to abandon legacy IS is also not found in the literature. This research contributes to the knowledge of abandoning legacy IS decision making by designing such a method to abandon legacy IS.

Guideline 5: Research rigor
This research is based on five cases, which abandoned around 2,020 legacy IS. Overall 51 stakeholders have been interviewed; the interviews were supported by a CSP and all the interviews were recorded, transcribed and processed with qualitative software analyses (ATLAS.ti). Research rigor is assured by using: "applicable knowledge". This research uses foundations and methodologies from the literature as presented in Chapter 3, and the validated results in this dissertation as presented in Chapter 7.

Guideline 6: Design as a search process
This study started with a systematic literature review (Chapter 3) and an explorative case study (Chapter 4 and Chapter 5). Based on the results of these chapters a conceptual model concerning aging factors, abandonment triggers, the decision making and the practical abandoning of legacy IS was proposed. In Chapter 7 these concepts were validated. Subsequently, a method is proposed in this chapter. The design based on this method, follows a generate/test cycle (Hevner et al., 2004). The elements from the method were based on the cases and the literature and then modeled. The design follows a heuristic approach which was created by means of informed arguments. By means of ATLAS.ti, the good practices were plotted in the constituent elements of the proposed legacy IS abandoning method and the elements were rearranged. Finally, this resulted in the proposed method to abandon legacy IS.

Guideline 7: Communication of research
The process and outputs are communicated via this dissertation and includes the description and validation of legacy IS aging factors and legacy IS abandonment triggers. Furthermore, this method to abandon legacy IS is described, including good practices.

8.4 Constituent elements of a method to abandon legacy information systems

In this section the design of a method to abandon legacy IS is described. The method draws upon the results of all cases and upon the literature. During the life cycle of an IS, distinct stages can be identified:

1. Operational stage; at this stage legacy IS are subject to internal and external aging factors (the aging process of legacy IS, see Section 7.5, p. 116).
2. Abandonment decision making stage; at this stage internal and external abandonment triggers make organizations decide to abandon legacy IS (see Section 7.6, p. 119). Good practices for legacy IS abandonment decision making are provided (see Table 32, p. 129).
3. Practical abandoning stage; at this stage organizations materialize the abandoning of legacy IS (see Section 7.7, p. 129). Good practices for the practical abandoning of legacy IS are provided (see Table 33, p. 132).

By studying the legacy IS abandoning processes in the cases and the dilemmas as validated in Section 7.8 (p. 132) and the good practices, constituent elements of a method to abandon legacy IS emerge. Based on the first design dilemma: "enhancing existing IS decision making methods or designing a new legacy IS abandonment decision making method?", it is argued that - due to the various distinct characteristics of abandonment decision making compared to inception and maintenance decision making - a separate decision making method for legacy

IS abandoning seems required. The constituent elements of such a method to abandon legacy IS follow a logical order, which is described by the following steps: (1) initialize a centralized IS attributes database; (2) maintain a centralized IS attributes database; (3) score on the legacy IS abandonment triggers of the centralized IS attributes database; (4) evaluate the legacy IS; (5) make the legacy IS abandonment decision; (6) communicate the legacy IS abandonment decision; (7) refine the legacy IS abandonment decision making; (8) materialize the abandoning of legacy IS and parallel: (9) monitor the progress of abandoning legacy IS[15]; (10) evaluate the legacy IS abandoning. These constituent elements of the method to abandon legacy IS are illustrated in Figure 30.

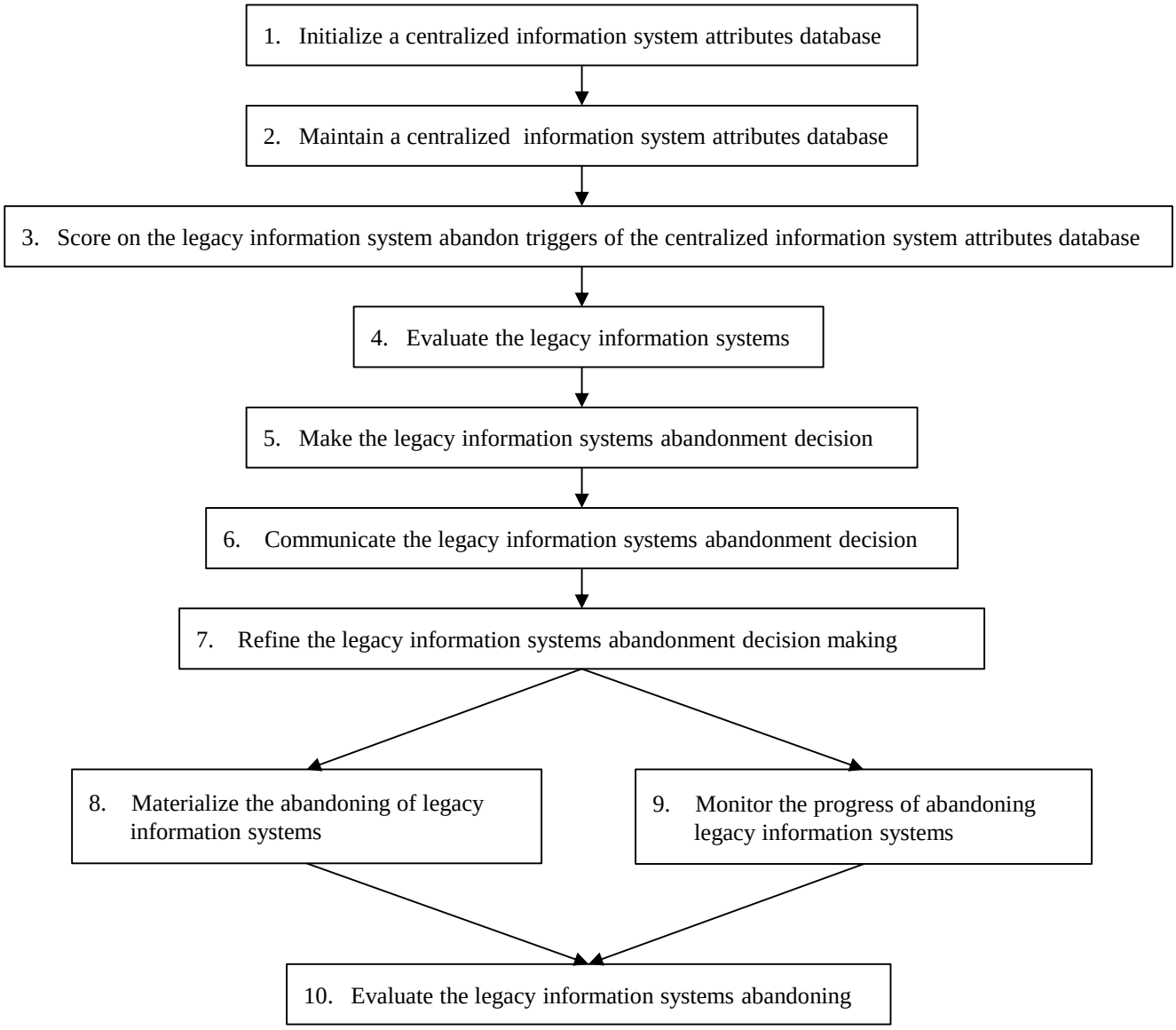


Figure 30. Constituent elements of a legacy IS abandoning method

Each constituent element is considered to be a process. Jacka and Keller (2009) describe a method to design processes. In the following sections each process is described and discussed.

[15] Process (9) monitors the progress of process (8) and are therefore parallel.

8.4.1 Initializing a centralized information system attributes database

All the case organizations had dispersed information in several IS information sources. Information was found in different systems (e.g. a procurement system contained license costs, an hourly system contained hours spent on maintaining IS, an architecture system contained information of business processes and interfaces, a configuration management database contained information of practical IT assets, application registers contained a description of the functionality). Sometimes information was on paper, in documents, on personal or local databases, only in people's mind or not available at all. None of the cases in this research had an overview of all existing IS in the organization. Establishing this overview of existing IS was always the first step in legacy IS abandonment decision making (Case A, Case B and Case C). An overview should be established by incepting an IS attributes database.

Concerning the type of IS attributes in such a database, there should always be generic IS information (e.g. name, age, cost). It is advised to have a unique application name and application number in all local databases, because in the case organizations it was found that in several information sources (e.g. databases, documents, manuals) sometimes different names were used for the same IS (Case A and Case C). In all the cases the evaluation of legacy IS had both financial and non-financial aspects, so the IS attributes database should cover IS characteristics that express both financial and non-financial evaluation aspects.

In Section 7.6 (p. 119) it was validated that there were differences between decision making COTS or proprietary legacy IS and between primary and secondary legacy IS. Therefore, the attributes database should include characteristics which express this difference. There was also a difference in decision making between large and small legacy IS, therefore the attributes database should include characteristics which express this difference. The number of function points is generally accepted as a good indicator for large or small IS. Based on Warren (1999) and Sommerville (2007) also business value attributes are added (e.g. number of users). An example of IS generic information and business value database attributes is provided in Table 44.

Information system generic information											**Business value**		
Application number	Application name	Description and IS owner	Age	Function points	Program language	Maintenance cost	Supplier	Hardware platform	COTS/Proprietary	(P)rimary/(S)ecundary IS	Number of users	Business importance	Required life time
101	WWX	…	30	10,000	COBOL	300,000	A	Bull Mainframe	C	P	5,000	H	2
102	Akti-Z	…	22	60,000	COBOL	50,000	B	IBM AIX	P	P	3,303	M	6
103	FBS	…	3	3,000	C+	34,000	C	Compac Unix	P	S	500	H	1
104	BZZ	…	7	7,000	C+	45,000	F	Compac Windows	P	P	500	L	1
…	…	…	…	…	…	…	…	…	…	…	…	…	…

Table 44. Information system generic information and business value database attributes

In addition to an IS database with generic information about attributes and business value attributes, for legacy IS abandonment decision making the legacy IS abandonment trigger attributes are also important. In the cases 38 abandonment triggers were found. In Table 45 (p. 151) these legacy abandonment trigger attributes are presented in a legacy IS database. When in the table an "X" is scored on a legacy IS abandonment trigger attribute, this means that the IS is a candidate for abandoning.

This IS attributes database consist of "generic information", "attributes expressing the business value" and "legacy IS abandonment trigger attributes". This database should consequently be filled and maintained by different disciplines within an organization. The technical attributes should be filled by stakeholders with a

technical perspective, while business stakeholders are responsible for attributes representing business value and functionality and financial stakeholders are responsible for economical attributes (see Table 31, p. 125). A centralized IS strategy department (e.g. CIO office or architecture department) is responsible for initializing a centralized IS attributes database (this was found in Case A), whereas the CIO should be accountable for the promptness of the data.

In all organizations a form of portfolio management was found regarding the inception of IS. Three organizations (Case A, Case B and Case C) also applied a portfolio approach for abandonment decision making. A portfolio approach is therefore, also suggested for legacy IS abandonment decisions. By means of an IS attributes database, the IS that have IS legacy abandonment triggers are identified very easily. When the decisions are made by means of portfolio analyses, interconnections between IS are also taken into account. Such analysis, always require information from stakeholders close to the IS.

All case organizations required that this centralized IS attributes database should be filled from many dispersed resources. This is illustrated in Figure 31, the arrows illustrate information flow from the source to the destination.

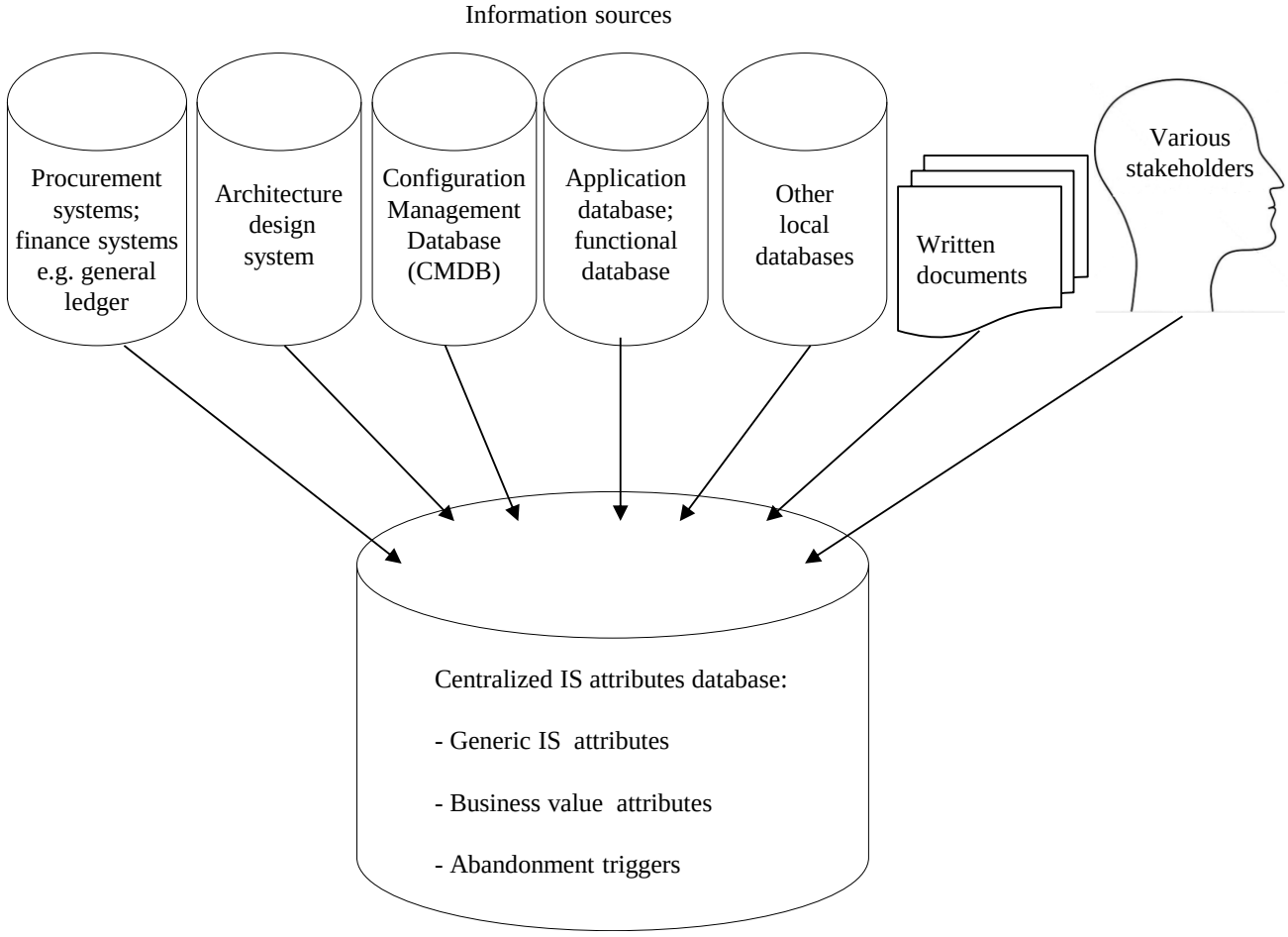


Figure 31. The centralized IS attributes database and various resources

The centralized IS attributes database is filled by collecting data from existing databases (formal, informal or personal databases), desk research, interviews, documents, manuals, robots (e.g. network scans) and physical inspections (Case A, Case B and Case C). In Case A and Case B collection of data required hundreds of men hours. Filling the centralized IS attributes database is the joint responsibility of the centralized IS strategy department (e.g. CIO office or architecture department) and specific stakeholders, who have specific knowledge: technical content by technical stakeholders (e.g. architects), functionality content by business stakeholders (e.g. Business

information managers) and financial content by financial stakeholders (e.g. controllers). The CIO is accountable for filling this centralized IS attributes database. Risks in attaining a centralized IS attributes database are: dispersed information, complexity and the required effort to collect all the data.

It is suggested to have generic information including: the application name of IS, a description and also the number of function points (expressing a small or large application), COTS or proprietary and whether the IS is a primary (e.g. main business process) or secondary (e.g. HRM support) IS. Also the business value should be expressed by attributes (e.g. number of users, business importance) and finally the legacy IS abandonment triggers are added as attributes. Table 45 proposes an IS attributes database, filled by different types of stakeholders (representing technical, functional or economical attributes) and from different existing information sources in the organization. It is based on Table 44 (p. 148) and is transposed for readability and further refined. When in the table an “X” is scored on a legacy IS abandonment trigger attribute, this means that the IS is a candidate to be abandoned.

Generic IS information	Application number.	101	102	103	104	…
	Application name.	WWX	Akti-Z	FBS	BZZ	
	Description (including owner).	…	…	…	…	
	Age (years).	7	3	30	22	
	Function points.	10,000	60,000	3,000	70	
	(C)OTS/(P)roprietary.	C	P	P	P	
	(P)rimary/(S)ecundary.	P	P	S	P	
	Program language.	COBOL	COBOL	C+	C+	
	Maintenance cost.	300k	50k	34k	45k	
	Supplier.	A	B	C		
	Hardware platform.	Bull Mainframe	IBM AIX	Compac Unix	Compac Windows	
Business value	Number of users.	5,000	3,303	500	500	
	Business importance.	H	M	H	L	
	Required lifetime.	2	6	1	1	
Legacy IS abandonment triggers see Appendix C.5 (p. 198)	High IS cost.	X				
	Merger or consolidation.					
	Abandoning products or services supported by the IS.		X			
	Aged/old hardware.					
	Redundant application functionality.					
	New technological opportunities.					
	Limited or ended supplier support.				X	
	Application functionality gap/not sufficient.					
	Redundant data centers.					
	Redundant IS.					
	High cost per user or customer.					
	Non-scalability of the IS.					
	Inadequate software quality.					
	Application functionality no longer needed.					
	Capability shortcomings.			X		
	Legal compliance (law).					
	Does not fit in technological strategy.					
	Instability of the IS.					
	Technical aging.					
	The IS is non-agile.					
	Structural lack of IS maintenance.					
	Vendor dependency/vendor lock-in.					
	Effort to migrate applications from an old data center to a new data center.					
	Redundant data sources.					
	Perception of the required IS capability is changed.					
	Capability not needed.					
	New demands from the market, which cannot be provided by the IS.					
	Infrastructure overhaul.					
	Limited or no technical skills available.					
	Cost-benefit.					
	Security issues.					
	Complexity (+/-).					
	Enormous proliferation of (unknown) information systems in unknown (different) places.					
	Outsourcing, first clean your IS landscape.					
	Standardization.					
	Internal directive COTS.					
	Data is no longer needed.					
	Momentum, opportunity for a new organization.					

Table 45. Attributes of a centralized legacy IS database

8.4.2 Maintaining a centralized information system attributes database

After the initial setup of the centralized IS attributes database, the goal is to have a complete overview of the IS assets covering the generic and business value attributes. Due to the fact that the IS landscape is changing in time, the centralized IS attributes database should also be maintained. Therefore, it is advised, that the centralized IS strategy department in the organization (e.g. a CIO office or architecture department) is responsible for writing a procedure for maintaining the centralized IS attributes database. This procedure should be approved by the organization. The CIO should be held accountable for these processes.

According to this procedure the IS attributes database should be kept "up-to-date". Maintaining the centralized IS attributes database should be the joint responsibility of the centralized IS strategy department (e.g. a CIO office or an architecture department) and specific stakeholders with specific knowledge; technical content by technical stakeholders (e.g. architects), functionality content by business stakeholders (e.g. Business information managers) and financial content by financial stakeholders (e.g. controllers), see Table 31 (p. 125). The CIO again is accountable. The quality of the IS attribute database can be measured by checking whether a maintenance procedure has been approved by the organization and whether this database has been recently updated.

8.4.3 Scoring the legacy information system abandonment triggers

The objective of this step is to score the legacy IS abandonment triggers in the centralized IS attributes database. Therefore, it is advised, that the centralized IS strategy department in the organization (e.g. a CIO office or an architecture department) is responsible for writing a procedure for scoring the legacy IS abandonment triggers in the centralized IS attributes database. This procedure should describe who completes scores and how these scores should be approved. This procedure should be approved by the organization. The CIO should be held accountable for these processes.

It is suggested that the procedure prescribes to periodically score all the IS on the legacy IS abandonment triggers. The centralized IS strategy department and stakeholders from several departments (technical, functional and economical) are responsible for completing the IS attributes database by scoring the IS legacy abandonment triggers into the centralized IS attributes database (see Table 31, p. 125). The CIO is accountable. Redundant application functionality, which is an abandonment trigger, is found by plotting application functionality on business processes. This can be done by architects, if more IS support similar functionality, this redundancy becomes visible (Case A, Case B and Case C). A business risk is that this scoring is done by for instance an architect who also developed the IS, resulting in biased scores. Whether this step is completed, can be checked by the availability of a physical centralized IS attributes database with the scored legacy IS abandonment triggers and a formalized process to score the legacy IS abandonment triggers. A measure of success are the stakeholders that actually scored IS abandonment triggers in the centralized database (e.g. 95%).

8.4.4 Evaluating the legacy information systems

The objective of this step is the evaluation of legacy IS. Therefore, it is advised, that the centralized IS strategy department in the organization (e.g. a CIO office or an architecture department) is responsible for writing a procedure for evaluating the legacy IS. This procedure should describe how the evaluation is performed and should be approved by the organization; the CIO is accountable. The evaluation process is subsequently described below.

Based on the centralized IS attributes database (including a score on legacy IS abandonment triggers) the centralized IS strategy department (e.g. CIO office or architecture department) and technical, business and financial stakeholders are responsible for analyzing and evaluating the legacy IS in the IS attributes database. This is done by making queries, interpreting data, input of missing data and discussions among stakeholders. The output of this evaluation process should include scenarios as to when to abandon which legacy IS. Prioritization and a planning is part of this process as not all legacy IS can be abandoned at once (e.g. Case A took six years to abandon all legacy IS). In this phase the comparison and evaluation of alternatives for

abandoning legacy IS are performed (Harrison, 1999, see Figure 3, p. 8). This results in a proposal for higher management. These proposals should include the following items: null alternative (what happens if nothing is changed) an alternative in which the IS is continued (more maintenance), and an alternative in which the IS is abandoned. The CIO should be held accountable for these processes.

Depending on the situation (a consolidation of organizations, a merger of information functions or single IS abandonment decision making), different hierarchies of business cases were identified. In case of a consolidation, merger or application rationalization programme, first a generic strategy is required. Subsequently, individual business cases for individual IS could be made (Case A, Case B and Case C).

This evaluation process should typically be done once or twice a year. A measure of success is the abandonment percentage of legacy IS compared to all IS. In case of a consolidation, for example three organizations consolidate, this could be as high as 66% of all IS. In normal situations this could be around 5% of all IS.

8.4.5 Making the legacy information systems abandonment decision

The objective of this step is the decision to abandon legacy IS. Regularly (e.g. once or twice a year) a meeting is organized by the CIO. Stakeholders of this meeting are the CFO and IS owners (e.g. director, business unit director). The process of abandonment decision making is facilitated on behalf of the CIO (who is accountable) by a manager of the centralized IS strategy department (e.g. Corporate portfolio manager, Chief architect, Corporate information manager). There are typically 5-15 stakeholders, who participate in the decision when to abandon which legacy IS.

A high-level business case should be used to support the legacy IS abandonment decision making. The decision maker accountable for abandoning a legacy IS, is the IS owner (e.g. director, business unit director). Pressure on the expense budgets of these IS owners, will accelerate these abandonment decisions. Every abandoned IS will no longer consume scarce resources, e.g. money and staff (Case A, Case B and Case E). Good practices for legacy IS abandonment decision making are provided (see Table 32, p. 129). A measure of success is the number of legacy IS to be abandoned. The case organizations typically include between five and 15 stakeholders; according to the interviewees this team size was sufficient to cover all required knowledge.

8.4.6 Communicating the legacy information systems abandonment decision

The objective of this step is to inform users, clients and suppliers about the decision to abandon legacy IS. When the decision to abandon a legacy IS has been made, this decision should be communicated to stakeholders (e.g. users, maintainers, suppliers). The communication department is accountable for making a legacy IS abandoning communication plan. Strong sponsorship was defined as a success factor. It is therefore suggested that the abandonment decision of a legacy IS is communicated by, or on behalf of a senior business manager, e.g. director, business unit director, CFO or CEO (Case A, Case B and Case C). Communication can be done through intranet, personal communications and meetings concerning the abandoning of a legacy IS; the IS owner is accountable for the communication. After the decision has been made and communicated, typically a lot of resistance against these decisions is experienced (Case A, Case B and Case C). A communication plan should therefore always be available and activated.

8.4.7 Refining the legacy information systems abandonment decision making

The objective of this step is to have detailed abandoning working plans. After the legacy IS abandoning plans have been approved by the highest management in the organization and the legacy IS abandonment decision has been communicated to the stakeholders, the plans to abandon the legacy IS should be further refined. The initial business case is further enhanced, including a detailed legacy IS abandoning plan and data orphanage plan (see Appendix D, p. 201). The centralized IS strategy department (e.g. CIO Office or architecture department) and stakeholders representing technical, business and financial departments should be responsible for making such a plan. The IS owner should always be accountable for abandoning the legacy IS. Due to the fact that the IS

owners, (e.g. director, business unit director) were busy performing their primary business process, focus was not on the practical abandoning of legacy IS (Case C and Case B). Although the IS owner (e.g. director, business unit director) was accountable for the abandoning, a separate centralized programme can help to realize the practical abandoning. A central programme to abandon the legacy IS (legacy IS abandoning skills are available) on behalf of the accountable IS owner can be incepted (Case A, Case B and Case C).

Additional information is included in the centralized IS attributes database, such as, all interfaces with other IS and contracts with suppliers. According to Application manager (1) this requires special skills: *"responsible staff should like to analyze archival files, should be analytic and expose endurance".* Depending on the state of documentation of the complete IS landscape, this could include a lot of work. According to the IS owner (1): *"we had to untangle the information function, it was like spaghetti and we discovered a lot of interfaces with other IS".* It was found that investing in research upfront pays off later by not missing interfaces or having other surprises (Case B). The work in this phase often identifies additional applications that were overlooked in earlier stages. In case part of systems or support are outsourced to suppliers, contract management is also important. Because suppliers will lose revenue, they might be less supportive. Furthermore, these same suppliers are needed to abandon the legacy IS (Case A and Case B).

The same applies for internal resources. In case staff is needed to abandon their own legacy IS, offering them a new challenge in the organization remains a success factor. If this is not possible a good exit plan is important (Case A). Staff responsible for the practical abandoning of legacy IS should not discuss the decision to abandon the legacy IS with staff close to the legacy IS. A good practice is to refer to the higher management decision that the legacy IS has to be abandoned (Case A).

Also procedures should be available on how to handle disagreements or new information (e.g. from stakeholders not agreeing with the abandonment decision). Also organize a process (Case B) to reverse the decision to abandon a legacy IS. This process should deliver a document, including arguments (e.g. a business case) why the legacy IS should not be abandoned. Again the IS owner (e.g. director, business unit director) decide, based on new information, to postpone, or not to abandon the legacy IS.

8.4.8 Materializing the abandoning of legacy information systems

The objective of this step is to actually abandon the legacy IS. The IS owner (e.g. director, business unit director) should always remain accountable for the abandoning of legacy IS (all cases). The rationale for this, is that IS support business processes and the CIO will typically be unaware when to unplug an IS supporting a business process. The realization of the practical abandoning can be done by a programme (Case A, Case B and Case C) or a project (Case D). In that case the IS owner delegates responsibilities to a programme or project manager. The abandoning is executed according to approved plans. If there is an exception, this is reported to the board and the original plan should be adjusted (an exception plan) and reapproved (Case A). The IS owner should be accountable for the exception plan.

The actual abandoning is often considerably later than the initial decision making to abandon the legacy IS, for instance 6 years in Case A. Furthermore, it was suggested to start with the low risk and simple IS and to cluster similar IS, in order to learn from the process (Case B). A standardized process should be developed (Case A, Case B and Case C). When the practical abandoning date is approaching (for example a period of four weeks in advance) the IS owner (e.g. director, business unit director) should sign the disposal plan (see Appendix D, p. 201). Only after this confirmation, the practical abandoning should be realized, because the legacy IS might still support operational business processes (Case A and Case B).

After this signature, first cut off any users from the IS and observe a cooling off period. Furthermore, make sure there is always a recovery plan in case anything goes wrong (Case B). If nobody complains (e.g. four weeks), save the data and dismantle the IS (Case B). Data left on devices should be either destroyed or stored. A data officer from the documentary information department is responsible for this decision (Case A). Subsequently, hardware

and software can be dismantled (the CIO is accountable). In the case of outsourcing, make sure that contracts with suppliers are ended. This is the accountability of contract management (Case A). Update the centralized IS attributes database and local databases (e.g. Configuration Management Database, procurement systems). A centralized IS strategy department (e.g. CIO office or architecture department) is responsible for this update and the CIO should be accountable.

Avoid acting on rumors during this process and have a process in place to restart the legacy IS, if needed (Case A, Case B and Case C). It is advised to apply a retirement checklist or disposal plan (see Appendix D, p. 201). Good practices for the practical abandoning of legacy IS are provided in Section 7.7 (p. 129). A measure of success for this process is the number of abandoned legacy IS, that had to be restarted, because there were unexpected operational problems.

8.4.9 Monitoring the progress of abandoning legacy information systems

The objective of this step is to monitor the actual abandoning of legacy IS, versus the plan of abandoning legacy IS, and take appropriate action if there are deviations (follow up and control, Harrison 1999, see Figure 3, p. 8). Monitoring the progress of the abandoning of legacy IS seems to be very important, due to the fact that there can be quite some time lapse between the decision to abandon a legacy IS and the actual abandoning (Case A and Case B). Also because of the fact that focus in organizations is frequently on new and operations and not on abandoning legacy IS (Case B).

Therefore it is important to monitor the realization versus plan of abandoning the legacy IS and to report regularly (e.g. monthly base) about its progress (Case A, Case B and Case C). These activities are the responsibility of the control department. Several progress reports were found at different hierarchical levels (Case A), at the lowest hierarchical level this was the individual legacy IS. There is a project plan with milestones to abandon the individual legacy IS. There is reporting against this plan. At management level there is progress reporting of sets of legacy IS, for instance, by means of colors lists indicating the progress of the actual abandoning versus the abandoning plan (red, orange and green). These lists consist of the legacy IS name, legacy IS number, planned abandoning date, actual abandoning date, progress indicated by color, the expected savings and possible remarks (Case C). Table 46 illustrates the categories, which were identified in Case A. Case A also reported the planned savings versus actual savings.

Application number	Application name	IS owner (director, business unit director)	Hardware data center	Planned practical abandoning date	Actual abandoning date	Status G(reen), O(range) and R(ed)	Monthly saving	Saving per year	Remarks
…	…	…	…	…	…	…	…	…	…
123	ISP	Division X	Data center K	01-07-2009	01-06-2009	G	€ 10k	€ 120k	…
167	IBG	Division Y	Data center L	01-08-2009	01-10-2009	R	€ 5k	€ 60k	…
251	IX2	Division Z	Data center M	01-11-2009	01-11-2009	O	€ 5k	€ 60k	…
…	…	…	…	…	…	…	…	…	…
							Total	€ …	

Table 46. Abandoning progress reporting Case A

It is suggested that monitoring and reporting should primarily be done by independent controllers from the control department. Furthermore, it is proposed that this progress should be clearly communicated (Case A). This could be the responsibility of the communication department. Finally, in case there are deviations in the original plan versus realization, the IS owner is responsible for taking appropriate actions. The comparison of the actual versus planned abandoning of legacy IS could be a measure of success for this activity.

8.4.10 Evaluating the legacy information systems abandoning

The objective of this step is to evaluate the legacy IS abandoning and to improve the process of abandoning legacy IS. To improve the process it is suggested to evaluate the abandoning process including the decision making and the lessons learned. The centralized IS strategy department (e.g. CIO office or architecture department) is responsible for evaluating the legacy IS abandoning. Finally it might also be worth celebrating the abandoning of IS, similar to the inception (Case A). The CIO is accountable for performing this evaluation.

In summary, a method to abandon legacy IS is described in section 8.4.1 – 8.4.10. The first process (step 1), the initialization of the centralized IS attributes database is initially done. The next process (step 2), maintaining the centralized IS attributes database is advised to be an ongoing process, to ensure that the legacy IS attributes database remains "up-to-date". Steps 3 to step 7 are advised to be executed once or twice per year. Steps 8 to step 10 are advised to be continuous processes.

8.5 Roles and responsibilities of associated stakeholders in the abandoning method

The former section described the constituent elements of an abandoning method, the identified stakeholders and their roles in particular processes. This section provides by means of a RACI matrix (Jacka & Keller, 2009) an overview of the abandoning method and roles and responsibilities of the associated stakeholders. According to Jacka and Keller (2009) a RACI matrix is a visual representation of individual roles in particular processes identifying those who are **R**esponsible, **A**ccountable, **C**onsulted and **I**nformed (RACI). Table 47 describes roles and responsibilities in each phase of the method to abandon legacy IS of the associated stakeholders.

Process number) Process name	**Process units.**	**Stakeholders**										
		Centralized IS strategy department (e.g. CIO Office or architecture department)	Technical (e.g. IT architect), Business (e.g. Business information manager) and Financial stakeholders (e.g. Financial controllers)	CIO	Project or Programme	e.g. Board (business directors, CFO,CEO)	IS owner (director, business unit director)	Users, clients, suppliers	Control department	Communication department	Contract management	Data officer
1) Initialize a centralized IS attributes database.	Initialize a centralized IS attributes database.	R	C	A		I						
	Fill centralized IS attributes database.	R	R	A		I						
2) Maintain a centralized IS attributes database.	An approved procedure for maintaining the centralized IS attributes database.	R	C	A		I						
	Maintain the centralized IS attributes database.	R	R	A								
3) Score on the legacy IS abandonment triggers of the centralized IS attributes database.	An approved procedure for scoring abandonment triggers on the legacy IS centralized attributes database.	R	C	A		I						
	Score legacy IS abandonment triggers.	R	R	A								
4) Evaluate the legacy IS.	An approved procedure for evaluating legacy IS.	R	R	A								
	Analyzing and evaluating IS landscape.	R	R	A								
	Scenario building, prioritization and planning.	R	R	A		I						
5) Make the legacy IS abandonment decision.	Facilitate the legacy IS abandonment decision making.	R	C	A		I	C					
	Legacy IS abandonment decision.			R		R	A					
6) Communicate the legacy IS abandonment decision.	Create communication strategy and plan.	C	C	R						A		
	Communication of the decision to abandon legacy IS.						A	I		R		
7) Refine the legacy IS abandonment decision making.	Refine plans to realize the practical abandoning of legacy IS.	R	R	A			R					
8) Materialize the abandoning of legacy IS.	Realizing the abandonment of legacy IS.	R	C		R		A					
	Exception plan.	R		R	R	I	A					
	Approval of the practical abandoning of the legacy IS (IS disposal plan).	R	R	R	R		A					
	Store or destroy data.	R		A	R		R					R
	Dismantle hardware and software.	C	C	A	R							
	End contracts with suppliers.	R	R						R		A	
	Update centralized IS attributes database.	R	R	A								
9) Monitor the progress of abandoning legacy IS.	Monitor abandoning realization versus plan.								A	I		
	Report (e.g. monthly) abandoning realization versus plan.	I	I	I	I	I	I		A	I		
	Communicate progress of abandoning legacy IS.								C	A		
10) Evaluate the legacy IS abandoning.	Evaluate and improve legacy IS abandoning procedures.	R		A						I		
	Celebrate successful abandoning and appraise team members.			R			A			I		

Table 47. Roles and responsibilities from associated stakeholders; Responsible (R), Accountable (A), Consulted (C) and Informed (I)

8.6 Limitations

The proposed method is based on the experience and knowledge gathered from all five case studies. There are three organizations that went through a consolidation or a merger (Case A, Case B and Case C) and abandoned around 2,020 IS, and they incepted dedicated programmes. There were two case organizations without a consolidation or a merger background; these two organizations only abandoned limited numbers of legacy IS which (Case D and Case E).

This presented method draws heavily upon the three organizations, which abandoned around 2,020 IS; and gained ample experience. In the three organizations (Case A, Case B and Case C) substantial programmes were initiated, and there was also involvement from the board of these organizations. In the other two organizations there was less board involvement.

Ultimo 2016, the proposed method has been shared and discussed with five stakeholders, of these five, four were interviewed earlier. The method has been discussed, shared under condition of non-disclosure and interviewees were asked to comment on the proposed method. No mayor comments on the proposed method were provided.

It is uncertain whether the proposed method would also fit smaller organizations or organizations without a consolidation or a merger background, e.g. those abandoning only a few legacy IS. This should be subject to further research.

8.7 Summary and conclusions

Based on the case organization research, a method is proposed consisting of 10 elements: (1) initialize a centralized IS attributes database; (2) maintain a centralized IS attributes database; (3) score on the legacy IS abandonment triggers of the centralized IS attributes database; (4) evaluate the legacy IS; (5) make the legacy IS abandonment decision; (6) communicate the legacy IS abandonment decision; (7) refine the legacy IS abandonment decision making; (8) materialize the abandoning of legacy IS and parallel: (9) monitor the progress of abandoning legacy IS; (10) evaluate the legacy IS abandoning. Also roles and responsibilities are provided. In each phase stakeholders and their activities are described. The design of the legacy IS abandoning method is evaluated by the design guidelines of Hevner et al. (2004). Finally limitations, of this method are provided.

9 CONCLUSIONS AND RECOMMENDATIONS

"A conclusion is the place where you got tired thinking".
---Fisher, M. H. (1945)---

9.1 Introduction

Like other systems, IS typically follow a life cycle, they are incepted, operated, maintained and finally abandoned (Berghout & Nijland, 2002). Throughout the years, the IS will increasingly resist meeting the requirements of the organization. This increasing resistance is referred to as "aging" of the IS (Swanson & Dans, 2000; Lehman, 1980a) and these systems are often referred to as legacy IS. Eventually these legacy IS are abandoned. Research on abandoning legacy IS remains scant (Furneaux & Wade, 2011; Sellars, 2004; Swanson & Dans, 2000). In the systematic literature review of this research, only twelve papers were identified, which described studies in this research area (see Section 3.3, p. 26). These studies provided limited guidance to identify legacy IS and did not include methodological support as to how to abandon these legacy IS.

The practical relevance of adequate theory on abandoning legacy IS is also paramount as legacy IS consume large amounts of resources. USA federal agencies, for instance, planned to spend about $58 billion on legacy systems in 2015 (Government Accountability Office, GAO, 2015, p. 47). Furthermore, legacy IS prevent organizations from fully exploiting computers and the value they bring to organizations (Ulrich, 2002). They may hinder progress and lack agility (Daga et al*.,* 2005) or remain barriers to strategic innovation (Kelly et al*.,* 1999). Legacy IS also put organizations at risk due to, increasing complexity (Kelly et al*.,* 1999; Richmond et al*.,* 2006; Ulrich, 2002), increasing security issues (Alderson & Shah, 1999), declining system reliability and diminished support from the supplier (Furneaux and Wade, 2011).

The need to improve our understanding of legacy IS and abandonment decision making was addressed in Chapter 1. This chapter also included the research definitions concerning "information systems (IS)", "legacy IS" and "decision making" and described why additional methodological support was necessary for abandonment decision making. Finally this chapter formulated the primary research question: *"how do organizations identify legacy IS and how do they manage abandonment?"* In Chapter 2, the research design of this study was provided, which was based on the research "onion" of Saunders et al. (2007). Argumented choices concerning the research philosophy, research approach, research strategy, research choice, time horizon, technique and procedures were provided. In Chapter 3, first, the concept of information system management was explored, including life cycle management and portfolio management, followed by a systematic literature review on legacy IS decision making. In Chapter 4, the antecedents of explorative Case A were described. Case A concerned a complex public consolidation of six similar branch organizations that abandoned 471 applications and four data centers. This case study included two trails of research. Trail 1 described the merger of six information functions (Section 4.3) and Trail 2 described the practical abandoning of 471 legacy IS (Section 4.4). This research was based on five years of longitudinal research. During this longitudinal research, the researcher was also able to capture adapted insights and arguments during this period. In explorative Case A, many more legacy IS were abandoned than originally planned. The original plan in 2006 was to abandon 260 applications. However, in 2011, effectively 471 applications were abandoned. In Chapter 5, the data analysis of the exploratory research at Case A was described. In this chapter the subsequent life cycle process stages of legacy IS were researched, being the (1) *aging process of legacy IS*, (2) *abandonment decision making process of legacy IS* and (3) *practical abandoning process of legacy IS*. The associated research questions were: (1) *"how do legacy IS age?"* (2) *"how do organizations decide to abandon legacy IS?"* and (3) *"how do organizations practically abandon legacy IS?"* In Chapter 6, based on the literature study and longitudinal study (Case A), first the aging process of legacy IS was conceptualized (Section 6.2), then abandonment triggers, decision making process (Section 6.3) and the practical abandoning process of legacy IS (Section 6.4) were conceptualized. Finally, design dilemmas regarding legacy IS abandonment decision making methods (Section 6.5) were described. In Chapter 7, the aging, abandonment decision making and practical abandoning of legacy IS were further explored by means of inductive research in

four case studies. Furthermore, design dilemmas for a method to abandon legacy IS were validated by means of cross-sectional deductive research in four validating case studies. Case organization B was a private organization and the result of a consolidation of three organizations. In this organization 850 applications were abandoned. Case organization C was a public organization, that used to operate fifteen independent (regional) business departments and initially, each business department had their own hardware, data center and software applications. Case C initiated a programme to reduce its number of applications from 2200 to 800. Case organization D was a public organization that abandoned a single large COBOL legacy IS. At Case organization E, which was a public organization, only the abandoning process was researched and no specific legacy IS was abandoned at the time of research. In Chapter 8, the research objective: *"propose a method to abandon legacy IS*" was further addressed. Based on the design science approach (Hevner et al*.,* 2004), a legacy IS abandoning method was proposed. The method consists of 10 constituent elements (see Section 8.4, p. 146) and was validated by informed arguments. Overall, this study was based on 51 interviews, five case organizations and the abandoning of 2,020 IS. The three predominant life cycle process stages of legacy IS will be used to further present the research findings in the subsequent Sections 9.2, 9.3 and 9.4. Subsequently, the legacy IS abandoning method will be discussed in Section 9.5. The primary research question will be revisited in Section 9.6. Furthermore, the external validity of this research (Section 9.7) and directions for further research (Section 9.8) are discussed.

9.2 The aging process of legacy information systems

The first research question was:

"How do legacy IS age?"

Although there was common understanding that IS may include more or less legacy characteristics, the existing literature lacks an overall overview of the characteristics that make IS resist meeting the requirements of the organization. This research had the opportunity to investigate five complex organizations that overall abandoned around 2,020 legacy IS. In these cases and from the literature, overall 149 unique aging characteristics of legacy IS were identified, which referred to the entire IS or its constituent components (applications, hardware, data sets, procedures and people). From these 149 legacy IS aging characteristics, 35 originated from the literature and 75 from the five case studies (39 legacy IS aging characteristics were found both in the literature and the cases). These legacy IS aging characteristics were also categorized into internal and external aging factors. Internal aging factors were within the control of the organizations and could be influenced and external aging factors remained outside the control of the organization and could not be influenced. Of the 149 legacy IS aging characteristics, 14 aging characteristics were considered to be external aging factors, 20 were considered both internal and external and 115 were considered internal aging factors.

This research was particularly suited to identify the most detailed level of legacy characteristics. The 149 legacy IS aging characteristics identified in this study, illustrate the complexity of IS aging. IS obtain additional legacy characteristics as time progresses and their context changes. At some point in time they were referred to as "legacy system", although in this research nobody could precisely indicate the turning point between legacy and non-legacy IS. The turning point from IS to legacy IS can be described when an IS "*significantly resists meeting organizations' requirements".* A tentative legacy IS aging model was provided in Figure 23 (p. 89). This model illustrates that in time legacy IS aging factors accumulate. Due to reengineering, the legacy IS can be rejuvenated. However, in time, the legacy IS aging factors keep accumulating. When the number of IS aging characteristics passed a certain (legacy IS aging) threshold, the legacy system was abandoned. Because many of these characteristics were beyond the control of the organization, the predicting of this threshold seems virtually impossible. Given the above complexity of aging legacy IS, it also appeared to be extremely difficult to pinpoint discrete categories of healthy and less healthy IS, or legacy and non-legacy IS. This did not imply that it was impossible to manage legacy IS. It was concluded that management of legacy IS should always start with a comprehensive database of all IS. Further research to validate this legacy IS aging model is recommended.

Looking at other fields of research might be an option, gerontology (the study of human aging) studies the changes that occur between the attainment of maturity and the death of the individual.

9.3 The abandonment decision making process of legacy information systems

The second research question was:

"How do organizations decide to abandon legacy IS?"

In this research the triggers that make organizations actually decide to abandon legacy IS were identified. In the cases and from the literature, overall 38 unique legacy IS abandonment triggers were identified, which may refer to the entire IS or its constituent components (applications, hardware, data sets, procedures and people). From these 38 unique legacy IS abandonment triggers, four originated from the literature and 27 from the five case studies (seven legacy IS aging characteristics were found both in the literature and in the cases) (see Table 29, p. 122). From these 38 abandonment triggers, seven abandonment triggers were labeled external and beyond the control of the organization, four were labeled both internal and external and 27 were categorized internal abandonment triggers (see Appendix C.3, p. 193 and Appendix C.4, p. 198). Consequently, this research was able to add many new abandonment triggers to the existing literature.

There was an overlap of 13 similar legacy IS aging factors and legacy IS abandonment triggers found in the exploratory research and the validation research. For example, *"high IS cost"* was mentioned as a legacy IS aging factor, but also mentioned as a trigger to abandon legacy IS. Legacy IS become increasingly expensive during the years (Sakthivel, 1994) and this was also described by interviewees as an IS legacy characteristic. However, when the cost pass a certain threshold, this was also a trigger to abandon a legacy IS. It was concluded that both legacy IS aging factors and the legacy IS aging triggers accumulated. However, it was always one of the 38 triggers, which made organizations decide to abandon a legacy IS. This was illustrated in Figure 25 (p. 91).

Some of the 38 abandonment triggers were also overlapping. This research focused on identifying the legacy IS abandonment triggers at the most detailed level. For example, interviewees mentioned "*high IS cost"* as an abandonment trigger, while others referred to *"cost-benefit"* as an abandonment trigger. These triggers obviously had relations. One equivocal (legacy IS) abandonment trigger "complexity" was identified. According to the interviewees this was a trigger to abandon a legacy IS, but was also mentioned as a trigger not to abandon legacy IS.

In legacy IS abandonment decision making, three eclectic decision making perspectives were predominant: the functional, technical and economical perspectives. All triggers could also be categorized in these three perspectives (see Table 31, p. 125). From the validating case studies it was, therefore, concluded that stakeholders representing functional, technical and economical departments should be participating in the decision making to abandon legacy IS. And even though legacy IS decision making was complex, a decision making group of 5-15 stakeholders seemed to be sufficient for all cases. In all the cases of this research, the IS owner (e.g. director, business unit director), was accountable for the decision to abandon legacy IS. Business interests seemed to be more important than IT-related interests in abandonment decision making.

According to interviewees, the financial impact of abandonment decisions was less clear than the financial impact of an investment in new IS. However, this difference could not be explained. This observation contradicts the overall assumption that there should be additional uncertainty regarding the financial consequences of incepting a new IS, which come without any history and financial track record, compared to abandoning legacy IS, which always has some form of history and financial track record. An explanation for this observation could be that IS cost accounting methods and IS asset management seemed hardly developed. Further research is suggested here.

Not all types of legacy IS were equally difficult to abandon. It was observed that it was easier to abandon commercial off-the-shelf systems (COTS) compared to proprietary (own build) IS. This was due to the fact that there was less affinity with COTS, compared to proprietary systems. There were also always more organizations that abandoned similar COTS, so there was always more knowledge available about how to abandon particular COTS. It was also concluded that there was a difference between the abandonment decision of primary business legacy IS and secondary (supportive) legacy IS (e.g. a Finance or HRM IS). Secondary legacy IS encompassed less impact. There were also often more opportunities for replacement due to the fact that secondary legacy IS were often used by many organizations (e.g. a general ledger IS), resulting in competing suppliers and scale efficiencies. Secondary legacy IS also often included fewer stakeholders. It was also concluded that it was easier to abandon outsourced legacy IS compared to proprietary in-house legacy IS. This was due to the fact that when an IS was outsourced, there were fewer personal connections with such legacy IS and no proprietary staff were made redundant. Furthermore, it was concluded that the stakes of various stakeholders in a legacy IS can be high (e.g. management roles, span of control, revenues for suppliers). Abandoning high stakes was particularly high-risk. Legacy IS were occasionally referred to as “friends”. For both staff and suppliers, these systems often formed the basis for their employment. Existing suppliers recommended continuation and new suppliers always promised much better systems. One case organization was even influencing national politics to make sure that politicians would not implement laws that the legacy IS capability could not handle.

Furthermore, it was concluded that there was a difference between abandoning large legacy IS and small legacy IS. Large legacy IS had a higher survival rate compared to smaller legacy IS. Throughout the cases the following arguments were identified:

1. Large legacy IS supported more products or services (e.g. sometimes up to 25 products or services) compared to small legacy IS (e.g. one product or service); consequently, if one product or service supported by the large legacy IS was abandoned there will many left (e.g. 24).
2. Large legacy IS had more users and often more powerful stakeholders, compared to small legacy IS.
3. It was expensive to train more users for a small IS than a few users for a large IS (e.g. in case of a merger).
4. Large legacy IS were in the heart of the organization and more risky to abandon than a small IS, compare it with an open heart surgery.
5. Specific functionality of a small legacy IS was relatively easy to implement in a large IS, compared to implementing large functionality of a large IS into a small IS (e.g. in case of a merger).
6. Large legacy IS could easily handle additional small volumes, while small legacy IS might have had difficulties to handle large amounts of extra volume (scalability); large IS in case of a merger were better scalable than small IS.
7. Large legacy IS were difficult to rebuild and to implement; it takes a lot of time and resources, risks and costs were very high.
8. It was simply not possible to rebuild a large legacy IS due to the fact that years of development (e.g. 5 years) and maintenance (10 years) had been invested in the legacy IS and organizations often no longer had the skills to replace this large legacy IS.
9. The dis-investment of abandoning a large legacy IS was also large.

Consequently, these exploratory results also provided an answer to the research question posed by Swanson and Dans (2000), why larger IS were surviving smaller IS.

In the longitudinal exploratory case the number of actually abandoned applications increased as the A&M programme of Case A continued. The initial plan in 2006 was to abandon 260 applications; however, in 2011, 471 applications had actually been abandoned. These increasing numbers were caused by: new insights into the functionality, technical status and economics of the legacy IS, additional (even routine) knowledge regarding the abandoning process of legacy IS and continuous management support, for instance, through dedicated board level reporting. Gradually, a new situation emerged, resistance to change reduced and staff working on legacy applications either left the organization, or had been relocated to other applications.

Furthermore, 37 good practices related to legacy IS abandonment decision making were identified; 16 were initially identified in the explorative phase of this research and 21 good practices were identified in the validating cases (see Table 32, p. 129), eight were found in both. These good practices represent accumulated and applied knowledge about what was working and not working in different situations and contexts, including lessons learned and the continuing process of learning, feedback, reflection and analysis. The interviewees identified these good practices themselves as repeated experiences in their proprietary organizations. All these findings articulated the importance of methodological support, adequate IS asset management data and higher senior management support. Consequently, this research contributed to theory by providing detailed descriptions of legacy IS abandonment decision making processes, it identified 38 legacy IS abandonment triggers (categorized them internal, external or both) and it identified 37 legacy IS abandonment decision making good practices.

9.4 The practical abandoning process of legacy information systems

The third research question was:

"How do organizations practically abandon legacy IS?"

The final phase in the life cycle of an IS concerns the practical abandoning of legacy IS in organizations. Important conclusions here primarily concerned good practices for the abandoning process, such as the requirement for a comprehensive database with important IS characteristics, the need for a fall back scenario and communication with stakeholders. Furthermore, maintenance in the period preceding practical abandoning should be reduced to a minimum. Practical abandoning could be stressful for stakeholders and organizations (management) should pay attention to the implications for the individuals involved. In the case that a legacy IS was succeeded by a new IS, involvement of legacy IS stakeholders in the new IS was a success factor. In situations where the IS was not succeeded and stakeholders were made redundant, an appropriate exit compensation programme was mentioned as a success factor.

Furthermore, it was concluded that the IS owner (e.g. director, business unit director) should always be held accountable for the practical abandoning of a legacy IS (all cases), although responsibility to realize the practical abandoning might be delegated to a project or programme. It was concluded that when an organization had a consolidation or merger background (Case A, Case B and Case C), a dedicated legacy IS abandoning programme should be preferred for efficiency reasons and because of accumulated knowledge of legacy IS abandoning.

Additional work was required to abandon legacy IS and, for example, switch over to new systems or archive essential data. Often new information was identified during this stage that required a change of plan. The practical abandoning stage, therefore, also required employees that have endurance, like to dig into archival files, like to work with old technology and should have analytical skills. Furthermore, inception of new IS was considered to provide a higher status in the organization compared to abandoning (unless abandoning concerned a major and successful programme with substantial financial impact). In the abandonment phase, documentation was often experienced as inadequate (Case A, Case B and Case C).

Furthermore, 60 good practices related to the practical abandoning of legacy IS were identified; 40 were initially identified in the explorative phase of this research, 20 good practices were identified in the validating cases and 12 occurred in both (see Table 33, p. 132). These good practices represent accumulated and applied knowledge about working methods in different situations and contexts, including lessons learned and the continuing process of learning, feedback, reflection and analysis. The interviewees identified these good practices themselves as repeated experiences in their proprietary organizations. Consequently, this research contributed to the theory on practically abandoning legacy IS by a detailed description of the process and through identifying good practices concerning this process.

9.5 Methodological support for abandoning legacy information systems

The final research objective of this study concerned the design of a method to support the abandonment decision making of legacy IS. This method reflects the most important experiences from the five case studies in this research. The method was based on the design science guidelines (Hevner et al., 2004) and was evaluated by means of informed arguments (see Table 43, p. 145). Design dilemmas for designing a method were conceptualized (see Section 6.5, p. 93) and validated (see Section 7.8, p. 132). It was concluded that a new method should be developed, that this method should be designed at the hierarchical level of the IS, should include financial and non-financial evaluation elements, that a pragmatic number of decision making attributes should be used, that a portfolio approach should be used and that between five and 15 stakeholders (representing different eclectic decision making viewpoints, such as; technical, functional and economical) should be held responsible for legacy IS abandonment decision making. The business manager that can be labeled "IS owner" should always be accountable.

The 10 constituent elements of the method to abandon legacy IS were the following: (1) initialize a centralized IS attributes database (see Section 8.4.1, p. 148); (2) maintain a centralized IS attributes database (see Section 8.4.2, p. 152); (3) score on the legacy IS abandonment triggers of the centralized IS attributes database (see Section 8.4.3, p. 152); (4) evaluate the legacy IS (see Section 8.4.4, p. 152); (5) make the legacy IS abandonment decision (see Section 8.4.5, p. 153); (6) communicate the legacy IS abandonment decision (see Section 8.4.6, p. 153); (7) refine the legacy IS abandonment decision making (see Section 8.4.7, p. 153); (8) materialize the abandoning of legacy IS (see Section 8.4.8, p. 154) and parallel with this: (9) monitor the progress of abandoning of legacy IS (see Section 8.4.9, p. 155); (10) evaluate the legacy IS abandoning (see Section 8.4.10, p. 156). These constituent elements were illustrated in Figure 30 (p. 147).

To the best of our knowledge, the literature on legacy IS abandonment decision making did not include a method to abandon legacy IS. Due to the financial impact of legacy IS to organizations, it was concluded that such a method was, however, highly relevant. By designing a legacy IS abandoning method, this research also meets its research objective, *"to propose a method to abandon legacy IS"*. The method was evaluated by means of informed arguments; however, it has not yet been validated in practice. It is recommended that the method should be tested in further research.

9.6 Primary research question

This research attempted to deepen the understanding of legacy IS and abandonment decision making and to develop methodological support. What does aging of legacy IS imply? Which components of IS age, and how? Why do organizations subsequently decide to abandon IS? What good practices regarding the management of IS, or decision support methods for abandonment decision making could be identified?

The primary research question was:

"How do organizations identify legacy IS and how do they manage abandonment?"

This study chose to research the subsequent life cycle stages of legacy IS, to provide a comprehensive view of the "end of life" of a legacy IS. The available literature was limited and social relevance was high. Subsequently the life cycle stages (1) "*aging process of legacy IS*", (2) "*abandonment decision making process of legacy IS*", (3) "*practical abandoning process of legacy IS*" of legacy IS were researched. Researching these life cycle stages, consequently, provided guidelines to (4) *"propose a method to abandon legacy IS"*.

Although only five organizations were included in this research, these organizations abandoned approximately 2,020 legacy IS. Furthermore, the opportunity was given to interview 51 stakeholders (and some of them were interviewed more than once). Most of these stakeholders had been working in the ICT industry for five to 30

year and served multiple employers. This study, therefore, provides insight into the aging, the abandonment decision making and the practical abandoning of legacy IS.

9.7 External validity

For the external validity of the results in the environments studied, a number of considerations are stated regarding the (1) generalization of results to similar and other organization types and (2) generalizations for other IS life cycle decision making stages. Regarding the generalization of the results to similar and other organizations types, it is noted that this research included private and public organizations, large organizations, between 1,200 and 18,000 employees, with more and less sophisticated information functions. In total 51 interviews were conducted. Typically, the interviewees had been working in the industry between five and 30 years. The first organization (Case A) was a public consolidation of similar branch organizations. Case B was a private consolidation of three former organizations, Case C was a single organization that consolidated IS. All these organizations had dedicated programmes to abandon legacy IS. Case D and Case E abandoned limited numbers of legacy IS in a regular IS life cycle. All the participants in the organization were asked for aging factors of legacy IS, abandonment triggers of legacy IS, and a description of the process of how organizations decided to abandon legacy IS and the process of the actual practical abandoning of legacy IS. Furthermore, good practices were recorded. Most of the aging factors and abandonment triggers of legacy IS were identified in the first longitudinal case and identification of new aging factors and abandonment triggers of legacy IS gradually decreased as cases were added to this research; for example, if Case E had not been added, no legacy IS aging factors would have been missed (see Table 26, p. 118; Table 27, p. 119), and only one abandonment trigger would have been missed (see Table 28, p. 120). It is therefore assumed that the results will not significantly change by adding other large organizations. Small (<100 employees) and midsize organizations (between 100 and 1,000 employees) often have different IT requirements and IT challenges compared to larger organizations. It was not possible to generalize the results to smaller organizations and concerning midsize organizations further research is recommended.

Regarding possible generalizations of findings for other IS life cycle stages, the centralized IS attributes database will also provide valuable information for inception and maintenance decision making. The identified 149 legacy IS aging characteristics will definitely support maintenance (or re-engineering) decisions and can probably be used to prevent the internal aging of IS. Furthermore, a large number of the legacy IS abandonment decision making good practices will also be valuable for inception and maintenance decision making. The three predominant eclectic decision making perspectives (functional, technical and economical) will also be valuable for the inception and maintenance decision making. The portfolio approach will also be valuable in the inception and maintenance phase.

9.8 Directions for further research

During this study the following issues of particular interest were identified for further investigation:

1. Due to a general lack of financial data and in particular historical financial data, it was impossible to make relevant financial estimates of IS aging. Such calculations seem, however, crucial in further explorations of the importance and relevance of individual legacy IS aging factors and legacy IS abandonment triggers. Therefore, further research in this area is suggested.
2. Healthy aging might also be applicable for IS. It is, therefore, advised to further research the internal legacy IS aging factors and their practical usage in the maintenance process.
3. It is concluded that most of the identified legacy IS abandonment triggers were internal and within the control of an organization, suggesting that they also could have been avoided. Further research into the frequency distribution of each individual legacy IS abandonment trigger is suggested.
4. The proposed method for abandoning legacy information systems, was evaluated by means of informed arguments, but not yet tested in practice. It is recommended that the proposed method should be further tested in practice.

5. The aging factors and abandonment triggers of legacy IS are provided at the most detailed level. Further research might categorize them at higher levels.
6. Good practices concerning abandonment decision making of legacy IS and practical abandoning of legacy IS were identified. It is advised to further research whether these good practices are best practices.
7. This research focused on abandonment triggers of legacy IS. One abandonment trigger (complexity) was also mentioned by interviewees as a factor not to abandon a legacy IS and to postpone the legacy IS abandonment decision. An interesting research question based on a continuance perspective instead of a discontinuance perspective to reveal more triggers not to abandon legacy IS could be: "*what are triggers not to abandon legacy IS?"* Further research on factors not to abandon legacy IS might be interesting and is advised.
8. To gain more insight in the decision making made by managers concerning the abandoning of legacy IS, the concept of "stakeholder salience" might be interesting. Mitchell, Agle and Wood (1997) define *"salience"* as, the degree to which managers give priority to competing stakeholder claims. Stakeholder salience is a useful addition to stakeholder theory, and in addition can explain some stakeholder behavior related to the abandoning of legacy IS. Further research in this area is suggested.
9. From the longitudinal research it was concluded that abandoning legacy IS leads to increasing abandonings. Additional insights in the existence and use of legacy IS and decreasing resistance to change presumably causes this effect. Gradually, a new situation is established in the organization, where staff working on older applications has left the organization or started working on other applications. Further research on this phenomenon is recommended.
10. The interviewees, unexpectedly, stated that the financial impact of inception decisions was better known than the abandonment of legacy IS, and legacy IS have been in the organizations for many years. These results hardly correspond with the proposed assumptions, where it was argued that due to the financial history the impact should be known better. An explanation could be that the process of inception was better established in the case organizations and also had a dedicated financial support. Further research in this area is advised.
11. The proposed method primarily depends on Case A, Case B and Case C. These organizations all had IS abandoning processes in place, ample experience, there was active involvement of the board of these organizations. In the other two organizations there was less board involvement. It is uncertain whether the proposed method would also fit smaller organizations or organizations without a consolidation or merger background, e.g. those abandoning only a few legacy IS. This could be subject to further research.
12. In the life cycle approach to IS, theories and supporting methods. Methods and techniques have been developed to optimize the efficient and effective development and maintenance of IS throughout the entire life cycle; from strategic planning, through development, through operations and through to end-of-life decisions (Berghout, 2002). Notable recent publications are:
    - On strategic planning, "Evaluation of information system proposals: design of a decision support method" (Berghout, 1997);
    - On evaluation, "Understanding objectivity in information system evaluation" (Schuurman, 2011);
    - On development, "Product focused software process improvement" (Solingen, 2000);
    - On software release decision, "Design of a methodology to support software release decisions: do the numbers really matter" (Sassenburg, 2005);
    - On IT operations, "The Alignment of Operational ICT" (Klompé, 2003).

    A direction for future research might be to integrate the different theories into one coherent theory covering the entire life cycle (Sassenburg, 2005).

10 REFERENCES

Adolph, W. S. (1996). Cash Cow in the Tar Pit: Reengineering a Legacy System. *IEEE Software - SOFTWARE*, *13*(3), 41-47. doi: 10.1109/52.493019

Alberti, A., & Bertucci, G. (2006). *Innovations in Governance and Public Administration: Replicating what works* (UN-DESA).

Alderson, A., & Shah, H. (1999). Viewpoints on Legacy Systems. *Communications of the ACM, 42*(3), 115-116. doi: 10.1145/295685.295722

Alderson, A., & Shah, H. (1998). Understanding Legacy Systems Through Viewpoints and Events, *8 th annual Business Information Technology (BIT) conference 4/5 th November*, Manchester Metropolitan University.

Andresen, J. L. (2001). A Framework for Selecting an IT Evaluation Method - in the Context of Construction. Danmarks Tekniske Universitet.

Angell, L., & Smithson, S. (1991). *Information Systems Management - Opportunities and Risks*. London: Macmillan Press UK.

Avgerou, C. (2002). *Information Systems and Global Diversity*. Oxford, UK: Oxford University Press.

Avgerou, C., & Cornford, T. (1998). *Developing IS - Concepts, Issues and Practice* (2nd ed.). Basingstoke: Palgrave Macmillan.

Bakehouse, G., & Wakefield, T. (2005). Legacy Information Systems. ACCA internet resource: http://www. accaglobal.com.

Bangia, R. (2010). *Dictionary information Technology*, Laxmi Publications.

Bemmelen, R. W. (1961). The Scientific Character of Geology. *The Journal of Geology*, *69*(4), 453- 463. doi: 10.1086/626761

Bennet, K. (1995). Legacy Systems: Coping with Success. *IEEE Software, 12*(1), 19-23. doi: 10.1109/52.363157

Benson, R. J., Bugnitz, T. L., & Walton, W. B. (2004). *From Business Strategy to IT action*. New York, NY: John Wiley and Sons, Inc.

Berghout, E. W. (1997). *Evaluation of IS proposals: design of a decision support method*. Doctoral dissertation, Delft University of Technology.

Berghout, E. W. (2002). IT Economics Research Program. Report, Business & ICT Section, Faculty of Economics, University of Groningen, 2002.

Berghout, E. W., & Nijland, M. (2002). Full life cycle management and the IT management paradox. In D. Remenyi & A. Brown, *The make or break issues in IT management: a guide to 21st century effectiveness* (pp. 77-107). Oxford: Butterworth-Heinemann.

Berghout, E. W., Nijland, M., & Powell, P. (2011). Management of life cycle costs and benefits: Lessons from information systems practice. *Computers in Industry, 62*(7), 755-764. doi: 10.1016/j.compind.2011.05.005

Bisbal, J., Lawles, D., Wu, B., & Crimson, J. (1999). Legacy IS: issues and directions. *IEEE Software*, *16*(5), 103-111. doi: 10.1109/52.795108

Blanchard, B., & Fabrycky, W. (1990). *Systems engineering and analysis.* Pearson Education Limited.

Boehm, B. W. (1981). *Software Engineering Economics*. Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Booth, M., & Philip, G. (2005). Information systems management: role of planning, alignment and Leadership. *Behaviour & Information Technology, 24*(5), 391-404. https://doi.org/10.1080/01449290512331335582

Boswell, J. (1791). The life of Samuel Johnson, LL.D. Printed by Henry Baldwin, for Charles dilly, in the poultry. London.

Braaksma, J., Commandeur, A., & Berghout, E. W. (2006). The Business Case for (ICT) Investment Evaluation in Nonprofit Organisations. *Proceedings of the 13th European Conference on Information Technology Evaluation* 28-29 September, pp. 91-102.

Brealey, R., & Myers, S. (1991). *Principles of corporate finance*. New York: McGraw-Hill Inc.

Brodie, M., & Stonebraker, M. (1995). *Migrating Legacy systems*. San Francisco: Morgan Kaufman publishers, Inc.

Brooke, C. (2002). A framework for evaluating Legacy IS. In D. Remenyi & A. Brown, *The make or break issues in IT management: a guide to 21st century effectiveness* (pp. 108-124). Oxford: Butterworth-Heinemann.

Brooke, C. (1994). Information Technology and the Quality Gap. *Employ relations, 16* (4), 22-34. doi: 10.1108/01425459410066265

Brooke, C., & Ramage, M. (2001). Organizational scenarios and Legacy systems. *International Journal of Information Management. 21*(5), 365-384. doi: 10.1016/S0268-4012(01)00023-8 Brussaard, B., & Tas, P. (1980). Information and organization policies in public administration. In S. Lavington (ed.), *Information Processing 80: IFIP congress* (pp. 821-826). Tokyo, Melbourne and Amsterdam: North-Holland Publishing Co.

Campbell-Kelly, M., & Williams, M. R. (1985). *The Moore School Lectures: Theory and Techniques for Design of Electronic Digital Computers*. Charles Babbage Institute Reprint. MIT and Tomash Publishers.

Capurro, R. (2009). Past, present, and future of the concept of information. *Journal for a Global Sustainable Information Society 7*(2), 125–141.

Capurro, R., & Hjorland, B. (2003). The concept of information Annual Review of Information Science and Technology. v37. *Annual Review of Information Science and Technology (ARIST)*, pp. 343-411, 2003. Ed. Blaise Cronin, Information Today, Inc. Medford, NJ (2003).

Chaffey, D., & Wood, S. (2005). *Business Information Management, Improving Performance Using Information Systems*. Harlow, Essex: Pearson Education Limited.

Chan, T., Chung, S. L., & Ho, T. H. (1996). An Economic Model to Estimate Software Rewriting and Replacement Times. *IEEE Transactions on Software Engineering*, *22*(8), 580-599. doi: 10.1109/32.536958

Chua, W. F. (1986). Radical developments in accounting thought. *The Accounting Review, 61*(4), 601-632.

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM, 13*(6), 377-387. doi: 10.1145/362384.362685

Cohen, L., Manion, L. & Morisson, K. (2007). *Research Methods in Education* (6th ed.). New York, NY: Routledge.

Comella-Dorda, S., Wallnau, K., Seacord, R., & Robert, J. (2000). A survey of Legacy system modernization Approaches. In: *CMU/SEI-00-TN-003*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA, April 2000.

Computerworld 8th April 2013, Should universities offer Cobol classes? https://www.computerworld.com/article/2496360/it-careers/should-universities-offer-cobol-classes-.html

Cornford, T., & Smithson, S. (2006). *Project Research in Information Systems - A Student's Guide* (2nd ed.). London, UK: Palgrave.

Daga, A., Cesare, D., Lycett, M., & Partridge, C. (2005). An ontological approach for recovering Legacy business content. *Proceedings of the 38th Hawaii international conference on system sciences 2005.*

Davenport, T. H., & Markus, M. L. (1999). Rigor vs. Relevance Revisited: Response to Benbasat and Zmud. *MIS Quarterly*, *23*(1), 19-24. doi: 10.2307/249405

Davis, G. B., & Olson, M. H. (1985). *Management Information Systems: Conceptual Foundations, Structure and Development* (2nd ed.). New York, NY: McGraw-Hill Book Company.

Decker, J. (2008). *How the CFO should approach the financial application portfolio*. Gartner Research.

Fisher, M. H. (1945). *Encore: A Continuing Anthology.* As quoted by Smith, D.

Furneaux, B., & Wade, M. (2011). An Exploration of Organizational Level Information Systems Discontinuance Intentions. *MIS Quarterly, 35*(3), 573-598. doi: 10.2307/23042797

Furneaux, B., & Wade, M. (2010). The end of the information system life: A model of IS discontinuance. *The DATABASE for Advances in Information Systems, 41*(2), 45-69. doi: 10.1145/1795377.1795381

Furneaux, B. (2009). *Time for a change: an exploration of information systems discontinuance*. Doctoral dissertation, may 2009, schulich school of business York university, Toronto, Ontario.

Gode, D., Barua, A., & Mukhopadhyay, T. (1990). On the economics of the software replacement problem. *Proceedings of the 11th International Conference of Information Systems (Copenhagen, Denmark)*, pp. 159–170.

Golafshani, N. (2003). Understanding Reliability and Validity in Qualitative Research. The Qualitative Report, 8(4), 597-606. Retrieved from http://nsuworks.nova.edu/tqr/vol8/iss4/6

Gold, N. (1998). The meaning of “legacy systems”. Technical Report 7/98, Dept. of Computer Science, University of Durham, UK.

Gorard, S. (2001). *Quantitative Methods in Educational Research: The Role of Numbers Made Easy*. London: Continuum.

Gottling, C., & Torgnysdotter, L. (2002). *Application portfolio management.* Master thesis Department of Informatics School of Economics and Commercial Law Göteborg University.

Government Accountability Office, GAO (2015). *High risk series an update*. GAO-15-290.

Grance, T., Hash, J., & Stevens, M. (2004). *Security Considerations In The Information System Development Life cycle. Recommendations Of The National Institute Of Standards And Technology (NIST).* Technology Administration, U.S. Department of Commerce, Computer Security Division, Information Technology Laboratory. SP 800-64.

Guba, E. G., & Lincoln, Y. S. (1994). Competing paradigms in qualitative research. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (pp. 105-117). Thousand Oaks, California: SAGE Publications Inc.

Halma, F. (1710). *WOORDENBOEK DER NEDERDUITSE EN FRANSCHE TAALEN* (tweede deel).

Harrison, F. (1999). *The managerial decision making process* (5th ed.). Boston, Massachusetts: Houghton Mifflin company.

Henderson, B. D. (1970). *The Product Portfolio*. BCG Perspectives, Boston Consultancy Group, Boston.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28*(1), 75-105.

Hirschheim, R., & Klein, H. (1989). Four paradigms of information systems development. *Communications of the ACM 32*(10), 1199-1216. doi: 10.1145/67933.67937

Howard, P. (2011). *FISMA Principles and Best Practices: Beyond Compliance*. CRC press.

Hughes, J. (1990). *The Philosophy of Social Research* (2nd ed.). Harlow, England: Addison-Wesley Longman, Inc.

Hunter, R., & Aron, D. (2006). *High Value, High Risk: Managing the Legacy portfolio*. Gartner EXP Premier.

Hurston, Z. N. (1942). Dust Tracks on a Road. Philadelphia: J. B. Lippincott, 1942. Reprinted: Chicago: University of Illinois Press, 1970.

Information. (n.d.). In *OED online,* Oxford University Press. Retrieved January 18, 2018, from http://www.oed.com/view/Entry/95568?redirectedFrom=information+system

Internal company report, April 2006.

ISO. ISO/IEC 25010:2011.

ISO. ISO/IEC 9126-1:2001 Software Engineering - Product Quality - Part 1: Quality model. International Organization for Standardization, 2001b.

ISO 9000:2005, Procedure Clause 3.4.5,

http://www.iso.org/iso/iso_catalogue/management_standards/quality_management/iso_9001_2008/guidance_on_the_documentation_requirements_of_iso_9001_2008.htm

Jacka, M. J., & Keller, P. J. (2009). *Business process mapping; improving customer satisfaction* (2nd ed.). Hoboken, NJ: John Willey & Sons, Inc.

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Software Development Process*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Jevremovic, T. (2005). *Nuclear Principles in Engineering*. Springer US.

Kanellis, P., & Papadopoulos, T. (2009). Conducting Research in Information Systems: An Epistemological Journey. In: Information Systems Research Methods, Epistemology and Applications. Information science reference.

Keaveney, S. M. (1995). Customer switching behaviour in service industries: an exploratory study. *Journal of Marketing*, *59*(2), 71-82. doi: 10.2307/1252074

Kelly, S., Gibson, N., Holland, C. & Light, B. (1999). Focus issue on legacy information systems and business process engineering: a business perspective of legacy information systems. *Communication of the Association for Information Systems, 2*(7). Retrieved from https://dl.acm.org/citation.cfm?id=374164

Kitchenham, B. (2004). Procedures for Performing Systematic Reviews. Software Engineering Group; National ICT Australia Ltd., Keele; Eversleigh, Technical report Keele University Technical Report TR/SE-0401.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3, EBSE Technical Report EBSE-2007-01, Keele University and University of Durham.

Kitchenham, B., Brereton, P. O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology, 51*(1), 7–15. https://doi.org/10.1016/j.infsof.2008.09.009

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering – A tertiary study. *Information and Software Technology, 52*(8), 792-805. doi: 10.1016/j.infsof.2010.03.006

Klompé, R. H. (2003). The alignment of operational ICT. Dissertation Delft University of Technology, Delft, Eburon Academic Publishers.

Krishnan, M. S., Mukhopadhyay, T., & Kriebel, C. H. (2004). A Decision Model for Software Maintenance. *Information Systems Research, 15*(4), 396-412. Retrieved from: https://pubsonline.informs.org/doi/abs/10.1287/isre.1040.0037?journalCode=isre

Kyte, A. (2008). When and how to upgrade applications. (Gartner).

Land, F., & Kennedy-McGregor, M. (1987). Information and IS: Concepts and perspectives in Information analysis. Selected readings by Robert Galliers, Addison Wesley (First published in the proceedings of Infotech State of the Art Conference, October, 1980, Infotech Limited, Maidenhead.)

Larsen, T. (1998). Information System innovation: A framework for research and practice. In *information systems innovation and diffusion* (pp. 411-434). Idea group publishing

Laudon, K., & Laudon, J. (2001). *Essentials of management information systems*. Prentice-Hall Inc., New Jersey.

Lawrence, C. (2007). Adapting Legacy systems for SOA. IBM publication. http://www.ibm.com/developerworks/webservices/library/ws-soa-adaptleg/

Lee, A. (1991). Integrating positivist and interpretivist approaches to organizational research. *Organization Science, 2*(4), 342–365. Retrieved from http://www.jstor.org/stable/2635169

Lehman, M. (1980a). On understanding laws, evolution and conservation in the large-program life cycle. *Journal of Systems and Software, 1*, 213-221. doi: 10.1016/0164-1212(79)90022-0

Lehman, M. (1980b). Program life cycles and laws of software evolution. *Proceedings of the IEEEE*, vol 68, no 9 September 1980.

Lehman, M., Kahen, G., & Ramil, J. F. (2000). Replacement Decisions for E-type Software - Some Elements. *ICSE'2000 Second Workshop on Economics-Driven Software Engineering Research*, Limerick, Ireland, June 2000.

Lehman, M., & Ramil, F. (2001). Rules and Tools for Software Evolution Planning and Management. *Annals of Software Engineering, 11*(1), 15-44. https://doi.org/10.1023/A:1012535017876

Lehman, M. (1996). Laws of Software Evolution Revisited. EWSPT96.

Lehner, F. (1989). The software life cycle in computer applications. *Long Range Planning, 22*(5), 38-50. https://doi.org/10.1016/0024-6301(89)90167-2

Lerner, M. P. (2005). THE ORIGIN AND MEANING OF “WORLD SYSTEM”. *Journal of the History of Astronomy, 36*(4), 407-441.

Liddell, H. G. (1940). Robert Scott. A Greek-English Lexicon. revised and augmented throughout by Sir Henry Stuart Jones with the assistance of Roderick McKenzie Oxford. Clarendon Press. 1940.

Light, B. (2003). An Alternative Theory of Legacy Information Systems. In: C. Ciborra, R. Mercurio, M. De Marco, M. Martinez & A. Carignani (Eds), *Proceedings of the 11th European Conference on IS Universita di Napoli Federico II: Naples*, CD R

Lindblom, C. E. (1959). The Science of "Muddling Through". Public Administration Review*, 19*(2), 79-88. doi: 10.2307/973677

Long, J. (2006). Modernizing Legacy applications. BPM.COM http://www.bpm.com/featureprint.asp?featureId=212

Looijen, M. (1995). *Beheer van informatie systemen*. Deventer: Kluwer bedrijfswetenschappen.

Maizlish, B. & Handler, R. (2005). *IT Portfolio Management step by step*. Hoboken, New Jersey: John Willey & Sons, Inc.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance, 7*(1), 77-91. doi: 10.1111/j.1540-6261.1952.tb01525.x

Martin, J. (1991). *Rapid Application Development*. New York, NY: MacMillan Publishing Co.

Mason, J. (2002). *Qualitative Researching* (2nd ed.). Thousand Oaks, California: SAGE Publications Inc.

McFarlan, F. W. (1981). Portfolio approach to information systems. *Harvard Business Review, 59*(5), 142-150.

Mingers, J. (2001). Combining IS Research Methods: Towards a Pluralist Methodology. *Information Systems Research*, *12*(3), 240-259. doi: 10.1287/isre.12.3.240.9709

Mitchell, R., Agle, B., & Wood, D. (1997). Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts. *The Academy of Management Review, 22*(4), 853-886. Retrieved from http://www.jstor.org/stable/259247

O'Brien, J., & Marakas, G. (2009). *Management Information Systems*. New York, NY: McGraw-Hill/Irwin.

OCG (2007). *ITIL*. London TSO.

OCG (1996). (the Office of Government Commerce) PRINCE (Project in controlled environment) http://www.ogc.gov.uk/index.asp

Orlikowski, W., & Iacono, S. (2001). Research commentary: desperately seeking the IT in IT research – A call to theorizing the IT Artifact. *Information System Research, 12*(2), 121-134. Retrieved from http://www.jstor.org/stable/23011075

Oxford Dictionary of Business and Management (6th ed.). (2016). Oxford, UK: Oxford University Press.

Oxford Dictionary of English (3rd ed.). (2010). Oxford, UK: Oxford University Press.

Parnas, D. (1994). Software Aging. *IEEE.*

Pettigrew, A.W. (1985). Contextualist research and the study of organizational change processes, Chapter 4 in Research Methods in Information Systems, E. Mumford et al. (Eds.), Amsterdam: North Holland.

Pirsig, R. M. (1974). *Zen and the Art of Motorcycle Maintenance: An Inquiry into Values*. William Morrow edition.

Pols, R., & Backer, Y. (2007). *BISL - A management guide*. Zaltbommel: Van Haren Publishing.

Pols, R., & Backer, Y. (2006). *ASL - Application service library: A management guide*. Zaltbommel: Van Haren Publishing.

Remenyi, D. (1999). *IT investment making a business case*. Oxford: Butterworth-Heinemann.

Remenyi, D., Williams, B., Money, A., & Swartz, E. (2005). *Doing research in business and management*. London: SAGE publications Ltd.

Renkema, T. J. W., & Berghout, E. W. (1999). *Investeringsbeoordeling van IT-projecten: een methodische benadering*. Deventer: Kluwer Bedrijfsinformatie.

Renkema, T. J. W., & Berghout, E. W. (1997). Methodologies for information systems evaluation at the proposal stage: a comparative view. *Information and Software Technology, 39*(1), 1-13. https://doi.org/10.1016/0950-5849(96)85006-3

Richmond, W. B., Nelson, P., & Misra, S. (2006). An Empirical Analysis of Software Life Spans to Determine the Planning Horizon for New Software*. Information Technology and Management, 7*(2), 131-49. https://doi.org/10.1007/s10799-006-8104-8

Rittinghouse, J. (2004). *Managing Software Deliverables; A Software Development Management Methodology*. Oxford, UK: Elsevier Digital press.

Robey, D. (1996). Research Commentary: Diversity in Information Systems Research: Threat, Promise, and Responsibility. *Information Systems Research, 7*(4), 400-408. https://doi.org/10.1287/isre.7.4.400

Royce, W. (1970). *Managing the Development of Large Software Systems, Concepts and Techniques.* Proceedings, IEEE Wescon, August 1970, pages 1-9.

Sabbagh, S. (2008). Application Life-Cycle Management. Master thesis university of Groningen (http://irs.ub.rug.nl/dbi/4935402d654e3)

Sakthivel, S. (1994). A decision model to choose between software maintenance and software redevelopment. *Journal of Software Maintenance: Research and Practice, 6*(3), 121-143. doi: 10.1002/smr.4360060303

Saldana, J. (2009). *The Coding Manual for Qualitative Researchers.* Thousand Oaks, California: SAGE Publications Inc.

Sassenburg, J. (2005). *Design of a methodology to support software release decisions: do the numbers really matte*r. PhD dissertation University of Groningen.

Saunders, M., Lewis, P., & Thornhill, A. (2007). *Research Methods for Business Students* (4th ed.). Harlow, England: Pearson Education Limited.

Scardino, L., Parameswaran, S. K., Young, A., & Buttorff Sikes, S. (2005). *Use this five-step process to analyse applications before deciding to outsource*. Gartner Research.

Schuurman, P. M. (2011). *Understanding objectivity in information system evaluation. Perceptions of information system economics*. Doctoral dissertation, University of Groningen. Gildeprint Drukkerijen.

Seacord, R., Plakosh, D., & Lewis, G. (2003). *Modernizing Legacy Systems: Software Technologies, Engineering Processes and Business Practices*. Boston, MA: Pearson Education, Inc.

Sellars, A. (2004). *Life cycle extension strategies for Legacy systems*, Master thesis, the Graduate School of Vanderbilt University.

Simon, H. A. (1996). *The Sciences of the Artificial*. (3rd ed.). Cambridge, MA: MIT Press.

Sommerville, I. (2007). *Software engineering 8th edition*. Harlow, England: Pearson Education Limited.

Swanson, E. B., & Dans, E. (2000). System Life Cycle Expectancy and the Maintenance Effort: Exploring Their Equilibrium. *MIS Quarterly, 24*(2), 277-297. doi: 10.2307/3250939

Swinkels, F. G. J. P. (1997). Managing the Life Cycle of Information and Communication Technology Investments for Added Value. *European Conference on the Evaluation of Information Technology*, Delft University Press, Delft, the Netherlands.

System. (n.d.). In *OED online,* Oxford University Press. Retrieved January 18, 2018, from http://www.oed.com/view/Entry/196665?redirectedFrom=system

Tashakkori, A., & Teddlie, C. (1998). *Mixed methodology: Combining Qualitative and Quantitative Approaches.* Thousand Oaks, California: SAGE Publications Inc.

Thesaurus Linguae Latinae [Thesaurus of the Latin language]. 1900. Leipzig, Germany: Teubner.

Tilley, S. R., & Smith, D. (1995). *Perspectives on Legacy System Reengineering.* Reengineering Center, Software Engineering Institute, Carnegie Mellon University.

Tornatzky, L., & Fleischer, M. (1990). *The process of technology innovation*. Lexington, MA: Lexington Books.

Ulrich, W. M. (2002). *Legacy systems: Transformation Strategies*. Upper Saddle River, New Jersey: Pearson Education Inc.

Ulrich, W. M., & Newcomb, P. (2010). *Information Systems Transformation, Architecture-Driven Modernization Case Studies*. Burlington, MA: Morgan Kaufmann Publishers.

Valacich, J., & Schneider, C. (2010). *Information Systems Today: Managing in the Digital World* (4th ed.). Upper Saddle River, New Jersey: Pearson Education Inc.

Van den Heuvel, W. (2007). *Aligning modern business processes and legacy systems: a component-based perspective.* Cambridge, MA: MIT Press.

Van Engelen, J. M. L., & Van der Zwaan, A. H. (1994). Bedrijfskundige Methodologie 2; een Technisch-Methodologische Context. *Bedrijfskunde*, *66*(2), 85-94.

Van Solingen, R. (2000). *Product focused software process improvement.* Doctoral dissertation, Eindhoven University of Technology.

Ward, J. M. (1988). Information Systems and Technology Application Portfolio Management - an Assessment of Matrix-Based Analyses. *Journal of Information Technology, 3*(3), 205-215. https://doi.org/10.1057/jit.1988.36

Warren, I. (1999). *The Renaissance of Legacy Systems*. London: Springer Pub.

White, J. D. (2007). *Managing Information in the Public Sector.* New York, NY: Routledge.

Winter, R., Baskerville, R., Frank, U., Heinzl, A., Hevner, A. R., & Venable, J. R. (2007). Relevance and Rigour - What are Acceptable Standards and How are they Influenced? *Wirtschaftsinformatik, 49*(5), 403-409. https://doi.org/10.1007/s11576-007-0087-5

Wu, B., Lawless, D. Bisbal, J., Grimson, J., Wade, V., O'Sullivan, D., & Richardson, R. (1997). Legacy systems migration a method and its toolkit framework. *Proceedings of the Joint 1997 Asia Pacific Software Engineering Conference and International Computer Science Conference (APSEC\'97/ ICSC\'97)*.

Yin, R. K. (2003). *CASE STUDY RESEARCH: Design and Methods* (3rd ed.). Thousand Oaks, California: SAGE Publications Inc.

Zvegintzov, N. (1984). Immortal Software. *Datamation, 30,* 170-180.

APPENDIX A DESIGN OF EXPLORATIVE RESEARCH

In this appendix in the first section the explorative questionnaire design is presented and in the second section the analysis model is presented.

Appendix A.1 Explorative questionnaire design

In Table 48, the explorative questionnaire design is presented; to ensure essential data are collected, a data requirements table is created (Saunders et al., 2007). The data requirements table includes the actual questionnaire as used in the interviews (Columns: "Question No."; "Investigative questions" and "Variables required") and additional design information concerning the questionnaire (Column: "Detail in which data measured").

<table>
<tr><td colspan="4">Understanding Legacy information systems (IS) and abandonment decision making. Towards methodological support.</td></tr>
<tr><td colspan="4">Type of research: open[16] in-depth and semi-structured.</td></tr>
<tr><td colspan="3">Explorative questionnaire.</td><td>Additional design information concerning the explorative questionnaire.</td></tr>
<tr><td colspan="3">Name interviewee: Function:
Role: Organization:
Date: Telephone:
Email:</td><td></td></tr>
<tr><td>Question No.</td><td>Investigative questions.</td><td>Variable(s) required.</td><td>Detail in which data measured[17].</td></tr>
<tr><td>Q1</td><td>Can the interviewee draw an organization chart (or is it already available) with the position of the interviewee, and can the interviewee explain his role?</td><td>Drawing chart or documentation.</td><td>A chart or documentation, to provide detailed insights into the position and role of the interviewee in the organization expressed by a global organization chart. (It will also show interest to the interviewee in his job which may reveal more information).</td></tr>
<tr><td colspan="4">Introduction questions concerning IS and legacy IS.
Due to the fact that these are very tough questions, which might upset the interviewee, researcher should carefully explain the purpose of these questions. The purpose is: "We want to come to a definition that you consider to be a legacy IS, therefore we have to go to your considerations what you consider to be an IS and a legacy IS".</td></tr>
<tr><td>Q2</td><td>What do you consider to be an IS?</td><td>Opinion about the artifact legacy IS.</td><td>A description of the artifact IS, to understand the definition of an IS of the interviewee and to investigate if different groups of stakeholders have different IS perspectives.</td></tr>
<tr><td>Q3</td><td>What characteristics does an IS have?</td><td>Technological, functional, economical or other IS characteristics.</td><td>A description of (technological, functional, economical or other) IS characteristics, to understand which variables describe characteristics of an IS.</td></tr>
<tr><td>Q4</td><td>Can you tell me your impression of the performance of your IS?</td><td>Opinion of the performance of the IS.</td><td>An opinion of the performance of the IS, to gain more information of the current performance of IS and also to get more information of IS characteristics.</td></tr>
<tr><td>Q5</td><td>Are your IS ready for the future? (are they flexible and can they deal with changes within the business processes)</td><td>Opinion of the flexibility of the IS.</td><td>A description of the flexibility of IS within the business, to gain more information about the future performance of IS and also get more information about required IS characteristics.</td></tr>
<tr><td>Q6</td><td>How would you translate the terminology legacy IS to the Dutch language?</td><td>Translation</td><td>Translation of the terminology: "legacy IS" to the Dutch language, to find a Dutch definition of a legacy IS. Legacy IS is difficult to translate from English to Dutch.</td></tr>
</table>

[16] Although this questionnaire seems structured, the interviews were open. This exploratory research is deemed to be inductive. It is always possible to skip questions and to ask other questions, it is just intended as guide for the interviewer.
[17] The original data requirements table as suggested by Saunders *et al*. (2007) is enhanced with the: "why question", because this gives the reason why this question is in the questionnaire.

Q7	What do you consider to be a legacy IS?	Opinion about the artifact legacy IS.	A description of the artifact legacy IS, to further explore the concepts of legacy IS and to investigate if different groups of stakeholders have different legacy IS perspectives.
Q8	What characteristics does a legacy IS have?	Technological, functional, economical or other legacy IS characteristics.	A description of (technological, functional, economical or other) legacy IS characteristics, to understand which variables are describing legacy IS characteristics and to categorize these characteristics.
Q9	Please describe the life cycle of an IS from inception till abandoning.	Opinion concerning the life cycle of an IS.	A detailed description of the aging process, to gain information concerning the aging of an IS during its life.
Q10	What are your personal triggers to decide when to abandon legacy IS?	Technological, functional, economical or other personal triggers.	A description of the (technological, functional, economical or other) abandonment triggers, to gain information concerning the personal abandonment triggers of the interviewee; these can be further categorized into technological, functional, economical or other. Finally these abandonment triggers can be used to analyze if there is a relation between the role of the interviewee and the personal abandonment trigger.
Q11	What are the triggers for the organization when to abandon legacy IS?	Technological, functional, economical or other organizational triggers.	A description of the (technological, functional, economical or other) organization abandonment triggers. To gain information concerning the organization abandonment triggers; these can be further categorized into technological, functional, economical or other. To understand whether organizational abandonment triggers are different compared to the personal abandonment triggers.
Q12	What do you consider to be an application?	Opinion concerning the artifact application.	A description of an application, to have the same reference concerning applications between interviewer and interviewee during the rest of this interview.
Q13	Do you think that IS in your organization are well-documented? (Y/N) Describe what your definition is of well-documented IS.	Technological (□ Y □ N) Functional (□ Y □ N) Economical (□ Y □ N) Other (□ Y □ N). Opinion	A description of the state of documentation of IS (e.g. technological, functional, economical), to understand the state of documentation. Opinion of the interviewee concerning the definition of well-documented IS.
Q14	Do you think that legacy IS in your organization are well-documented? (Y/N) Describe what your definition is of well-documented legacy IS.	Technological (□ Y □ N) Functional (□ Y □ N) Economical (□ Y □ N) Other (□ Y □ N). Opinion	A description of the state of documentation of legacy IS (e.g. technological, functional, economical), to understand the state of documentation. Opinion of the interviewee concerning the definition of well-documented IS.
Q15	How is the abandonment decision making process of a legacy IS? (Or describe a practical example).	Process description.	A description of the legacy IS abandonment decision making process, to gain information concerning the abandonment decision making process concerning legacy IS.
Q16	Is this legacy IS abandonment decision making process described in the organization?	□ No □ Yes, describe the process.	To know if this legacy IS abandonment decision making process is described in the organization, and to understand this process.
Concerning the following questions, it is good to have full information on all the abandoned legacy IS decisions. However, not all interviewees will have knowledge of all these IS or will be able to answer all questions. In that case the interviewee can use a recent situation of an abandoned legacy IS and can use that as input for the questions being asked. The interviewee should be clear that the answer to a question is based on a recent case and not on the full IS portfolio. At this stage it is important to know what was typical and what the exceptions to the rule were.			
Q17	Are alternatives for abandoning legacy IS evaluated? (If Yes, describe the process)	□ No □ Yes, describe the process.	If alternatives are evaluated, understanding of this process, to gain insights into the maturity of the legacy IS abandonment decision making process.
Q18	Who are stakeholders in legacy IS?	Names, functions and roles.	Names, functions and roles, to understand who the stakeholders are, to check if important stakeholders to be interviewed are missing. To understand the relation between the stakeholder and the impact of the legacy IS abandonment decision.
Q19	Which stakeholders are involved in the abandonment decision making of legacy IS and what is their role in the abandonment decision making?	Names, functions and roles.	Names, functions and roles, to understand which stakeholders are involved and to check if important stakeholders are missing in the legacy IS abandonment decision making. To inquire the role of stakeholders in the abandonment decision making of legacy IS. To understand the relation between the stakeholder and the impact of the abandonment decision.

Q20	Do you consider abandonment decisions concerning legacy IS to be successful. How do you define successful?	□ Not successful? □ Moderate successful? □ Very successful? Opinion concerning the definition of successful.	To understand if legacy IS abandonment decisions are: "Not", "Moderate" or "Very" successful. To understand how interviewee define successful legacy IS abandonment decision making.
Q21	What are success criteria and what are failures in the process of legacy IS abandonment decision making?	Do: success factors. Don't: failures.	Description of do's and don'ts, to collect learning points or "good practices" for a method to abandon legacy IS.
Q22	Who is accountable/responsible for the legacy IS abandonment decision making?	Names, function and roles.	Names, functions and roles of stakeholders, to understand which stakeholders are accountable/responsible for the legacy IS abandonment decision making.
Q23	How often is the legacy IS abandonment decision making process performed during the year?	Continuous process or intermittent process (No./year).	Description and No. of times per year, to know if the legacy IS abandonment decision making process is a continuous process or a process that is performed a few times each year.
Q24	Can the interviewee draw the process of practical abandoning legacy IS?	Drawing chart or documentation.	A chart or documentation, to gain detailed insight into the processes related to the practical abandoning of legacy IS.
Q25	Who is accountable/responsible for practically abandoning legacy IS?	Names, function and roles.	Names, functions and roles, to know who is accountable/responsible for the practical abandoning.
Q26	What are the problems of practically abandoning legacy IS?	Description of different problems.	Description of problems, to understand the problems of practically abandoning legacy IS.
Q27	How are these problems handled?	Description of handling these problems.	Description, to understand how problems are handled.
Q28	Is the impact of abandoned legacy IS on other IS known in the organization?	Description of impact amongst IS.	Description of impact between IS, used to understand relations between IS.
Q29	How many IS are there in your organization (including legacy IS) and what are the costs?	Numbers of IS Y,+2,Y+1,Y,Y-1,Y-2 No. of IS / No. of applications.	Numbers of IS, to know how many IS are yearly involved.
Q30	How many legacy IS are abandoned?	Numbers of IS Y,+2,Y+1,Y,Y-1,Y-2 No. of IS / No. of applications.	Numbers of legacy IS, to know how many legacy IS are yearly abandoned.
Q31	What is the impact of abandoning legacy IS on applications?	Description of the impact of abandoning legacy IS on applications.	Described impact, to gain information concerning the impact of abandoning legacy IS on applications and to know the number of applications.
Q32	What is the impact of abandoning legacy IS on hardware?	Description of the impact of abandoning legacy IS on hardware and the amount of hardware.	Described impact, to gain information concerning the impact of abandoning legacy IS on hardware and to know the amount of hardware.
Q33	What is the impact of abandoning legacy IS on people?	Description of the impact of abandoning legacy IS on people.	Described impact, to gain information concerning the impact of abandoning legacy IS on the human factor.
Q34	What is the impact of abandoning legacy IS on data?	Description of the impact of abandoning legacy IS on data.	Described impact, to gain information concerning the impact of abandoning legacy IS on the data.
Q35	What is the impact of abandoning legacy IS on processes?	Description of the impact of abandoning legacy IS on processes.	Described impact, to gain information concerning the impact of abandoning legacy IS on business processes.
Q36	What is the financial impact of the abandoned legacy IS? Impact is defined as operational IT costs, impact on the business process. No. people, etc.	No. Euro's, as part of the whole ICT cost.	Detailed cost information (e.g. total IT spend, share legacy systems, impact on the business), to gain information concerning the impact of abandoning legacy IS on the total spend.
Q37	What are the costs of abandoning legacy IS?	No. Euros.	Detailed cost information (project costs, supplier costs, cost of mirroring, cost of practical disposal), to gain information concerning the total costs of abandoning legacy IS.

Q38	Can you describe a critical incident[18], how were you involved, how did you solve this problem?	Critical incidents, involvement of interviewee and solution of the critical incident.	Detailed description, to know about things which can go wrong and how they are solved.
Q39	Is there a communication process of abandoning legacy IS; if so, what is the role of communication in this abandoning process?	Communication process. □ No □ Yes, describe the process.	Information or examples of the communication process and content concerning abandoning legacy IS, to understand the role of communication in abandoning a legacy IS, to understand the relation between successful abandoning and communication.
Q40	Are you missing relevant questions?	□ No □ Yes, provide additional questions.	To add more relevant questions, if necessary.
Q41	Do you have suggestions for improving this questionnaire?	□ No □ Yes, describe the improvements.	Add new suggestions. To improve the questionnaire.
Q42	Do you have suggestions for other internal or external stakeholders to be interviewed?	□ No □ Yes, provide names of stakeholders.	To know if important stakeholders (name function, email phone) are missing in this research.
Q43	Can I reference you for approaching these stakeholders?	□ No □ Yes	To gain access to more stakeholders it is important to reference.
Q44	Do you like to have feedback of the total study?	□ No □ Yes	To give interviewees feedback of the results of the research.
Q45	How many years work experience within the organization?	No. years.	To understand the knowledge of the organization; assumption is that more years' experience is more knowledge.
Q46	How many years work experience in this field?	No. years.	To understand the knowledge of the interviewee; assumption is that more years' experience is more knowledge.

Table 48. Design of the questionnaire

[18] Use of critical incident technique Saunders *et al*. (2007). Ask respondents to describe in detail a critical incident or a number of incidents that are key to the research question. A critical incident is defined as an activity or event where the consequences were so clear that the respondent has a definite idea regarding the effects (Keaveney, 1995).

Appendix A.2 Analysis model explorative research

Table 49 illustrates the analysis model within the exploration phase. The analysis model provides the relationship between investigative life cycle processes, the appropriate research questions, sub-research questions (SRQ) and interview questions (Q).

Understanding Legacy information systems (IS) and abandonment decision making. Towards methodological support.			
Primary research question: How do organizations identify legacy IS and how do they manage abandonment?			
Start	Can you draw an organization chart?		Question number Q1
Life cycle stage.	Research questions.	Sub-research questions (SRQ).	Question numbers Q(n)
1) The aging process of legacy IS. .	1) How do legacy IS age?	SRQ 1: How are legacy IS defined in practice? SRQ 2: What are specific characteristics of legacy IS? SRQ 3: Which perspectives can be identified regarding the evaluation of legacy IS? SRQ 4: How are legacy IS evaluated in practice? SRQ 5: How can the identification of legacy IS be efficiently organized?	Q2-Q9
2) The abandonment decision making process of legacy IS.	2) How do organizations decide to abandon legacy IS?	SRQ 6: How is the decision making process to abandon legacy IS organized in practice? SRQ 7: What are the triggers to abandon a legacy IS? SRQ 8: Which stakeholders participate in abandonment decision making of legacy IS? SRQ 9: Who is held responsible for the abandonment decision of the legacy IS? SRQ 10: What are good practices for legacy IS abandonment decision making?	Q10-Q22
3) The practical abandoning process of legacy IS.	3) How do organizations practically abandon legacy IS?	SRQ 11: How are legacy IS practically abandoned? SRQ 12: Who is held responsible for practically abandoning legacy IS? SRQ 13: What are good practices for practically abandoning legacy IS?	Q23-Q39
All life cycle stages.	Develop/Build.	Sub-research questions (SRQ).	Question numbers Q(n)
The aging, the abandonment decision and the practical abandoning process of legacy IS.	4) Propose a method to abandon legacy IS.	All SRQ, especially the more practical ones, e.g.: “SRQ 10 What are good practices for legacy IS abandonment decision making?”, will provide input to answer this question.	All Q’s
General/closing.			Q40-Q46

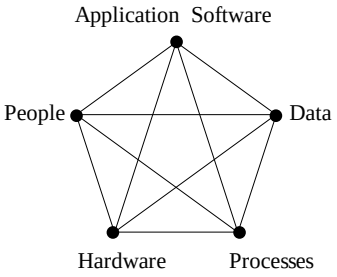
Table 49. The relationship between investigative processes, SRQ and interview questions.

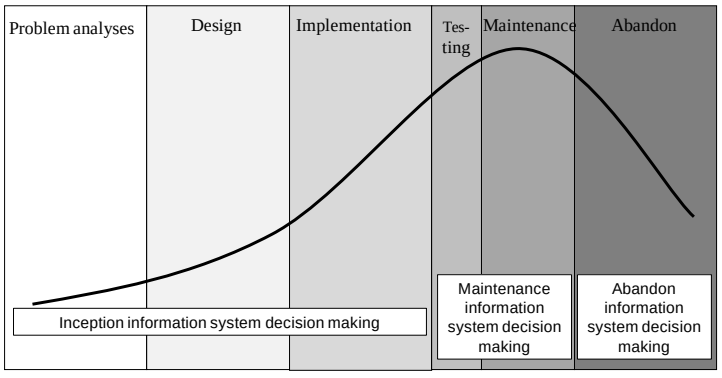
APPENDIX B DESIGN VALIDATION RESEARCH

In this appendix the design of the testing phase, the validation research (Chapter 7) is presented. Based on the research questions an explorative research is done (Chapter 3, 4 and 5), in the explanation phase (Chapter 6): first the aging of legacy IS is conceptualized, then abandonment triggers, decision making and the practical abandoning process of legacy IS were conceptualized. Finally design dilemmas on legacy IS abandonment decision making methods were described. In the first section the validation questionnaire design is presented and in the second section the analysis model is presented.

Appendix B.1 Validation questionnaire design

In Table 50 the validation questionnaire design is presented; to ensure essential data are collected, a data requirements table is created (Saunders, 2007). The data requirements table includes the actual questionnaire as used in the interviews (Columns: "Question No."; "Investigative questions" and "Variables required") and additional design information concerning the validation questionnaire (Column: "Detail in which data measured").

<table>
<tr><th colspan="3">Validation questionnaire.</th><th>Additional design information concerning the validation questionnaire.</th></tr>
<tr><td colspan="3">Name interviewee: Function:
Role: Organization:
Date: Telephone:
Email:</td><td></td></tr>
<tr><th>Question No.</th><th>Investigative questions.</th><th>Variable(s) required.</th><th>Detail in which data measured.</th></tr>
<tr><td>Q1</td><td colspan="3">The IS which is candidate for retirement is defined to consist of application software, hardware, data sets, people and procedures; the components are interrelated. This is illustrated in the following figure and defined as IS pentagon. IS have a system life cycle, during this life cycle decision making takes place. The IS life cycle is illustrated in the following figure:
 

The IS pentagon
 

The IS life cycle
In this questionnaire we look at operational IS, this means that we do not look at IS that are abandoned in their inception phase. The focus is on business applications, this is application software.</td></tr>
<tr><td></td><td>Can this definition of an IS be applied to your organization?</td><td>□ No
□ Yes</td><td>The definition of an IS can or cannot be applied to the organization. To ensure that in this interview/questionnaire the same concept of an IS is used. This should avoid biases among different interviewees and between the interviewer and the interviewee.</td></tr>
<tr><td>Q2</td><td>Does the interviewee understand and recognize these phases</td><td>□ No
□ Yes</td><td>The interviewee does or does not understand and recognize the decision</td></tr>
</table>

	(inception, maintenance, abandon) of decision making concerning IS in his organization?	□ Do not know.	making phases within the organization. To ensure that in this questionnaire the same concept of an IS is used. Does the interviewee recognize the phases in the life cycle? This should avoid biases among different interviewees and between the interviewer and the interviewee.
Q3	How many staff are employed in the organization? How many ICT staff are employed?	No. staff employed. No. ICT staff employed.	Number of staff and ICT staff employed. In analyses this might be used to distinguish between large and small companies and to know the number of IS staff.
Q4	What is the organization total revenue and total capital spend? (in Euro)	Revenue, capital spend (in Euro).	Organizational revenue and capital spend might be used to distinguish organizations within the sample.
Q5	What is the yearly ICT spend? Has this ICT spending been stable during last 3 years? Expenses to ICT staff Expenses to hardware Expenses to software Expenses Other Total	..... (in Euro) □ No □ Yes □ Do not know ... Euro ....Euro ....Euro ....Euro ....Euro (100%)	ICT spend on several categories, to know the yearly ICT spend and the percentage of ICT spend within the total spend (see Q4). It might be used to categorize later. If the ICT spend is stable it would suggest a "stable" ICT organization, this in comparison to organizations were the ICT spend has fluctuated. It might be used to categorize later. It might be used to distinguish organizations within the sample.
Q6	What % of ICT domains is outsourced? Apps development Apps maintenance Data center Working productivity Servers	0% - 100%, ...% .....% .....% .....% .....%	Percentage of outsourced domains, to know the percentage of ICT spend which is outsourced. It might be used to distinguish/categorize within the sample.
Q7	How many IS are there in your organization? (Business applications)	No. of business applications.	To know the number of IS (Business applications are counted here).
General questions concerning inception in new IS.			
Q8	Which stakeholders are involved in the decision to incept a new IS and what is their role? CIO Finance Department Other Other Average group size	 □ No □ Yes □ describe role. □ No □ Yes □ describe role. □ No □ Yes □ describe role. □ No □ Yes □ describe role.	Names, functions and roles, to understand which stakeholders participate in the decision to incept an IS and to have a description of their role. To know how many persons participate in the decision making (average group size).
Q9	Were - in your opinion - important stakeholders missing in the decision to incept a new IS?	□ No □ Yes, describe who was missing and what - in your opinion - this meant? □ Do not know.	Function names and roles, to know which stakeholders were missing in the decision to incept an IS.
Q10	Which stakeholders are responsible for the decision to incept a new IS?	Function name. Role	Function names and roles of responsible stakeholders, to know which stakeholders are responsible for the inception decision.
Q11	Which rules, if any, are applied in deciding to invest in the inception of an IS?	□ Majority vote. □ Delegation (one person is appointed to make the decision). □ Negotiation (compromise to the middle position). □ Spontaneous agreement (a decision is arrived at without considering the decision factors). □ Arbitrary (decision is made by some arbitrary means such as flipping a coin). □ Decision leader decide without discussion. □ Decision leader decide after discussion. □ Consensus (a state of mutual agreement is reached). □ Others.....	Rules applied in deciding to invest in IS, to know which rules are applied in the decision making concerning the inception decision of an IS. To see if there are differences between inception, maintenance and abandonment decision making.

		□ Do not know.	
Q12	Are inception investments of IS prioritized with other (non) IS investments?	□ No □ Yes, describe the process. □ Do not know.	To know if there is a form of portfolio management and to have a description of this prioritization process.
General questions concerning maintenance of IS.			
Q13	Which stakeholders are involved in the decision to maintain an IS and what is their role? CIO Finance Department Other Other Average group size	 □ No □ Yes □ describe role. □ No □ Yes □ describe role. □ No □ Yes □ describe role. □ No □ Yes □ describe role.	Function names and roles of responsible stakeholders, to know which stakeholders are responsible for the maintenance decision and what their role is.
Q14	Were - in your opinion - important stakeholders missing in the decision to maintain an IS?	□ No □ Yes, describe who was missing and what - in your opinion - this meant? □ Do not know.	To know which stakeholders were missing in the decision to maintain an IS.
Q15	Which stakeholders are responsible for the decision to maintain an IS?	Function name. Role	To know which stakeholders are responsible for the maintenance decision making of an IS and to understand the role of these stakeholders.
Q16	Which rules, if any, are applied in deciding to maintain an IS?	□ Majority vote. □ Delegation (one person is appointed to make the decision). □ Negotiation (compromise to the middle position). □ Spontaneous agreement (a decision is arrived at without considering the decision factors). □ Arbitrary (decision is made by some arbitrary means such as flipping a coin). □ Decision leader decide without discussion. □ Decision leader decide after discussion. □ Consensus (a state of mutual agreement is reached). □ Others..... □ Do not know.	Rules applied in deciding to maintain IS, to know which rules are applied in the decision making, concerning the maintenance decision of an IS. To see if there are differences between inception, maintenance and abandonment decision making.
Q17	Are maintenance investments prioritized with other (non) IS investments?	□ No □ Yes, describe this process. □ Do not know.	To know if there is a form of prioritization management and to gain information concerning this prioritization.
Q18	Where does the trigger to invest in the maintenance of an IS come from? Please provide some examples.	Description of triggers.	Description of maintenance triggers, to know were these triggers come from to further understand the life cycle, internal and external aging factors and to further understand this process.
Q19	Is there a portfolio management approach for IS maintenance?	□ No □ Yes, describe this process. □ Do not know.	To know if there is a form of portfolio management approach within IS.
Q20	Is there a life cycle approach for IS?	□ No □ Yes, describe this process. □ Do not know.	To know if there is an individual IS life cycle approach.
Q21	Is there a process to invest in the maintenance of IS?	□ No □ Yes, describe the process. □ Do not know.	To know if there is a process available and to further understand this process.
Q22	Are there differences between maintenance decisions concerning spending on relatively small IS and large IS?	□ No □ Yes, describe the differences. □ Do not know.	To know if there is a difference between large and small IS and to understand the difference between small and large IS.
Q23	Are there differences between the life expectancy concerning relatively small IS and large IS?	□ No □ Yes, describe this process. □ Do not know.	To know if there is a difference between the life cycle expectancy of small and large IS and to understand the difference between small and large IS.
General questions concerning the abandonment phase of an operational IS.			
Q24	How is the abandonment decision making process of an operational IS	Description of abandonment decision making process.	To gain insight in how the abandonment decision making process is organized.

	organized (or describe a practical example)?		
Q25	Is this abandonment decision making process formally described within the organizations?	□ No □ Yes describe the process. □ Do not know.	To know if there is a formal abandonment decision making process written on paper, and to understand this process.
Q26	Are alternatives for abandoning of operational IS evaluated?	□ No □ Yes, describe this process. □ Do not know.	To know if alternatives for abandoning of operational IS are evaluated and to further understand this process.
Q27	Which stakeholders were involved in the decision to abandon an operational IS and what is their role? CIO Finance Department Other Other Average group size	□ No □ Yes □ describe role. □ No □ Yes □ describe role. □ No □ Yes □ describe role. □ No □ Yes □ describe role. Number	To know which stakeholders participate in the decision to abandon an IS and what their role is.
Q28	In your opinion have important stakeholders been missed in the decision making to abandon an operational IS?	□ No □ Yes, describe who was missing and what according to your opinion did this mean? □ Do not know.	To know which stakeholders were missing in the decision making to abandon an IS, and to understand the impact.
Q29	Which stakeholders are responsible for the decision making to abandon an operational IS?	Function name. Role	To know which stakeholders were responsible for the abandonment decision. What is the role of this stakeholder?
Q30	Which rules, if any are applied in deciding to abandon an operational IS?	□ Majority vote. □ Delegation (one person is appointed to make the decision). □ Negotiation (compromise to the middle position). □ Spontaneous agreement (a decision is arrived at without considering the decision factors). □ Arbitrary (decision is made by some arbitrary means such as flipping a coin). □ Decision leader decide without discussion. □ Decision leader decide after discussion. □ Consensus (a state of mutual agreement is reached). □ Others..... □ Do not know.	Rules applied in applied in the decision making concerning the abandonment decision of an IS, to know which rules are applied in the decision making concerning the abandonment decision of an IS. To see if there are differences between inception, maintenance and abandonment decision making.
Q31	Do you consider abandonment decisions concerning operational IS to be successful? How do you define successful?	□ No □ Yes □ Do not know.	To know if the interviewees consider an abandonment decision to be successful and what to define as success.
Q32	What are success criteria and what are problems in the decision making process of abandoning operational legacy IS decisions?	Describe success criteria and problems.	Description of success criteria and problems, to know what interviewees consider to be success factors and problems in decision making and to understand how these problems are handled.
Q33	What are success criteria and what are problems, in the process of practical abandoning operational legacy IS decisions?	Describe success criteria and problems.	Described success criteria and problems, to know what interviewees consider to be success criteria and problems in practical abandoning and to understand how these problems are handled.
Q34	Is the impact of abandoned operational legacy IS known on other IS within the organization?	□ No □ Partly (%) □ Yes, how is this impact researched and what does this mean? □ Do not know.	To know what the impact is of abandoned legacy IS on other IS and to further understand this impact.
Q35	Where does the triggers to decide to abandon operational IS come from? Total is 100%	-A merger of organizations with double functionality. ... % -A new technology. ... % -A product or product line which is abandoned. ... % -Old infrastructure. ... % -No support from suppliers. ... %	Scored triggers to decide to abandon IS, to know what the triggers are and to distinguish between internal aging and external aging.

		-High cost. ... % -Inadequate software quality. ... % -Other ... % -Other ...% Total 100%	
Q36	The decision making to abandon a primary IS is easier than a secondary IS?	□ Agree, because. □ Disagree, because. □ Do not know.	To distinguish differences in decision making between primary IS compared to secondary IS (a primary IS is an IS supporting a vital business function).
Q37	The decision making to abandon a proprietary (own build) IS is easier than one which is bought commercial off-the-shelf (COTS)?	□ Agree, because. □ Disagree, because. □ Do not know.	To distinguish differences in decision making between proprietary (own build) IS and COTS IS, and to further understand this difference.
Q38	Rank in order, 1, 2 and 3 which is the most important discipline in a decision making concerning the abandoning of operational IS. (1 is most important).	.....Political .....Mathematical .....Statistical .....Economical .....Sociological .....Technological .....Anthropological .....Functional .....Security .....Philosophical .....Psychological	A ranking of the most important disciplines in IS abandonment decisions according to the interviewees.
Specific open questions concerning the abandonment decision of one IS (one specific business application). Please provide documentation if possible.			
Q39	What is the application Name? Can you describe the IS? When was it built? In what program language was it built? The size of the IS compared to others within your organization? How many people were working with this IS? Can you describe the life cycle of the IS? Can you tell me how the maintenance was performed? Who was responsible for the maintenance decision? Describe internal aging factors and external aging factors to this IS. What happened to decide to abandon the IS (triggers)? Describe the abandonment decision process. Who was responsible for the abandonment decision? Describe the process to practically abandon the IS. What went well? What went wrong? Lessons learned?	Name Description Year/age Language □ Large □ Medium □ Small Number Description Description Function name and role. Internal aging factors. External aging factors. Triggers Description Function name and role. Description Description Description Description	A description of one IS within an organization, it might be used to distinguish between a technical interviewee and a business interviewee and to further understand the life cycle, internal and external aging factors, abandonment triggers, decision making and good practices.
These following questions relate the total life cycles of all IS.			
Q40	How much of your ICT budget in % is spent on IS inception, IS maintenance and IS abandoning? Total is 100%	IS inception .....% IS maintenance .....% IS abandoning .....% Total 100%	ICT spend relatively divided in: inception, maintenance and abandoning, to understand what money is spent on these three phases.
Q41	Relative time involvement of higher management versus lower management in inception, maintenance and abandonment decision of IS with similar financial impact.	Higher Mngt / Lower Mngt Inception ...% / ...% Maintenance ...% / ...% Abandonment ...% / ...% Total 100% / 100%	Relative time involvement of higher management compared to lower management, within the several IS lice cycle phases, to understand time involvement of higher and lower management in the different phases.
Q42	Questions concerning financial impact of decision making.		Financial impact known in the several IS life cycle phases, to understand if the

	Is the financial impact known of an inception decision of new IS? Is the financial impact known of a maintenance decision of an IS? Is the financial impact known of an abandonment decision of an IS?	Not at all 1 2 3 4 5 completely □ Do not know. Not at all 1 2 3 4 5 completely □ Do not know. Not at all 1 2 3 4 5 completely □ Do not know.	financial impact on decision making in several phases is different.
Q43	Please score and explain: High (H) Medium (M) Low (L) risk in the different phases of the IS life cycle. Inception decision Maintenance decision Abandonment decision	Risks: □ H □ M □ L explain □ H □ M □ L explain □ H □ M □ L explain	A score on risk in several phases of the IS life cycle, to know whether organizations consider the inception of a new IS to be more risky than maintaining a current IS, and to further understand these risks.
Q44	Estimate working staff (# FTE) of organization on Inception, Maintenance, Abandonment. Total 100%	IS inception .....% IS maintenance .....% IS abandonment .....% Total 100%	The percentage score of working staff on the three life cycle phases, to know what % is working in which phase of the IS life cycle.
Q45	Is there a difference for staff working in inception phase, maintenance phase or abandonment phase?	□ No □ Yes, describe the difference. □ Do not know.	Difference for staff working in different IS phases, to understand if inception of IS is higher valued by staff than maintenance of IS.
Q46	Rank job satisfaction off employees and status off work for employees on inception, maintenance and abandoning activities, 1 is high in rank, 3 is lowest in rank.	Job satisfaction Inception … Maintenance … Abandonment …	Difference in job satisfaction for staff working in different IS phases, to know if there is a difference in job satisfaction of employees within the life cycle of an IS.
Q47	Do you think that operational IS within your organization are well-documented? Describe what your definition is of well-documented.	□ No □ Yes, please explain. □ Do not know.	To know if the documentation of IS is sufficient, and to further understand the definition of well-documented.
Q48	How satisfied are you with the documentation of an IS within the IS life cycle phases Inception, Maintenance or Abandoning? Inception IS Maintenance IS Abandoning IS If there are differences please describe.	□ Do not know □ Satisfied □ Neutral □ Unsatisfied □ Do not know □ Satisfied □ Neutral □ Unsatisfied □ Do not know □ Satisfied □ Neutral □ Unsatisfied Description	Description of the satisfaction of the documentation of an IS in the life cycle phases, to know if the documentation of IS is different in different phases.
Q49	Estimate IS life time spending in euros on Inception, Maintenance, Abandonment total is 100%	Inception …% Maintenance ...% Abandonment ...% Total 100%	Spending within the IS life cycle, to know the relatively spend in the IS life cycle.
Q50	Are you missing relevant questions?	□ No □ Yes, namely	To ensure no relevant questions are missed and to improve the questionnaire.
Q51	Do you have suggestions for improving this questionnaire?	□ No □ Yes, namely	To ensure improvements of the questionnaire.
Q52	Do you have suggestions for other internal or external stakeholders to be interviewed?	□ No □ Yes, namely	To gain more names.
Q53	Can I reference you for approaching these stakeholders?	□ No □ Yes	To have easier access to follow on interviewees.
Q54	Do you like to have feedback of the total study?	□ No □ Yes	To feedback the results of the research.
Q55	How many years work experience within the organization?	No. years.	Validation; assumption is: more years higher reliability.
Q56	How many years work experience in this field?	No. years.	Validation; assumption is: more years higher reliability.
Q57	In case I have additional questions, is it possible to have an additional interview?	□ No □ Yes	

Table 50. The validation questionnaire

Appendix B.2 Analysis model validation research

Table 51 illustrates the analysis model within the testing phase. The analysis model provides the relationship between the investigative life cycle processes, the appropriate research questions, sub-research questions (SRQ) and interview questions (Q).

Understanding Legacy information systems (IS) and abandonment decision making. Towards methodological support.			
Primary research question: How do organizations identify legacy IS and how do they manage abandonment?			
Start	Introduction and company information.		Question number: Q1-Q7
Life cycle stage.	Research questions.	Sub-research questions (SRQ).	Question numbers Q (n)
1) The aging process of legacy IS.	1) How do legacy IS age?	SRQ 14: What are internal legacy IS aging factors? SRQ 15: What are external legacy IS aging factors?	Q18, Q35, Q39
2) The abandonment decision making process of legacy IS.	2) How do organizations decide to abandon legacy IS?	SRQ 16: What are triggers to abandon legacy IS? SRQ 17: What legacy IS abandonment decision making perspective (e.g. technological, functional, economical) is predominant? SRQ 18: To what extent does a commercial off-the-shelf (COTS) or a proprietary (own build) IS influence the characteristics of abandonment decision making? SRQ 19: To what extent do IS supporting primary business processes versus IS supporting secondary processes influence the characteristics of abandonment decision making? SRQ 20: To what extent does the size of an IS (small or large) influence the characteristics of abandonment decision making? SRQ 21: What are good practices in legacy IS abandonment decision making?	Q8 - Q32, Q36 - Q49
3) The practical abandoning process of legacy IS.	3) How do organizations practically abandon legacy IS?	SRQ 22: What are good practices in practically abandoning legacy IS?	Q33, Q34, Q39
All life cycle stages.	Develop/Build.	Sub-research questions (SRQ).	Question numbers Q (n)
The aging, the abandonment decision and the practical abandoning process of legacy IS.	4) Validating design dilemmas for a method to abandon legacy IS; Propose a method to abandon legacy IS.	SRQ 23: Enhancing existing IS decision making methods or designing a new legacy IS abandonment decision making method?	Q8 - Q32, Q36 - Q49
		SRQ 24: Decision making at information function, IS or application level?	Q39
		SRQ 25: Financial versus non-financial evaluation aspects?	Q39
		SRQ 26: Many or few attributes in the abandonment decision making method?	Q24
		SRQ 27: Life cycle or portfolio oriented legacy IS assessment?	Q19, Q20
		SRQ 28: Many stakeholders versus few stakeholders involved in legacy IS abandonment decision making?	Q27
General/closing.			Q50 - Q57

Table 51. The relationship between life cycle stage of investigative processes, research question, sub-research questions interview questions.

APPENDIX C LEGACY INFORMATION SYSTEM AGING

Appendix C.1 Legacy information system aging characteristics

In this section in total 140 specific legacy characteristics (aging factors) of the IS (IS-related legacy characteristics) and the individual components of the IS (application-related legacy characteristics, hardware-related legacy characteristics, data-related legacy characteristics, people-related legacy characteristics and procedure-related legacy characteristics) as found in the literature and in Case A are described and categorized in internal and external aging factors.

Concerning IS-related legacy characteristics 42 legacy characteristics were found in the literature and in Case A. The legacy IS characteristics are categorized exclusively internal (27) and exclusively external (3) both internal and external (12), and the results are presented in Table 52.

No.	IS-related legacy characteristics.	Source	(I)nternal (E)xternal
1	Limited architectural fit.	Ulrich (2002); Case A	I
2	Aging legacy architectures. Architect (4) said: *"IS which are not compatible with state of the art technology or consisting of non-proven or "exotic" technology are considered legacy IS characteristics".*	Alderson and Shah (1999); Ulrich (2002); Case A	I/E
3	Systems designed in stove pipe fashion or silos.	Daga et al. (2005); Kelly et al. (1999); Ulrich (2002); Van den Heuvel (2007); Case A	I
4	Legacy IS may hinder progress (not agile).	Daga et al. (2005); Case A	I
5	Legacy IS as a barrier to strategic innovation.	Kelly et al. (1999)	I/E
6	The documentation becomes increasingly inaccurate thereby making future changes even more difficult.	Parnas (1994)	I
7	Inadequate documentation of the system.	Adolph (1996); Warren (1999); Case A	I
8	Understanding of system details is often lacking.	Bisbal et al. (1999); Warren (1999); Case A	I
9	System skills hard to find.	Alderson and Shah (1999); Kelly et al. (1999); Case A	I/E
10	Diminishing support from the supplier.	Furneaux and Wade (2011); Case A	E
11	Difficult, if not impossible, to extend.	Bisbal et al. (1999)	I
12	Increasing complexity.	Kelly et al. (1999); Richmond et al. (2006); Ulrich (2002); Case A	I
13	Declining system reliability.	Furneaux and Wade (2011); Case A	I
14	Inadequate performance.	Alderson and Shah (1999)	I
15	Inadequate interoperability, ambiguous interfaces prohibits integration with other systems.	Alderson and Shah (1999); Bisbal et al. (1999); Brodie and Stonebraker (1995); Case A	I
16	Lack of integration of systems, dispersed landscape (e.g. locally focused and fragmented).	Kelly et al. (1999); Case A	I
17	No longer meeting the needs of the environment.	Brooke (2002)	I/E
18	Significantly resisting modification and evolution in order to meet new and constantly changing business requirements.	Alderson and Shah (1999); Brodie and Stonebraker (1995); Kelly et al. (1999)	I
19	Growing gap between capabilities of the system and the needs of the business in which it is used.	Brooke (1994); Comella-Dorda et al. (2000); Kelly et al. (1999); Larsen (1998); Warren (1999)	I/E
20	Increasing security issues.	Alderson and Shah (1999); Case A	I/E

21	Benefits of system are lower than costs.	Alderson and Shah (1999); Case A	I
22	Risks. Business information manager (2) said: *"when the IS is exposing an organization to risks, this is considered to be a legacy IS characteristic".*	Case A	I
23	Prevent organizations from fully exploiting computers and the value they bring in organizations.	Ulrich (2002)	I
24	Legacy portfolios are 30-50% more expensive to run than their packaged systems or newer technology.	Hunter and Aron (2006)	I
25	Excessive system support costs.	Furneaux and Wade (2011)	I
26	Expensive to maintain and operate, high IS cost.	Brodie and Stonebraker (1995); Warren (1999); Case A	I/E
27	No cost awareness of users of the IS.	Case A	I
28	IS costs and IS benefits are unknown.	Case A	I
29	Fewer customers helped by more expensive IS, high cost per user.	Case A	I/E
30	Cheaper and/or better alternative.	Case A	I/E
31	There is no cost transparency resulting in worse decision making.	Case A	I
32	Similar or redundant IS capabilities (e.g. due to a merger).	Case A	I
33	A poor quality of the IS.	Case A	I
34	Lack of maintenance/dis-investment for many years on the IS.	Case A	I
35	Unstable IS.	Case A	I
36	Workarounds on legacy IS, unintended use of the IS.	Case A	I
37	The capabilities provided by the IS are less used, less needed or no longer needed anymore by the users or the customers (e.g. a products or services supported by the IS is abandoned).	Case A	I/E
38	Non-scalability, the IS is not capable to meet rising demands.	Case A	I/E
39	Lots of manual interventions.	Case A	I
40	(In an outsourced situation) legacy IS characteristics are: vendor lock-in, dependency on a vendor, resulting in optimization from a vendor perspective.	Case A	E
41	IS contracted as part of lumpsum with suppliers or that include "exit" costs of abandoning an IS.	Case A	E
42	Individual IS which constitutes different contracts with different suppliers (e.g. data center, application maintenance, licenses, support) which have different suppliers, owners and stakeholders within the organization.	Case A	I/E

Table 52. IS-related legacy characteristics

Above table describes the legacy characteristics from an IS perspective; besides these, also legacy characteristics of the individual components of the IS are found in legacy IS literature.

The following 46 application-related legacy characteristics were found in legacy IS literature and in Case A. The legacy IS characteristics are categorized exclusively internal (36) and exclusively external (7) both internal and external (3). These are presented in Table 53.

No.	Application-related legacy characteristics.	Source	I/E
1	Inadequate software quality. E.g. structural deterioration: due to time pressure new functionality is added and the software is not programmed in a structured way or old code is not removed by a programmer, resulting in dead code, in the software which does not support any operational functionality anymore, or there is duplicated code which is the result "quick and dirty" programming. Aging software often degrades in its performance as a result of a gradually deteriorating structure; aging software often becomes "buggy" because of errors introduced when changes were made. Written in the past when the main design considerations were memory optimization and processing speed instead of program structure.	Adolph (1996); Bennet (1995); Brodie and Stonebraker (1995); Chan et al. (1996); Kelly et al. (1999); Krishnan et al. (2004); Lehman (1980a); Lehman (1980b); Lehman and Ramil, (2001); Parnas (1994); Richmond et al. (2006); Sakthivel (1994); Seacord et al. (2003); Swanson and Dans (2000); Van den Heuvel (2007); Case A	I
2	Complexity of the software (increases).	Chan et al. (1996); Krishnan et al. (2004); Lehman (1980a); Lehman (1980b); Seacord et al. (2003); Warren (1999); Case A	I
3	Reduced or inadequate performance (e.g. due to growing application size and inadequate design).	Brodie and Stonebraker (1995); Krishnan et al. (2004); Parnas (1994)	I
4	Decreasing reliability due to errors which are introduced during the application maintenance.	Krishnan et al. (2004); Parnas (1994)	I
5	(Structural) lack of maintenance on the application.	Swanson and Dans (2000); Case A	I
6	Use of older program language, or technical opportunities available (e.g. higher generation languages or object oriented methods).	Adolph (1996); Bennet (1995); Brodie and Stonebraker (1995); Chan et al. (1996); Krishnan et al. (2004); Richmond et al. (2006); Zvegintzov (1984)	I
7	Supporting software: compilers no longer supported by hardware supplier.	Sommerville (2007); Warren (1999); Case A	E
8	Support from application supplier has ended.	Furneaux and Wade (2011)	E
9	No hardware support for the platform on which the application is running.	Adolph (1996); Parnas (1994); Zvegintzov (1984); Case A	E
10	Software does not have an architectural fit.	Richmond et al. (2006); Case A	I
11	Limited agility (changes take longer).	Parnas (1994); Case A	I
12	"Dying" application programme language, application programming language is hardly taught at universities anymore.	Computerworld, 8th April, 2013; Case A	E
13	Skills are hard to find, staff with knowledge skills concerning the application are retired or have left the organization and taken a substantial amount of knowledge with them and the application is not well-documented, new programmers do not have this knowledge and up-to-date documentation of the application is usually lacking.	Adolph (1996); Alderson and Shah (1999); Brodie and Stonebraker (1995); Daga et al. (2005); Parnas (1994); Seacord et al. (2003); Case A	I
14	Limited or missing documentation of the application.	Bennet (1995); Bisbal et al. (1999); Brodie and Stonebraker (1995); Daga et al. (2005); Parnas, (1994); Zvegintzov (1984); Case A	I
15	Inadequate documentation of the application.	Adolph (1996); Kelly et al. (1999); Parnas (1994); Seacord et al. (2003); Case A	I
16	Software embedded in the hardware.	Adolph (1996); Ulrich (2002)	I
17	Embedded business logic.	Warren (1999)	I
18	Understanding of system details is often lacking.	Bisbal et al. (1999); Daga et al. (2005); Parnas, (1994)	I
19	Tracing faults within applications is costly and time consuming.	Bisbal et al. (1999)	I
20	New functional requirements.	Lehman (1996); Parnas, (1994); Van den Heuvel (2007); Zvegintzov (1984); Case A	I
21	Functionality does not meet the requirements.	Bennet (1995); Krishnan et al. (2004); Lehman (1996); Parnas, (1994); Van den Heuvel (2007); Case A	I

22	Changing user perception. Business information manager (3) quoted the following analogy concerning changing user perception*: "Fifteen years ago nobody had an air-conditioning in their car and nobody missed it, nowadays a car without an air-conditioning makes it a legacy car".*	Lehman (1996); Parnas, (1994); Seacord et al. (2003); Case A	I/E
23	Alternatives/opportunity cost (a better or cheaper IS available).	Sakthivel (1994); Zvegintzov (1984); Case A	E
24	Expensive to maintain.	Bennet (1995); Bisbal et al. (1999); Brodie and Stonebraker (1995); Chan et al. (1996); Daga et al. (2005); Krishnan et al. (2004); Parnas, (1994); Sommerville (2007); Swanson and Dans (2000); Case A	I
25	Increasing maintenance costs of applications.	Chan et al. (1996); Lehman (1980b); Lehman (1980a); Richmond et al. (2006); Sakthivel (1994); Seacord et al. (2003); Case A	I
26	High application costs.	Case A	I/E
27	Source code not available.	Case A	I
28	No access to source code.	Case A	E
29	No knowledge of the application.	Case A	I
30	Supplier goes bankrupt.	Case A	E
31	The core functionality of a system (e.g. calculating unemployment benefits for employees) which is static and very stable is surrounded by less stable and dynamic process related functionality.	Case A	I
32	Application is difficult to maintain.	Case A	I
33	No programmer skills available.	Case A	I/E
34	Lower versions of the software, not using latest release.	Case A	I
35	Software not developed with "computer aided software engineering" (CASE) tooling.	Case A	I
36	Not service oriented, software architecture is not service oriented.	Case A	I
37	Not latest design environment, software not developed in latest design environment.	Case A	I
38	Applications are programmed in older generation languages and not programmed in components and therefore difficult to untangle and to maintain.	Case A	I
39	Redundant (double or similar) functionality.	Case A	I
40	Growing functionality gap.	Case A	I
41	It is no longer possible to do major functionality changes as the business requires.	Case A	I
42	Functionality no longer needed, no users anymore, or no products anymore. Another positioning of the Case A services.	Case A	I
43	Functionality of an application becomes of lesser value.	Case A	I
44	Workarounds by users (e.g. functionality is incorrectly used). For example, a remark field of a customer is used to fill in a second mobile number, due to the fact that the original functionality provided only one phone number.	Case A	I
45	Not meeting user expectations.	Case A	I
46	Lack of objective measures, like cost per function points, cost per release. This will make decision making more subjective.	Case A	I

Table 53. Application-related legacy characteristics

The following 19 hardware-related legacy characteristics were found in legacy IS literature and in Case A. The legacy IS characteristics are categorized exclusively internal (14) and exclusively external (3) both internal and external (2). These are presented in Table 54.

No.	Hardware-related legacy characteristics.	source	I/E
1	Inadequate performance or slow.	Adolph (1996); Bisbal et al. (1999); Daga et al. (2005); Kelly et al. (1999); Parnas, (1994); Case A	I
2	Poor reliability.	Parnas (1994); Case A	I
3	Lack of knowledge of old hardware.	Alderson and Shah (1999)	I/E
4	Old hardware type.	Brodie and Stonebraker (1995); Sommerville (2007); Warren (1999); Case A	I
5	Old platform type.	Adolph (1996); Parnas (1994); Case A	I
6	Legacy IS run on obsolete hardware.	Bisbal et al. (1999); Daga et al. (2005)	I
7	Hardware supplier discontinues the product line.	Adolph (1996); Case A	E
8	Not compatible with current procurement policies.	Sommerville (2007)	I
9	Expensive to maintain.	Bisbal et al. (1999); Brodie and Stonebraker (1995); Daga et al. (2005); Sommerville (2007); Case A	I
10	Hardware embedded with software.	Adolph (1996); Case A	I
11	Inadequate stability.	Case A	I
12	Poor availability.	Case A	I
13	No architectural fit.	Case A	I
14	"Dying" platform.	Case A	E
15	Supplier bankrupt.	Case A	E
16	Four decentralized data centers instead of one centralized. Redundant data centers (not being fail over).	Case A	I
17	Alternative (better and/or cheaper) available (opportunity costs).	Case A	I/E
18	High hardware costs.	Case A	I
19	High exploitation costs.	Case A	I

Table 54. Hardware-related legacy characteristics

The following 16 data-related legacy characteristics were found in the legacy IS literature and in Case A. The legacy IS characteristics are categorized exclusively internal (15) and exclusively external (1). These are presented in Table 55.

No.	Data-related legacy characteristics.	source	I/E
1	Immense volume of data has accumulated over the lifetime of the system. This data may be inconsistent and may be duplicated in several files.	Sommerville (2007)	I
2	Aging data architecture; aging database.	Adolph (1996); Brodie and Stonebraker (1995); Ulrich (2002)	I
3	Data is defined and stored redundantly across multiple stovepipe business units and applications.	Ulrich (2002)	I
4	The same or similar data is defined inconsistently across multiple systems.	Ulrich (2002)	I
5	The same data terminology may be used to define different data across multiple applications and business units.	Ulrich (2002)	I
6	The integrity of the data may be poor and contain information it should not contain.	Ulrich (2002)	I
7	Data may not be easily accessible by modern systems or through user-based inquiries.	Ulrich (2002)	I
8	Data cannot be readily shared across systems, business units and organizational boundaries.	Ulrich (2002)	I
9	Corrupted data or unreliable data.	Case A	I
10	Redundant data, redundant data sources.	Case A	I
11	Old database, no support from database supplier.	Case A	E
12	Database not conform to the database architecture.	Case A	I
13	Security.	Case A	I
14	Data was included in the software as a file system, instead of in a relational database. The data is not stored in relational database, but in hierarchical databases or even flat files (being part of the software).	Case A	I
15	Performance and capacity of a database is not sufficient anymore.	Case A	I
16	Decentralized databases (each office has a decentralized database, data is collected once every day and sent to a data warehouse).	Case A	I

Table 55. Data-related legacy characteristics

In the legacy IS literature and in Case A the following six procedural-related legacy characteristics were found. The legacy characteristics are categorized exclusively internal (6). These are presented in Table 56.

No.	Procedure-related legacy characteristics.	Source	I/E
1	Do not take advantage of streamlined facilities and standard routines; use programming and design practices that current management would not permit.	Zvegintzov (1984)	I
2	Definitions of how the business should be carried out and constraints on the business. Use of the legacy application system may be embedded in these policies and rules.	Sommerville (2007)	I
3	Inadequate support documentation.	Case A	I
4	Maintenance procedures outdated or not in place.	Case A	I
5	Outdated test procedures.	Case A	I
6	Internal hiccups in the procedures and processes of supporting the IS.	Case A	I

Table 56. Procedure-related legacy characteristics

The following 11 people-related legacy characteristics were found in the legacy IS literature and in Case A. The legacy characteristics are categorized exclusively internal (8) and both internal and external (3). These are presented in Table 57.

No.	People-related legacy characteristics.	source	I/E
1	Lacking technical expertise, retiring staff, skills hard to find.	Adolph (1996); Daga et al. (2005); Warren (1999)	I/E
2	Dislike of the tedious work on legacy IS, "if something is difficult, tedious and slow we will try to avoid doing it". Engineers prefer working on new system development instead of maintaining old software.	Adolph (1996); Bennet (1995); Ulrich (2002)	I
3	Changes made by people who do not understand the original design concept almost always cause the structure of a programme to degrade; after many such changes the original designers no longer understand the modified product.	Parnas (1994)	I
4	Many people who are doing software development do not have an education appropriate to the job.	Parnas (1994)	I
5	No knowledge of the IS available anymore (skills left the organization resulting in no knowledge of the IS anymore); application programme languages not taught at the university anymore.	Case A	I/E
6	Expertise is no longer available anymore or only available at high costs.	Case A	I/E
7	Expertise of the system is lacking, making the IS vulnerable, due to the fact that people left.	Case A	I
8	Knowledge and expertise only available in the heads of persons and not on paper makes an organization dependent on these persons (knowledge is power).	Case A	I
9	Expertise of the IS is splintered across different stakeholders within different business units in the organization.	Case A	I
10	IS professionals derive, power and status from IS, abandoning IS is abandoning power and status.	Case A	I
11	Myopia of stakeholders concerning their own IS, not open to something new, or different.	Case A	I

Table 57. People-related legacy characteristics

Appendix C.2 New legacy information system aging characteristics, from validating research

In Table 58 nine new legacy IS characteristics are presented, these were found in the validation research. These legacy IS aging characteristics are categorized in exclusively internal legacy IS aging factors, no new external legacy IS aging factors were found.

New IS-related legacy characteristics.	**Source**	**I/E**
Standardization is preferred.	Case C	I
Enormous proliferation of (unknown) information systems in unknown (different) places.	Case B, Case C, Case D	I

New application-related legacy characteristics.	**Source**	**I/E**
COTS software changed according to the working process instead of using the standardized process.	Case B, Case C	I
A lot of unnecessary functionality added to the IS.	Case B	I
Inadequate usability.	Case C	I
Inadequate maintainability.	Case C	I
Company policy changed to COTS, which means that custom-made software is no longer preferred.	Case C	I
Outsourcing is preferred and all applications are reassessed (first clean your application landscape before you outsource).	Case C	I

New data-related legacy characteristics.	**Source**	**I/E**
Data is no longer needed.	Case C	I

Table 58. New internal aging factors of IS

Appendix C.3 Legacy information system abandonment triggers

In the legacy IS literature and Case A 31 legacy IS abandonment triggers were found, these are presented in Table 59.

No.	Legacy IS abandonment triggers.	Remarks	Source	IS	Applications	Hardware	Data sets	People	Procedures	(I)nternal (E)xternal
1	Cost-benefit.	No business value anymore.	Case A	X						I / E
2	High IS cost.	High legacy IS cost, lower cost alternative possible. Every legacy IS not abandoned, will consume scarce resources. Growing marginal maintenance cost due to the fact that an IS becomes more complex, opportunity cost.	Chan et al. (1996); Gode et al. (1990); Richmond et al. (2006); Sakthivel (1994); Case A	X						I / E
3	High cost per user or customer.	High cost per user or customer: "Only supporting a few users (or no users) while costs are fixed. The legacy IS is built for 400,000 customers and only serving 4,000 customers".	Case A	X						I / E
4	Aged/old hardware.	It no longer runs on its (aging) hardware, taxing the limits of the legacy systems host computer.	Adolph (1996); Furneaux and Wade (2011); Zvegintzov (1984)			X				I
5	Inadequate software quality.	Due to ongoing software maintenance, the software structure deteriorates and maintenance becomes problematic.	Adolph (1996); Chan et al. (1996); Furneaux and Wade (2011); Kelly et al. (1999); Sakthivel (1994); Swanson and Dans (2000)		X					I
6	New technological opportunities.	Superior hardware and software alternatives are available (at a lower price).	Chan et al. (1996); Furneaux and Wade (2011); Sakthivel (1994); Zvegintzov (1984); Case A		X	X	X			E
7	Limited or ended supplier support.	Limited or ended supplier support concerning software and hardware. Supplier stops services (end of service) or is bankrupt.	Adolph (1996); Furneaux and Wade (2011); Kelly et al. (1999); Zvegintzov (1984); Case A		X	X				E
8	Redundant IS.	Redundant systems.	Kelly et al. (1999)	X						I
9	Merger or consolidation.	Due to the fact that two or more organizations merge having the same services and products, a decision has to be made to abandon one or more IS because there is similar or double functionality.	Case A	X						I
10	Redundant application functionality.	Similar or double application functionality within an organization.	Case A		X					I
11	Limited or no technical skills available.	Reduced availability of individuals with the required expertise; technical skills not available anymore.	Furneaux and Wade (2011); Kelly et al. (1999)					X		I / E

12	Infrastructure overhaul.	Partial or complete disassembly of the infrastructure.	Swanson and Dans (2000)			X				I
13	Application functionality no longer needed.	Application functionality is no longer needed.	Zvegintzov (1984)		X					I
14	Application functionality gap/not sufficient.	The gap between required functionality and delivered functionality is so large that the IS should be abandoned.	Case A		X					I
15	Capability shortcomings.	When an IS is not or only partly capable of delivering the capability as required by an organization, there are system capability shortcomings. When the gap between business needed IS capability and supplying capability is growing and is too large, the IS is candidate for abandoning.	Comella-Dorda et al. (2000); Furneaux and Wade (2011); Kelly et al. (1999); Larsen (1998); Zvegintzov (1984)	X						I
16	Security issues.	A legacy IS which is vulnerable to cyber-attacks might be abandoned due to security issues.	Case A		X	X	X			I
17	Legal compliance (law).	By law (legal compliance) legacy IS were abandoned. Examples are: business functionality went from Case A to IRS, making the legacy IS surplus. New law required different calculations. Changing law by politicians; when by law a certain social security product line or service line within social security that the case organization is supplying to the clients is abandoned, the legacy IS supplying this functionality should be abandoned.	Case A	X						E
18	Does not fit in technological strategy.	The legacy IS does not fit the technological strategy, it is not conform to the architectural guidelines.	Case A		X	X	X			I
19	Instability of the IS.	Instability of the legacy IS (due to technical changes or a platform which is unstable).	Case A		X	X				I
20	Non-scalability of the IS.	For example, demand is rising; however, due to technical constraints, it is not possible to meet this demand, therefore the legacy IS should be abandoned". In case of a merger, a large IS can handle the small additional volumes relatively easily, while a small IS might have difficulties to handle large amounts of extra volume.	Case A		X	X	X			I
21	Technical aging.	Technical aging was a trigger to abandon legacy IS. For example, when there are no upgrades available anymore or there are outdated procedures or there is no support for the assembler anymore.	Case A		X	X	X		X	E
22	The IS is non-agile.	If the legacy IS is non-agile, the organization is not able to quickly adapt to changing environmental circumstances; therefore the legacy IS should be abandoned.	Case A	X						I
23	Structural lack of IS maintenance.	A lack of structurally maintaining the system will result in the trigger to abandon the IS. Business information manager (1) said: *"If you do not put oil in your engine, you have the guarantee that your car will stop one day".*	Case A		X	X				I
24	Vendor dependency/ vendor lock-in.	Unwanted vendor dependency or vendor lock-in was a trigger to abandon a legacy IS. "A strategic IS was built and maintained by a supplier, there was a vendor lock-in. That was an unfavorable	Case A	X						E

		situation and the IS was abandoned and changed by an IS, owned by the organization.								
25	Effort to migrate applications from an old data center to a new data center.	"Due to the fact that Case A selected a new data center, for each application a decision had to be taken to abandon or to migrate the application from a former data center to a new data center". "Migrating a legacy IS was a lot of work this made it more attractive to abandon the legacy IS".	Case A		X					I
26	Redundant data centers.	Redundant data centers should be abandoned. In Case A, one new data center was selected and four data centers were abandoned.	Case A			X				I
27	Redundant data sources.	Redundant data sources, in one case there were three redundant data sources.	Case A				X			I
28	Perception of the required IS capability is changed.	"Changing user perception; users nowadays have computers at home, their perception of an IS is changing". Example: "IS with monochrome displays and without mouse functionality". "Perception of the needed functionality is changed" (by organization or by users or by customers).	Case A	X						I
29	Abandoning products or services supported by the IS.	Products or services supported by the legacy IS were abandoned; there was no justification for the legacy IS anymore (e.g. another positioning of the Case A services).	Case A	X						I
30	Capability not needed.	The legacy IS was abandoned when the capability of the legacy IS was not needed (e.g. product or service delivered by the IS is discontinued).	Case A	X						E
31	New demands from the market which cannot be provided by the IS.	"If clients expect another service this might be a trigger to abandon a legacy IS (e.g. in the past clients used to correspond with Case A by means of letters and Case A scanned letters into an IS; nowadays clients are communicating by email or phone and the IS providing the scanning functionality should be abandoned)". "Case A is working more and more in supply chains; there is the: "National Reference Architecture; it describes references and principles on behalf of the government. It is used to ensure that several different Government bodies are able to communicate and share information with each other. If the data structure of an IS, or the communication protocol is not according to the National Reference Architecture and not working in the information chain, the legacy IS might be abandoned". "New demands from the market (external market environment) which could not be provisioned by the current IS; changing external market environment".	Case A	X						E

Table 59. Abandonment triggers

Besides triggers to abandon also an equivocal trigger was found. According to interviewees this was a trigger to abandon a legacy IS, but also a trigger not to abandon a legacy IS. This equivocal trigger is presented in Table 60.

No.	Equivocal legacy IS abandonment trigger.	Remarks	Source	IS	Applications	Hardware	Data sets	People	Procedures	(I)nternal (E)xternal
32	Complexity.	If a legacy IS is very complex, this might be an argument to abandon a legacy IS, due to the fact that implementing changes in a complex legacy IS is risky. On the other hand: abandoning a complex legacy IS is more difficult and risky than a less complex legacy IS. If a legacy IS is very complex, this might be an argument not to abandon or to postpone abandoning a legacy IS. By maintaining the IS, the IS is getting more complex and therefore maintenance will become more difficult and more expensive. It was formulated by Furneaux and Wade (2011) as the opposite: low-levels of technical integration (low complexity) were key determinant of increased intentions to replace an existing system.	Case A Furneaux and Wade (2011)		X	X				I

Table 60. Equivocal abandonment trigger

Appendix C.4 New legacy information system abandonment triggers, from validating research

In Table 61 six new legacy IS abandonment triggers are presented that were found in the new data set. No new external legacy IS aging factors were found.

No.	Legacy IS abandonment triggers.	Remarks	Source	IS	Applications	hardware	Data sets	People	procedures	(I)nternal (E)xternal
33	Enormous proliferation of (unknown) information systems in unknown (different) places.	The number of application has increased beyond control, there is an enormous proliferation of (unknown) information systems in unknown (different) places.	Case C, Case B	X	X	X				I
34	Outsourcing, first clean your IS landscape.	Organizations who want to outsource, first have to rationalize their IS landscape. Outsourcing is a lot of work; if a lot of legacy IS are abandoned upfront, this saves a lot of resources.	Case C	X	X	X				I
35	Standardization.	All legacy IS which are not in conformance with a standardization are abandoned (e.g. application no longer applies to standards).	Case C	X						I
36	Internal directive COTS.	According to Business information manager (9): *"our company goal is to use as much COTS as possible, all IS which could be replaced by a COTS, should be abandoned".*	Case E	X	X					I
37	Data is no longer needed.	According to IS owner (2): *"we kept data in IS and nobody knew why we kept this data; by abandoning this data we could abandon the IS".*	Case C	X			X			I
38	Momentum, opportunity for a new organization.	Momentum, opportunity for a new organization, a new way of working. According to CIO/ICT director (2): *"we were bought by an equity firm who were willing to invest in our organization. This was a perfect chance to have a greenfield approach and get rid of all the legacy IS".*	Case B	X	X	X	X	X	X	I

Table 61. Six new legacy IS abandonment triggers

All abandonment triggers are illustrated in Figure 28 (p. 123).

Appendix C.5 Legacy information system abandonment triggers glossary

In this section the legacy IS abandonment triggers are defined, these abandonment triggers are mentioned by interviewees and are categorized by means of coding. Concerning the granularity, a PhD dissertation is a perfect opportunity to describe in detail the legacy IS abandonment triggers. Therefore categorization is done at an elementary level. Based on this categorization other researchers might use a higher level categorization. For example: cost-benefit (No.1), high cost (No. 2) and high cost per user (No. 3) might all be categorized in a coarser granularity under cost-benefit. In Table 62 the glossary of the legacy IS abandonment triggers are described.

Legacy IS abandonment triggers.	**Remarks**	No. in Figure 28 (p. 123)
Cost-benefit.	When the business value of the process supported by the legacy IS, is lower than the cost of the legacy IS, the legacy IS should be abandoned.	1
High IS cost.	High legacy IS cost was a trigger to abandon a legacy IS. High legacy IS cost can be due to inefficient internal processes.	2
High cost per user or customer.	A legacy IS is built for 400,000 customers and only serving 4,000 customers; the depreciation, and maintenance costs are not recovered by fewer users.	3
Aged/old hardware.	A legacy IS which is part of an old infrastructure (old platform) or hardware (aged hardware/old infrastructure/old platform) or no longer runs on its (aging) hardware, taxing the limits of the legacy systems host computer, should be abandoned.	4
Inadequate software quality.	Due to ongoing software maintenance, the software structure deteriorates and maintenance becomes problematic.	5
New technological opportunities.	If newer technology is available or there are technological opportunities (e.g. new technology for application, hardware, data sets). There are newer technologies for lower prices.	6
Limited or ended supplier support.	A legacy IS should be abandoned if suppliers have limited or ended support (limited or no supplier support for application, hardware). This might be due to a bankruptcy, or a supplier who deliberately stops support to force transition to a new IS.	7
Redundant IS.	Redundant IS are legacy IS and should be abandoned; redundancy might be due to a proliferate landscape, no information planning in the past or a merger.	8
Merger or consolidation.	Merger or consolidation of similar organizations is causing redundant legacy IS (merger or consolidation).	9
Redundant application functionality.	Redundant functionality (redundant/double functionality; redundant application functionality) “Similar or double application functionality within an organization”: by means of plotting application functionality on business processes this redundancy is made visible.	10
Limited or no technical skills available.	Reduced availability of individuals with the required expertise; technical skills not available anymore.	11
Infrastructure overhaul.	Partial or complete disassembly of the infrastructure. This can be the result of a tender in which a new data center is selected, with different technology.	12
Application functionality no longer needed.	When the functionality of the application is no longer needed the IS can be abandoned.	13
Application functionality gap/not sufficient.	New functionality needed (functionality gap of application is so large that IS has to be abandoned). For example, the speed of changes is high, or the changes are so big that it is not possible to follow the demanded functionality. Functionality not sufficient. The gap between required functionality and delivered functionality is so large that the IS is abandoned.	14
Capability shortcomings.	When an IS is not or only partly capable of delivering the capability as required by an organization, there are system capability shortcomings. When the gap between business needed IS capability and supplying capability is growing and is too large, then the IS is candidate for abandoning.	15
Security issues.	When an IS is vulnerable to cyber-attacks, a decision to abandon an IS might be due to security issues.	16

Legal compliance (law).	By law, business functionality from organization went to IRS, making the IS redundant. Changing law by politicians: when by law a certain social security product line or service line within social security that the case organization is supplying to the clients is abandoned, the IS supplying this functionality is a legacy IS and should be abandoned".	17
Does not fit in technological strategy.	If an organization decides to make a reference architecture including a technology strategy, and the legacy IS does not fit in the technological strategy (application, hardware, data sets) the consequence is that the legacy IS should be abandoned.	18
Instability of the IS.	Instability of the legacy IS due to technical changes or a platform which is unstable resulting in random reboots, hangs and software crashes.	19
Non-scalability of the IS.	Non-scalability of the IS (application, hardware, data sets). Demand is rising, however due to technical constraints it is not possible to supply this demand. A large IS can handle the small additional volumes relatively easily, while a small IS might have difficulties to handle large amounts of extra volume.	20
Technical aging.	Technical aging (application, hardware, data sets, procedures): for example "no upgrades available anymore", "outdated procedures", "no support for the assembler anymore".	21
The IS is non-agile.	The organization is not able to quickly adapt the legacy IS to changing environmental circumstances.	22
Structural lack of IS maintenance.	A lack of structurally maintaining the system (application, hardware). If the maintenance of the IS is neglected for a long time, this will result in the trigger to abandoning the legacy IS (If you do not put oil in your engine, you have the guarantee that your car will stop one day).	23
Vendor dependency/vendor lock-in.	Dependency on vendors/vendor lock-in., e.g. intellectual property is sometimes unwanted, ultimately the legacy IS should be abandoned.	24
Effort to migrate applications from an old data center to a new data center.	Abandoning a legacy IS is a lot of work. There is a tendency to keep the legacy IS (for another year). However, it was found that due to migration pressure, which at least is similar work, more legacy IS were abandoned; this effect was named migration pressure.	25
Redundant data centers.	In case there are more data centers, this is a trigger to abandon redundant data centers.	26
Redundant data sources.	There were legacy IS using three different data sources; this was a trigger to abandon the legacy IS.	27
Perception of the required IS capability is changed.	Changing user perception, users nowadays have computers at home, their perception of an IS is changing, example IS with monochrome displays and without mouse functionality. Perception of the needed functionality is changed (by organization or by users or by customers).	28
Abandoning products or services supported by the IS.	Product or service abandoned which is supported by the IS (e.g. a decision of an organization to stop supplying letters on paper). There is no justification for the IS anymore, or there is another positioning of the organizational services.	29
Capability not needed.	The legacy IS was abandoned when the capability of the legacy IS was not needed (e.g. product or service delivered by the IS was discontinued). A legacy IS with no use at al.	30
New demands from the market, which cannot be provided by the IS.	If clients expect another service this might be a trigger to abandon a legacy IS. E.g. clients use to correspond in the past with organization with letters and organization scanned this letters into an IS, nowadays clients are communicating by email or phone and the IS providing the scanning functionality should be abandoned. The organization is working more and more in supply chains, there is the: "National Reference Architecture" it describes references and principles. It is used to govern that several different Government embodies are able to communicate and share information with each other. In case the data structure of an IS, or the communication protocol is not according to the National Reference Architecture and not working in the information chain the legacy IS might be abandoned. New demands from the market (external market environment) which could not be provisioned by the current IS, changing external market environment.	31
Complexity (+/-).	If a legacy IS is very complex, this might be an argument to abandon a legacy IS due to the fact that implementing changes in a complex legacy IS is risky. On the	32

	other hand, abandoning a complex legacy IS is more difficult and risky than a less complex legacy IS. If a legacy IS is very complex, this might be an argument not to abandon or to postpone the abandoning of a legacy IS. By maintaining the IS, the IS is getting more complex and therefore maintenance will become more difficult and more expensive. It was formulated by Furneaux and Wade (2011) as the opposite: low-levels of technical integration (low complexity) were key determinant of increased intentions to replace an existing system.	
Enormous proliferation of (unknown) information systems in unknown (different) places.	The number of application has increased beyond control; there is an enormous proliferation of (unknown) information systems in unknown (different) places.	33
Outsourcing, first clean your IS landscape.	Organizations who want to outsource, first have to rationalize their IS landscape. Outsourcing is a lot of work; if a lot of legacy IS are abandoned upfront, this saves a lot of resources.	34
Standardization.	A standardization is provided across the organization. All IS that do not conform to this standardization are abandoned (e.g. application no longer applies to standards). This looks similar to "does not fit in technological strategy". But there is a difference: the technological strategy might say "we use a Microsoft environment", whilst the standardization relates to the hardware platform, e.g. everybody has the same processor speed.	35
Internal directive COTS.	If the company goal is to use as much COTS as possible, all IS which could be replaced by a COTS, should be abandoned.	36
Data is no longer needed.	The organization kept data in IS and nobody knew why the organization kept this data. By abandoning this data the organization could abandon the legacy IS.	37
Momentum, opportunity for a new organization.	Momentum, opportunity for a new organization, a new way of working.	38

Table 62. Legacy IS abandonment triggers glossary

APPENDIX D LEGACY IS DISPOSAL PLAN

In this appendix a retirement checklist and legacy IS abandonment disposal plan are presented. The plan is based on the case study research (all cases) and appropriate literature. When the legacy IS are practically abandoned, a disposal plan should be written (Rittinghouse, 2004). Examples of such plans are provided by the U.S. Department of Justice, the department of justice systems development life cycle guidance document, chapter 12 (January 2003), www.justice.gov/archive/jmd/irm/lifecycle/ch12.htm (accessed December 2014). Another disposal plan (a template, a checklist and a practice guide) is provided by the U.S. department of Health and Human services www.hhs.gov/ocio/eplc/Enterprise%20Performance%20Lifecycle%20Artifacts/eplc_artifacts.html (accessed December 2014). In Table 63 a legacy IS disposal plan is presented:

Checklist:
Complete and signed filled legacy IS abandonment disposal plan.
Communicate abandoning of legacy IS to stakeholders.
End contracts with suppliers.
Data strategy (dispose, accessible, back up) signature data officer.
User free statement by IS owner (e.g. director, business unit director).
Recovery plan available.
Remove all owner access to IS.
Freezing period of 4 weeks.
Dispose legacy IS.
Notify the stakeholders that the legacy IS is abandoned.
Archive life cycle deliverables (manuals, documentation and source code).
Update legacy IS attributes database.

Table 63 provides a template of a legacy IS disposal plan.

<table>
<tr><td colspan="4">Application name:</td></tr>
<tr><td colspan="4">Description application:</td></tr>
<tr><td colspan="2">Application inception date:</td><td>Application abandoning date:</td><td>Contact person
Phone number:</td></tr>
<tr><td colspan="4">Reason legacy IS abandonment:</td></tr>
<tr><td>Software disposal.</td><td colspan="3">Archiving, deleting software (e.g. source code) and manuals.</td></tr>
<tr><td>Hardware disposal.</td><td colspan="3">Disposing or transferring hardware.</td></tr>
<tr><td rowspan="4">Data disposal.

Signature data officer:</td><td>1</td><td colspan="2">No data storage.</td></tr>
<tr><td>2</td><td colspan="2">Data stored on CD or database.</td></tr>
<tr><td>3</td><td colspan="2">Data accessible by web functionality.</td></tr>
<tr><td>4</td><td colspan="2">IS downsized to a small server providing some accessible functionality.</td></tr>
<tr><td colspan="4">Name IS owner (e.g. director, business unit director):</td></tr>
<tr><td colspan="4">Signature IS owner:</td></tr>
<tr><td colspan="4">Date signature:</td></tr>
</table>

Table 63. Legacy IS disposal plan

APPENDIX E EVALUATION TRAIL 1 AND TRAIL 2

Evaluation of the merger of the six information functions (Trail 1)
Case A was incepted by law, which caused six similar branch organizations to consolidate. This consolidation included the merger of the six information functions in the new Case A business units, which resulted in substantial cost savings. In the business unit WW the merger of the six information functions took six years. During this research, there was a focus on the subcase merger of information function Government and Education (G&E). This is evaluated in the following section.

Evaluation of the merger of information function Government and Education (G&E)
In this section - based on the interviews - a qualitative evaluation is described from a G&E stakeholder perspective. The G&E case was the smallest of the IS in the WW business unit. Although it was the smallest, it was however the most difficult to merge due to the fact that the population (clients) of this information function was the population of G&E which had different unemployment laws, e.g. every four weeks payments instead of every month. Also the employer was directly responsible for the unemployment wages.

In the beginning of 2002 it was decided to merge all the information functions from former branch organizations (organization 2 - organization 6) to one target information function (organization 1). On July 2007 the planning was presented of how to merge the last G&E information function (organization 6) including the main IS: “ABC” to the information function of organization 1 with the main IS: “XYZ”. The “XYZ” IS was adapted to receive these clients from G&E; clients were informed about their changing payments and the G&E information function was abandoned in 2008 as the last information function that merged to the information function of organization 1.

Although the G&E information function including the IS underneath was by far the smallest, it was more sophisticated. The G&E information function carried an electronic archive, which made it possible to work paperless and it was easy to look at files of clients in other geographical locations. This was functionality that the target information function organization 1 was lacking. G&E information function was developed using latest development environments. This was a “to be” situation for the Case A organization. However, based on workload, G&E covering 5% of the total volume, it was easier to integrate into a system already covering 95% of the total volume, compared to the opposite which was transferring 95% of the volumes to a system covering only 5%. This might be very risky: can a “small” system handle this enormous extra load? Also, from a cost perspective training and migrating 5% is cheaper than training and migrating 95% of the population. Researcher was able to interview management responsible for this decision and management and users of the G&E information function affected by this abandonment decision. This resulted in two different views, all describing their own reality.

The view of the “former” G&E IS owner:

The view of the former IS owner was a rational one. Merging and abandoning this information function G&E with the other bigger information function organization 1 would result in one uniform information function with one standardized working process, with lower costs for IS, due to the fact that one IS was abandoned. When the decision was taken to abandon the G&E information function, the maintenance budget for this information function was cut; only maintenance due to implementing new law which was mandatory by the Government was allowed. The goal of the merger was to end up with one information function. This was the ultimate goal of the merger and this goal was consistently kept in view. This was done with a social HRM programme to help employees affected by this decision. Success factors mentioned by the IS owner (business unit director) and his team was: a good project team, clear communication, a good social programme, sticking to the plan, bringing the system to the grave with a party and no discussing the quality of the former information function and the “to be” information function.

The view of the G&E employees:

For years the G&E information function had their own special G&E clients, with their own specific laws and their self-developed: "state of the art" information function. They were only a small department and they were situated in a provincial town whilst headquarters were in the capital city a few hundred kilometers away. Although they were only small, covering 5% of the population, their working concept including an electronic file; a paperless office could be the future way of working for Case A (the other information functions were lacking this functionality).

It was, however, decided at the headquarters of the business unit that their information function should be abandoned. The staff of the G&E information function provided headquarters with many arguments as to why the G&E information function should not be abandoned. Headquarters however did not change their plan and even decided that no more new maintenance on the system was allowed. In the beginning the management of the G&E information function tried to persuade headquarters management with business cases, even with very short pay back periods to have maintenance carried out. However, headquarters management did not change their minds, maintenance on the system was not allowed. Employees came to local management of the G&E information function to tell about their ideas about improving the functionality of the system. The responsible Business operations manager (1) said: "*I told these employees, this is a good idea, however I am not going with this plan for improving the functionality to headquarters because I know that we are not allowed to do any maintenance to improve the information function except for necessary maintenance for implementing new laws*". Management of G&E information function accepted this situation and managed this situation professionally. The G&E information function should be abandoned.

Management also though that the employees were accepting the situation. However, management noticed after some time that if things went wrong and if management was asking why management was not informed about this situation to take action, the answer was: "*well I didn't tell because it does not matter, there is no money for changes in the information function. Management is not going to do anything with this*".

When finally the plan to actually abandon the IS was presented and technical staff of the G&E information function were asked to abandon their own system by migrating data from the old information function to the new information function, management of the G&E information function discovered that at this moment of truth there were still a lot of emotions with these employees. Although these emotions were there, these people worked professionally to abandon their proprietary (own build) information function and to merge it to the organization 1 information function.

Case A offered them a job in the headquarters ICT department or a social package to leave the organization. Working in headquarters would in practice have meant moving to or close to the capital due to the distance (> 200 km); also houses were more expensive in or near the capital. Because Case A could not give them job security, (Case A can give nobody job security) this offer was not attractive to the employees in the provincial town and nobody went to headquarters.

The Business operations manager (1) who was responsible to migrate the operational employees from the "old" functionally superior G&E information function to the "new, functionally lacking, character based information function of organization 1" said: *"there was training for these employees and the trainer complained that instead of learning the new information function, for the first two days he only heard: this is bad, that was better in our former information function, it was already in our former information function, can this information function not do so and so"*. User (2) who was migrated from the "old" superior information function to the "new" functionally lacking information function said: *"I was back in the stone age".*

After the merger to the organization 1 information function, when the responsible Business operations manager (1) was discussing production figures with his team, he asked his management why the production was so low

and the staff responded with: "*what do you think with a system like this*". So in this first period the system was blamed for everything.

Evaluation and results "Abandonment and Migration" (Trail 2)
In Case A, a special Programme: "Abandonment and Migration" (A&M programme) was incepted after tendering the new data center, meaning four data centers had to be abandoned and 1 new data center was incepted. The programme surveyed all applications, asking the business what the strategy was with each application (abandon or migration). Then the programme decided a strategy on which applications were first abandoned or migrated. The decision to abandon an application was made in the business. The service for abandoning an application or migration of an application to the new data center was performed by the A&M programme for the IS owner (e.g. director, business unit director).

When the A&M programme started (2006), most of the applications were labeled: 410 migration versus 260 abandoning. However, after 2 years it was the opposite: 193 were labeled migration and 471 were labeled abandoning (see Figure 22, p. 51). Several explanations by interviewees were given for this fact. It seemed that migration of an IS was a lot of work for the business and due to the fact that they were forced to migrate the IS, the alternative - abandoning the IS - was a better alternative. In the past such an application just remained at the current data center and the contract was renewed for another year. There was a time aspect: because it was 2 years later, new insights were available. Applications which in the beginning were a question mark (for example because not all connections to other IS were known), were better known and the impact of abandoning was better known. The business already abandoned more applications, suggesting a learning effect. According to Abandonment and migration project manager (1) this was a human aspect: *"if you buy a new TV, you move your old TV to a shed because it was still working. After two years you decide to abandon the TV".*

The A&M programme practically abandoned hundreds of applications which were hosted, maintained and developed by external suppliers (these suppliers in a lot of the cases were former colleagues of the Case A population and were separated at the inception of Case A).

By abandoning or migrating an application, revenue was lost for these suppliers and the A&M programme experienced heavy resistance to their programme from suppliers and also from internal Case A colleagues, who would sometimes lose their job, their system or their contacts with their former colleagues at external suppliers. According to Abandonment and migration project manager (1): *"the whole human spectrum of resistance came along: there is no documentation, we don't know what is going to happen, we first have to make the documentation, keep asking questions, can you explain again what you mean".*

According to Abandonment and migration project manager (2): *"all objections against the programme were aired to everybody who wanted to hear, high and low in the organization".* Complaints that the A&M programme was not well managed, was not well governed and had a wrong strategy. According to some interviewees it seemed that there was a group of people inside Case A and inside the supplier who worked together against the Case A A&M programme. This could be explained by the fact that some of these people had already worked with each other for 20 years and had been colleagues in the past. Researcher asked how the programme managed these aspects and kept the programme going. Abandonment and migration project manager (1) replied: *"with everything I had, from power skills, charming, humor, anger, checking and phoning".* Abandonment and migration project *manager (1) said: "abandoning IS is a process of mourning".*

APPENDIX F INTERVIEW WITH A RESPONSIBLE CIO

The following section includes a part of an interview with a CIO, responsible for the abandoning of legacy IS.

"In my 20 years as CIO, I always saw a growing IS landscape, there is always inception and never abandoning. The problem is the cost of operating and maintaining. This IS landscape will become every year more and more expensive. I saw the consolidation of Case B as an enormous opportunity to get rid of all the legacy IS. I told my colleagues within the management team, that it should be tough. However, I told them that it should be worth it, if we had a new landscape without the legacy IS in 2 years' time. The colleagues supported me and that was very important, tone at the top.

There were difficult moments, but we managed it and the colleagues (board, directors level) kept supporting me, despite de problems we faced. My colleagues knew that all the IT spent on IS maintenance could not be spent by them. They also didn't want to have a growing and money consuming IS spaghetti landscape. So we rationalized data centers, applications, locations, buildings, suppliers, standards. Everything that was not within the standard was abandoned.

For me it was an advantage that I was new to the organization and had not been employed in one of the former organizations. I was deemed value free.

As an example one of the former predecessor organization had IBM and another predecessor organization had HP, I did not have any emotional connection with it. I had to decide what to take; the decision I took, was considered a rational decision and not an emotional decision. If you look at other organizations, where the CIO has already been responsible for years for four data centers, 12 CRM systems and such an CIO is asked to rationalize, it is difficult. It is asking the CIO to amputate his own legs.

If we were abandoning a system, we would not invest in maintenance, due to the fact that it is costly, risky for the stability of the system and it makes abandoning even more difficult. The problem with abandoning legacy IS was that they have cost a lot of money in the past and that there was no trade-in value; the rest value was zero. That hurts, but you had, a lot of benefits in the past from this IS.

What was important was preparation, we did not have enough preparation time, due to the time constraint given by the management. Asset management was very important: which applications and which contracts did we have. I met with suppliers on the 2nd of January, they told me: "congratulations Mr. CIO, you are a customer of mine, look here is the signed contract of your predecessor. Can we send the bill to you, because you did not end the contract before 31 December?"

We were persevering in abandoning legacy IS (applications). Daily people in the organization were shouting that nasty things would happen with our organization if we abandoned certain legacy IS. I invited them to my office and then most of the time it was a non-issue.

We had two large finance packages: "package A" and "package B". We made a decision for one financial package A. Internal stakeholders from package B were shouting "this is not possible in package A", and then we invited the supplier of package A and they explained that it was also possible in package A.

We never acted on rumors, people with worries concerning abandoning were invited to fill in a business case kind of template in which they described why the legacy IS should stay and not be abandoned. Doubt was killing, If one of the legacy IS to be abandoned slipped, hundreds would follow, therefore we persevered till the end. If somebody had a good story not to abandon a legacy IS, we solved the problem. We found an isolated server somewhere in Groningen with some software on it, it was our intention to abandon this legacy IS. However this

system was used for getting subsidiary funds from Europe, it earned a lot of money, so we solved it in a correct manner.

It was also our opinion as management that if we wanted to be a new organization there should be as little as possible from the past. That's why we moved headquarters to a city in the center of the country. I also abandoned suppliers which I found to be no partners in this consolidation. They had too much of a profit and loss mentality; when it started to be difficult they were getting out the contracts and were reading the small letters instead of solving the problems. I abandoned a very large consultancy company".

11 GLOSSARY

Abandoning a legacy IS, a process of removing a current legacy IS constellation, a combination of application software, hardware, data sets, people and procedures.

Abandonment and Migration (A&M) Programme, A programme in Case A, responsible for practically abandoning legacy IS and migrating IS to a new data center.

Accountable, the main difference between responsibility and accountability is that responsibility can be shared while accountability cannot. Being accountable not only means being responsible for something but also ultimately being answerable for your actions.

Act of choice, the actual choice between different alternatives or courses of action taken by the appropriate authority in an organization.

Alternative, a possibility one can choose from; alternatives can be identified (that is, searched for and located) or even developed (created where they did not previously exist).

Agile, in response to the rapid changes in users' requirements, a new generation of information systems (IS), namely, agile IS, has emerged. Agile IS, defined as IS developed using agile methods, are characterized by frequent upgrades with a small number of new features released periodically.

Aging characteristics of legacy IS, characteristics which describe the aging of legacy IS, these characteristics are further categorized into internal and external legacy IS aging factors.

Aging factors of legacy IS, are aging characteristics categorized into internal and external legacy IS aging factors.

Application functionality, a useful function within a computer application or program.

Application functionality gap, the difference between required functionality of the application by users and the actual provided functionality of the application to the users.

Application life cycle management (ALM), is the continuous process of managing the life of an application from its initial planning through retirement. It also refers to how changes to an application are documented and tracked.

Application Portfolio Management (APM), is the set of activities that documents and drives how an organization measures and responds to the business value, cost, performance and risk of its portfolio of application assets".

Application-related legacy characteristics, these are legacy characteristics specifically related to the individual application-related component of the IS, for example: during the maintenance the application software will structurally deteriorate. Legacy application software is characterized by inadequate software quality, this can happen for example that, due to time pressure new functionality is added and the software is not programmed in a structured way.

Application software, a program or group of programs designed for customers or end-users; figuratively speaking, applications software sits on top of system software because it is unable to run without the operating system and system utilities. It is defined as a logical entity that provides the business services and which is usually composed of a number of separate programs that have been developed at different times.

Artifact, a person-made object.

Architect, (Enterprise or domain architect) work with stakeholders, both leadership and subject matter experts, to build a holistic view of the organizations' strategy, processes, information, and information technology assets. The role of the enterprise architect is to take this knowledge and ensure that the business and IT are in alignment. The architect links the business mission, strategy, and processes of an organization to its IT strategy, and documents this using multiple architectural models or views that show how the current and future needs of an organization will be met in an efficient, sustainable, agile, and adaptable manner.

Automate, subscribes to refinement and rationalization; the substitution of manual work, often of transactional character, by computerized information technology. It is related to efficiency, i.e. achieving better performance by conducting current work practices the best possible way.

Availability, (application) is the extent to which an application is operational, functional and usable for completing or fulfilling a user's or business's requirements.

Axial Coding, extends the work from initial coding, the purpose is to reassemble data that were "split" or "fractured" during the initial coding process. Axial coding is the process of relating categories to their subcategories.

Benefit, the economic value of a scheme, usually measured in terms of the cost of damages avoided by the scheme, or the valuation of perceived amenity or environmental improvements (not necessarily expressed in a monetary value.

Business information manager, (also referred to as Information Manager) is positioned in an IT supply demand model as part of the user organization and is responsible for the total information function in a business unit, the automated and the non-automated part.

Business information administrator, provides support to business information users.

Bounded rationality, limitations on decision making caused by costs, human abilities, time, technology, and availability of information.

Business processes, the processes that are used in the business to achieve some business objective.

Capability of an IS, versus functionality of an application.

Case study, an empirical enquiry that investigates a contemporary phenomenon within its real life context, when the boundaries between phenomenon and context are not clearly evident.

CEO, a Chief Executive Officer (CEO in American English) or managing director (MD in British English describes the position of the most senior corporate officer, executive, or administrator in charge of managing a non-profit or for-profit organization. The CEO of a corporation or company typically reports to the board of directors and is charged with maximizing the value of the entity. Titles also often given to the holder of the CEO position include president and chief executive (CE).

CIO, a Chief Information Officer (CIO) or Information Technology (IT) Director, is a job title commonly given to the most senior executive in an enterprise responsible for the information technology and computer systems that support enterprise goals. Generally, the CIO reports to the chief executive officer, chief operating officer or chief financial officer.

CFO, Chief Financial Officer (CFO) or chief financial and operating officer (CFOO) is a corporate officer primarily responsible for managing the financial risks of the corporation. This officer is also responsible for financial

planning and record-keeping, as well as financial reporting to higher management. In some sectors the CFO is also responsible for analysis of data. The title is equivalent to finance director (FD), a common title in the United Kingdom. The CFO typically reports to the chief executive officer and to the board of directors, and may additionally sit on the board. The CFO supervises the finance unit and is the chief financial spokesperson for the organization. The CFO reports directly to the president/chief executive officer (CEO) and directly assists the chief operating officer (COO) on all strategic and tactical matters as they relate to budget management, cost–benefit analysis, forecasting needs and the securing of new funding.

Client, a person who receives services, in this dissertation client and customer are used interchangeably (a customer, who can, however, not choose for the supplier, typical the relation between government and citizens).

Complexity, is the state of having many different parts connected or related to each other in a complicated way and being difficult to understand or find an answer to. Furthermore, complexity is subjective, it has to do with the ability of an individual to abstractly understand how the components work together to function as a more complex assembly.

Computer aided software engineering (CASE), software tools used to design and implement applications.

Configuration Management Database (CMDB), is a repository that acts as a data warehouse for information technology (IT) organizations. Its contents are intended to hold a collection of IT assets that are commonly referred to as configuration items (CI), as well as descriptive relationships between such assets.

Consolidation, the combining of assets, liabilities and other financial items of two or more entities into one. In the context of financial accounting, the term consolidate often refers to the consolidation of financial statements, where all subsidiaries report under the umbrella of a parent company. These statements are called consolidated financial statements. Consolidation also refers to the merger and acquisition of smaller companies into larger companies. A consolidation, however, differs from a merger in that the consolidated companies could also result in a new entity, whereas in a merger one company absorbs the other and remains in existence while the other is dissolved.

COO, Chief Operating Officer, also called the chief operations officer, director of operations, or operations director, is a position that can be one of the highest-ranking executive positions in an organization on par with the Chief Services Officer, comprising part of the "C-Suite". The COO is responsible for the daily operation of the company.

Cost, the economic value of the amount of resources applied to a cost object (not necessarily expressed in a monetary value).

Cost-benefit analysis, a method of appraisal that assesses different alternatives through a comparison between their costs and benefits.

COTS (commercial off-the-shelf), an adjective that describes software or hardware products that are ready-made and available for sale to the general public.

Customer, person who buys goods or a service (In this dissertation customer and client are used interchangeably).

CRM (Customer relationship management), a system is a tool for collecting and managing the information and interactions your business has with your customers, sales leads, suppliers or other businesses.

Data, is discrete objective facts about events. Data are transformed into information by adding value through context, categorization, calculations, corrections and condensation.

Data base, an organized collection of data. It is the collection of schemas, tables, queries, reports, views, and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information.

Data-related legacy characteristics, these are legacy characteristics specifically related to the individual data-related component of the IS, for example: organizations are amassing and storing more data than ever before, the amount of data in our world has been exploding. Legacy data characteristics are an immense volume of data which has accumulated over the lifetime of the system, this data may be inconsistent and may be duplicated in several files.

Data set, a collection of related information made up of separate elements that can be treated as a unit in data handling.

Decision, a moment in an ongoing process of evaluating alternatives for meeting an objective, at which expectations about a particular course of action impel the decision-maker(s) to select that course of action most likely to result in attaining the objective.

Decision making process, the choice process in which one among several alternatives or courses of action is selected.

Disinvestment, the elimination of a major investment.

Disposal, the orderly termination of the system.

Documentary information department, department responsible to assure that assure that organization documentary records are kept according to legal compliancy

Economic obsolescence, reduction in the desirability or economic life of an asset caused by factors such as regulatory changes, technological changes and excess supply.

Fall back, a contingency option to be taken if the preferred choice is unavailable.

Function point, a unit of measurement to express the amount of business functionality an IS (as a product) provides to a user. Function points measure software size. The cost (in dollars or hours) of a single unit is calculated from past projects.

Functional requirement, a requirement that specifies a function that a system or system component must be able to perform (in this study, the system considered is the software product).

Guideline, a recommended practice that provides guidance to appropriate behavior.

Hardware, the physical equipment used for input, processing and output activities in an IS, usually the mainframes or servers, where the application runs on.

Hardware-related legacy characteristics, these are legacy characteristics specifically related to the individual hardware-related component of the IS, for example: Computer hardware has become infinitely more powerful through the years, resulting in an exponential growth of computing performance. Aging legacy hardware, however, is characterized by slow or inadequate performance.

Hierarchical database, a data model in which the data is organized into a tree-like structure. The data is stored as records which are connected to one another through links. A record is a collection of fields, with each field containing only one value. The entity type of a record defines which fields the record contains.

Human Resource Management (HRM), the process of hiring and developing employees so that they become more valuable to the organization. Human Resource Management includes conducting job analyses, planning personnel needs, recruiting the right people for the job, orienting and training, managing wages and salaries, providing benefits and incentives, evaluating performance, resolving disputes, and communicating with all employees at all levels.

ICT domains (also named logical ICT domains), in this dissertation telephone, WAN, LAN, distributed computing (pc and printing), software, application development, application maintenance and data center.

Incremental Budgeting, always begins with the budget from the last period. Once there is an established starting point, if a department needs more money than the previous budget, they have to be able to justify the extra expenses. Also, if you do not use your budget, then the next period's budget will be reduced. This type of budgeting often leads to wasteful spending by employees because they do not want to lose their budget.

Information Communication Technology (ICT), are the applications, the hardware and the data technology (IT) and added are the communication components, Telecom, WAN, LAN. ICT is therefore a sub set of IS. ICT and IT are often used synonymously, however ICT stresses the role of unified communication and the integration of telecommunications. In this dissertation they are used interchangeable.

Information, data that has been interpreted, translated, or transformed to reveal the underlying meaning.

Information function, are all the IS within a business unit.

Information provisioning, are all the information functions within an organization, it consist therefore of all the IS in an organization.

Information systems (IS), logical entities comprised of hardware, software, data sets, people and procedures.

Information system life cycle, the life cycle of an IS starts with development and implementation, during development there might already be some external aging, for example, due to the fact that the technology development in the real world proceeds. After implementation of the IS, the IS is operated and maintained, during this maintenance the IS will age due to accumulation of internal and external aging factors and becomes a legacy IS (a legacy IS is defined as any IS that significantly resists meeting organizations' requirements). During maintenance and operation of the legacy IS, there are constantly opportunities to decide to continue or discontinue the legacy IS (Sommerville, 2007), this is illustrated in Section 3.6, p. 34. A decision to continue the legacy IS means continuing with regular legacy IS maintenance in which functionality is added or re-engineering of the legacy IS in which the structure of the system is improved to make maintenance in the future easier and cheaper. A decision to abandon a legacy IS may result in the abandoning and scrapping of the legacy IS, referred to as decommissioning by Warren (1999), or the decision to abandon and replace all or parts of the legacy IS with a new system developed from scratch. At some point of time within this conceptual model, the aging crosses a certain legacy IS aging threshold, which is a trigger for decision makers to decide not to maintain a legacy IS anymore, but to re-engineer the legacy IS. Re-engineering is often incorrectly treated as being synonymous with software engineering, its true scope is broader, although the software is often the focal point of re-engineering efforts and it alone is only one of the components of the complete system that must be considered when re-engineering (Tilley & Smith, 1995). The salient difference between engineering (development) and re-engineering is that the legacy system already exists compared to engineering a new system (development) which have to be incepted first (Tilley & Smith, 1995). Re-engineering is considered to

again lower the internal aging factors. After re-engineering the legacy IS, the IS is maintained again and the internal and external legacy IS aging factors are growing again. This process continues until there is a moment that it is not beneficial anymore to maintain or re-engineer the legacy IS, this moment is attained when the external aging factors passes the legacy IS aging threshold. Or when a legacy IS abandonment trigger appears. The legacy IS should be abandoned, possibly be replaced by a new IS. If the decision has been taken to abandon a legacy IS, it takes some time to practically abandon a legacy IS, during this time no maintenance should be performed anymore and the legacy IS aging still continues.

Information system-related legacy characteristics, these are legacy characteristics related to the IS as an entity, for example: the IS is expensive, means that the combination of the individual components (application, hardware, data sets, procedures, people) makes the IS expensive. Unclear is whether it is for example the application, the hardware or the staff of a combination of these components. Examples are: legacy IS are no longer meeting the needs of the environment or are significantly resisting modification and evolution in order to meet new and constantly changing business requirements.

Information Technology (IT), are the applications, the hardware and the data technology and are therefore a sub set of IS.

Infrastructure, consist of hardware, software and network facilities.

Instability of the IS, random reboots, hangs, random software crashes.

Interface, a shared boundary across which two separate components of a computer system exchange information. The exchange can be between software, computer hardware, peripheral devices, humans and combinations of these. Some computer hardware devices such as a touchscreen can both send and receive data through the interface, while others such as a mouse, microphone or joystick are one way only.

Internal Revenue Service (IRS), is responsible for collecting taxes.

IS attribute database, consists of Generic IS information [Application number; Application name; Description (including owner); Age (years); Function points; (C)OTS/(P)roprietary; (P)rimary/(S)ecundary; Program language; Maintenance cost; Supplier; Hardware platform], Business value [Number of users; Business importance; Required lifetime] and Legacy IS abandonment triggers [High IS cost; Merger or consolidation; Abandoning products or services supported by the IS; Aged/old hardware; Redundant application functionality; New technological opportunities; Limited or ended supplier support; Application functionality gap/not sufficient; Redundant data centers; Redundant IS; High cost per user or customer; Non-scalability of the IS; Inadequate software quality; Application functionality no longer needed; Capability shortcomings; Legal compliance (law); Does not fit in technological strategy; Instability of the IS; Technical aging; The IS is non-agile; Structural lack of IS maintenance; Vendor dependency/vendor lock-in; Effort to migrate applications from an old data center to a new data center; Redundant data sources; Perception of the required IS capability is changed; Capability not needed; New demands from the market, which cannot be provided by the IS; Infrastructure overhaul; Limited or no technical skills available; Cost-benefit; Security issues; Complexity (+/-); Enormous proliferation of (unknown) information systems in unknown (different) places; Outsourcing, first clean your IS landscape].

IS owner, owner of the information system, typically a director or a business unit director.

IT asset management repository (ITAM), stores data about all known enterprise assets, including IT assets, and represents the most current information that exists about the IT assets present within the enterprise.

Large organization, more than 1000 employees.

Legacy IS characteristics, legacy characteristics of the legacy IS, or its individual constituents (application, hardware, data sets, procedures, people) it is assumed that (legacy) IS gradually contain more legacy characteristics, making an IS a legacy IS. The legacy IS characteristics are further categorized into internal and external aging factors. External aging factors are considered to be out of the control of organizations, while internal aging factors are within the control of organizations.

Legacy IS aging factors, are aging characteristics of legacy IS (IS-related legacy characteristics, application-related legacy characteristics, hardware-related legacy characteristics, data-related legacy characteristics, people-related legacy characteristics and procedure-related legacy characteristics), which are categorized into internal or external aging factors.

Legacy IS abandonment triggers, are triggers which makes organizations decide to abandon a legacy IS.

Legacy IS, any IS that significantly resists meeting organizations' requirements.

Lumpsum contract, (or stipulated sum contract) is the traditional means of procuring construction, and still the most common form of construction contract. Under a lump sum contract, a single 'lump sum' price for all of the works is agreed before the works begin.

Maintenance, refers to the changes made to a system to fix or enhance its functionality.

Maintainability, refers to the ease with which software can be understood, corrected, adapted, and enhanced.

Practical abandoning, the orderly termination of the system.

Procedure-related legacy characteristics, these are legacy characteristics specifically related to the individual procedure-related component of the IS, for example: do not take advantage of streamlined facilities and standard routines; use programming and design practices that current management would not permit.

People-related legacy characteristics, these are legacy characteristics specifically related to the individual people-related component of the IS, for example: skills which are hard to find, retiring staff, or staff lacking technical expertise.

Merger, one corporation, known as the survivor, takes over another corporation, known as the merged. A consolidation is the joining of two corporations to form a new, third corporation.

Method, a coherent set of, instructions, techniques, guidelines, or practices, that prescribe how somebody who is willing to follow them should continue, and under what circumstances.

Methodology, the study of how research is done, how we find out about things, and how knowledge is gained. In other words, methodology is about the principles that guide our research practices. Methodology therefore explains why we're using certain methods or tools in our research.

Migration, the process of moving applications from one environment to another environment.

Mid-size organization, between 100 and 1000 employees.

MIP, Millions instructions per second.

Non-functional requirement, a product property that puts one or more constraints upon one or more functional requirements (examples in this study: reliability and maintainability); non-functional requirements include the compliance to relevant standards.

Organization, a social entity that is goal-directed and is a deliberately structured activity system with an identifiable boundary.

Platform, a computer platform generally means the operating system and computer hardware only.

Process, a set of instructions that defines a path to accomplish a predetermined objective.

Procedures, a specified way to carry out an activity or a process, procedures are part of a process. Since each process requires a set of procedures, there are a number of procedures within a process.

Programme, a coherent set of one or more projects to be carried out to attain long-term organizational goals, defined in the business strategy project - an intervention that consists of a set of planned, interrelated activities designed to achieve defined project objectives within a given budget and a specified period of time.

Project Portfolio Management (PPM), is the centralized management of the processes, methods, and technologies used by project managers and project management offices (PMOs) to analyze and collectively manage current or proposed projects based on numerous key characteristics. The objectives of PPM are to determine the optimal resource mix for delivery and to schedule activities to best achieve an organizations' operational and financial goals — while honoring constraints imposed by customers, strategic objectives, or external real-world factors.

RACI (Responsible, Accountable, Communicated, Informed) model, a straightforward tool used for identifying roles and responsibilities and avoiding confusion over those roles and responsibilities during a project**.**

Release, the distribution of the final version of an application. A software release may be either public or private and generally constitutes the initial generation of a new or upgraded application. A release is preceded by the distribution of alpha and then beta versions of the software.

Release management, the process of managing, planning, scheduling and controlling a software build through different stages and environments; including testing and deploying software releases**.**

Relational database, is a digital database whose organization is based on the relational model of data, as proposed by Codd in 1970. This model organizes data into one or more tables (or "relations") of rows and columns, with a unique key for each row. Generally, each entity type described in a database has its own table, the rows representing instances of that type of entity and the columns representing values attributed to that instance. Because each row in a table has its own unique key, rows in a table can be linked to rows in other tables by storing the unique key of the row to which it should be linked (where such unique key is known as a "foreign key"). Codd (1970) showed that data relationships of arbitrary complexity can be represented using this simple set of concepts.

Reliability, the probability that a product will operate without failure under given conditions for a given time interval (in this study: software product).

Research goal, the purpose toward which an endeavor is directed, the research goal may not be strictly measurable or tangible. Goals are broader than objectives in the sense that goals are general intentions and are not specific enough to be measured.

Research objective, efforts or actions are intended to attain or accomplish a purpose or a target. The research objective is measurable and tangible.

Risk, a probability or threat of damage, liability, loss, or any other negative occurrence that is caused by external or internal vulnerabilities.

Scalability, is the ability of a computer application or product (hardware or software) to continue to function well when it (or its context) is changed in size or volume in order to meet a user need. Typically, the rescaling is to a larger size or volume. The rescaling can be of the product itself (for example, a line of computer systems of different sizes in terms of storage, RAM, and so forth) or in the scalable object's movement to a new context (for example, a new operating system).

Service oriented architecture (SOA), is an architectural pattern in computer software design in which application components provide services to other components via a communications protocol, typically over a network. The principles of service-orientation are independent of any vendor, product or technology.

Small organization, less than 100 employees.

Social system, a set of interrelated units that are engaged in joint problem-solving to accomplish a common (set of) goal(s).

Socio-technical system, a system including hardware and software components that has defined operational processes followed by human operators and that operates within an organization. It is therefore influenced by organizational policies, procedures and structures.

Software (product), the complete set, or any of the individual items of the set, of computer programs, procedures, and associated documentation and data designated for delivery to a customer or end-user (IEEE 1990) (in this study: application software); a software product can be either a standalone software application or software embedded in a system.

Software, the collection of all application software also referred to as application and the support software also referred to as platform software such as operating systems.

Software maintenance, a set of activities that results in a newer version of an existing (software) product and the exploitation of that (software) product after its release; major revisions of an existing (software) product are accomplished through (software) development.

Stakeholder, a person or constituency whose interests are at stake; being involved in a decision is not a prerequisite to be a stakeholder.

Silo concept, an insular management system incapable of reciprocal operation with other, related IS**.**

Stovepipe design, a pejorative term for a system that has the potential to share data or functionality with other systems but which does not. The term evokes the image of stovepipes rising above buildings, each functioning individually.

Structural coding, name of a method within first cycle coding. The coding method “acts as a labeling and indexing device, allowing researchers to quickly access data likely to be relevant to particular analysis from a larger data set”. Structural coding both codes and initially categorizes the data corpus.

System software, low-level programs that interact with the computer at a very basic level; this includes operating systems, compilers, and utilities for managing computer resources (see also application software).

TCO (Total Cost of ownership), a financial estimate intended to help buyers and owners determine the direct and indirect costs of a product or system.

Technical aging, (e.g. application, hardware, data sets, procedures) for example “no upgrades available anymore”, outdated procedures, “no support for the assembler anymore”.

Technique, a well-defined procedure used to accomplish a specific activity or task.

Theory, a plausible or scientifically acceptable general principle or body of principles offered to explain phenomena.

Threshold, the level or point at which you start to experience something, or at which something starts to happen.

Total legacy IS aging, the sum of (internal and external) legacy IS aging factors and (internal and external) legacy IS abandonment triggers.

Vendor lock in, also known as proprietary lock-in or customer lock-in, makes a customer dependent on a vendor for products and services, unable to use another vendor without substantial switching costs. Lock-in costs which create barriers to market entry may result in antitrust action against a monopoly.

Workaround, a method, sometimes used temporarily, for achieving a task or goal when the usual or planned method isn't working. In information technology, a workaround is often used to overcome hardware, programming, or communication problems. Once a problem is fixed, a workaround is usually abandoned.

Wrapping, a technique in which an interface is created around an existing piece of software, providing a new view of the software to external systems, objects, or users. Wrapping can be accomplished at multiple levels: around data, individual modules, subsystems or entire systems.

Zero-Based Budgeting, always begins the new budget from an established point of zero. Instead of starting off with last period's budget and adding or subtracting from it, you begin with zero, and then go through every expense that you will incur during the course of business. This budgeting method utilizes much more detail and makes everyone accountable for their necessary expenses. The biggest drawback with this method is that it requires much more work to implement this method and it is often unpopular with employees.

12 SAMENVATTING (SUMMARY IN DUTCH)

Inleiding
Dit proefschrift is erop gericht te komen tot een verbeterd begrip over legacy informatiesystemen (IS-en) en inzicht te krijgen in het proces om afscheid te nemen van legacy IS-en. Het proefschrift beschrijft het verouderingsproces van legacy IS-en, het besluitvormingsproces om afscheid te nemen van legacy IS-en en vervolgens het proces om in de praktijk het afscheid van legacy IS-en te realiseren. Tenslotte wordt een methode om afscheid te nemen van legacy IS-en voorgesteld.

Legacy IS-en hebben een grote impact op de maatschappij (de Amerikaanse overheid begrootte in 2015, 58 miljard dollar voor legacy IS-en [GAO, 2015, p. 47]). Verder blijkt dat er nog weinig onderzoek gedaan is naar het afscheid nemen van legacy IS-en (Furneaux, 2009; Furneaux & Wade, 2010, 2011; Swanson & Dans, 2000).

Een IS bestaat uit vijf componenten (applicatiesoftware, hardware, gegevens, mensen en procedures) en deze componenten hebben onderlinge relaties. IS-en en de onderliggende componenten zijn net als andere systemen onderhevig aan veroudering en worden op een gegeven moment aangeduid met de term legacy IS-en. Een legacy IS is in dit proefschrift gedefinieerd als een IS dat moeilijk is aan te passen aan de veranderende eisen die een organisatie aan het IS stelt (Brodie & Stonebraker, 1995). Uiteindelijk zal een organisatie besluiten om afscheid te nemen van een legacy IS en na dit besluit zal het afscheid ook in de praktijk gerealiseerd worden.

Om inzicht te krijgen in de veroudering van legacy IS-en, hoe organisaties legacy IS-en herkennen, hoe organisaties vervolgens besluiten afscheid te nemen van legacy IS-en en hoe organisaties dit afscheid van legacy IS-en in de praktijk realiseren is de volgende primaire onderzoeksvraag gesteld:

Hoe identificeren organisaties legacy IS-en en hoe managen ze het afscheid nemen van legacy IS-en?

Op basis van deze onderzoeksvraag wordt in Hoofdstuk 2 het onderzoeksontwerp beschreven. Deze bestaat uit keuzes ten aanzien van de: onderzoeksfilosofie, onderzoeksaanpak, onderzoekstrategie, methodologische keuze, tijdshorizon, technieken en procedures (Saunders et al., 2007). Vervolgens wordt het onderzoek gepresenteerd en wordt de rol van de onderzoeker in het onderzoek toegelicht.

In Hoofdstuk 3 is een systematische literatuurstudie beschreven. In deze literatuurstudie (in 11 toonaangevende IS journals) is op basis van kernwoorden systematisch naar factoren gezocht die antwoord geven op de vraag wat zijn legacy IS-en? hoe worden deze geïdentificeerd? hoe worden deze gemanaged? wat zijn de besluitvormingsopties? hoe wordt er besloten om afscheid te nemen van legacy IS-en en op basis waarvan? en hoe wordt vervolgens in de praktijk het afscheid van legacy IS-en gerealiseerd? De bronnen in de gevonden artikelen zijn verder gebruikt om aanvullende artikelen te vinden.

In Hoofdstuk 4 wordt de exploratieve Case A beschreven. Case A is een complexe publieke consolidatie van zes gelijksoortige brancheorganisaties, waarbij afscheid is genomen van honderden IS-en. Er zijn twee sporen van onderzoek. In spoor één is onderzoek gedaan naar de besluitvorming rondom het samenvoegen van zes informatiefuncties, waarbij afscheid is genomen van honderden legacy IS-en. In spoor twee is onderzoek gedaan naar het daadwerkelijk realiseren van het besluit om afscheid te nemen van legacy IS-en. In dit spoor is afscheid genomen van 471 applicaties en vier rekencentra. Dit onderzoek betrof een longitudinaal onderzoek van vijf jaar.

Ter ondersteuning van de primaire onderzoeksvraag worden in Hoofdstuk 5 onderzoeksvragen gedefinieerd. Deze onderzoeksvragen zijn gebaseerd op de life cycle van een IS (Berghout & Nijland, 2002).

De laatste fasen in de life cycle van legacy IS-en zijn, (1) *het verouderingsproces van legacy IS-en*, (2) *het besluitvormingsproces om afscheid te nemen van legacy IS-en*, en tenslotte (3) *het proces om in de praktijk het afscheid van legacy IS-en te realiseren*. Op basis van deze drie processen zijn de volgende drie onderzoeksvragen geformuleerd:

1) *Hoe verouderen legacy IS-en?*
2) *Hoe besluiten organisaties om afscheid te nemen van legacy IS-en?*
3) *Hoe realiseren organisaties in de praktijk het afscheid van legacy IS-en?*

De antwoorden op deze onderzoeksvragen geven invulling aan:

4) *Een methode om afscheid te nemen van legacy IS-en.*

Op basis van de onderzoeksresultaten uit Hoofdstuk 5, worden achtereenvolgens in Hoofdstuk 6, (1) het verouderingsproces van legacy IS-en, (2) het besluitvormingsproces om afscheid te nemen van legacy IS-en en (3) het proces om in de praktijk het afscheid van legacy IS-en te realiseren geconceptualiseerd. Tevens worden ontwerp dilemma's ten aanzien van (4) een methode om afscheid te nemen van legacy IS-en geformuleerd.

Daarna worden in Hoofdstuk 7 op basis van een viertal vervolg cases (Case B, C, D & E), de concepten (zoals geformuleerd in Hoofdstuk 6) gevalideerd. Case B is een private organisatie en het resultaat van een consolidatie van drie organisaties, er is afscheid genomen van 850 applicaties. Case C is een publieke organisatie die in het verleden 15 regionale bedrijfsafdelingen had, met ieder een eigen informatievoorziening. Er was een programma om het aantal applicaties te reduceren van 2200 naar 800. Case D is een publieke organisatie en nam afscheid van een groot Cobol gebaseerd legacy IS. Bij Case E (een publieke organisatie) werd er ten tijde van het onderzoek geen afscheid genomen van een legacy IS en is alleen het proces onderzocht.

Deze resultaten worden vervolgens in Hoofdstuk 8 gebruikt om een methode te ontwerpen om afscheid te nemen van legacy IS-en. Tenslotte worden in Hoofdstuk 9 conclusies en aanbevelingen voor vervolgonderzoek gegeven. Hierna worden de antwoorden op de onderzoeksvragen gegeven, de externe validiteit van het onderzoek wordt beschreven en tenslotte worden suggesties voor vervolgonderzoek geformuleerd.

Hoe verouderen legacy IS-en?

Uit de literatuur en uit het onderzoek bij vijf case-organisaties, die in totaal afscheid hebben genomen van 2.020 legacy IS-en, zijn 149 legacy karakteristieken van legacy IS-en geïdentificeerd. Deze legacy karakteristieken hebben betrekking op het IS of op de individuele componenten van het IS. Deze 149 legacy karakteristieken zijn verder gecategoriseerd in (interne) verouderingsfactoren die beïnvloedbaar zijn door een organisatie (115), externe verouderingsfactoren die niet beïnvloedbaar zijn door een organisatie (14) en verouderingsfactoren die deels beïnvloedbaar zijn door een organisatie (20).

IS-en krijgen legacy IS karakteristieken naarmate de tijd verstrijkt en de context verandert. Op enig moment wordt er dan aan een IS gerefereerd als een legacy IS, niemand kon echter precies vertellen wanneer dat zo was. Het omslagpunt betreft het punt waarop het significant moeilijker wordt om het IS aan te passen aan de veranderende eisen die een organisatie aan dit IS stelt. Een tentatief verouderingsmodel is geïllustreerd in Figuur 23 (p. 89). De figuur illustreert hoe legacy IS verouderingsfactoren accumuleren. Door re-engineering, kan een legacy IS worden "verjongd". Echter, naarmate de tijd verstrijkt, blijven er verouderingsfactoren bij komen, tot er op enig moment een drempel (threshold) wordt overschreden, dan wordt er besloten om afscheid te nemen van een legacy IS. Het voorspellen wanneer deze drempel bereikt wordt - mede door externe factoren buiten de invloed van een organisatie - blijkt zeer moeilijk. Hoewel het lastig is IS-en discreet in te delen in IS-en en legacy IS-en, wil dit niet zeggen dat het niet mogelijk is om deze legacy IS-en te managen. Een voorwaarde voor het managen van legacy IS-en is een gedetailleerde IS repository waarin alle IS-en zijn opgenomen.

Hoe besluiten organisaties om afscheid te nemen van legacy IS-en?
In het onderzoek zijn 38 triggers geïdentificeerd die organisaties doen besluiten om afscheid te nemen van legacy IS-en. Er is een overlap tussen 13 triggers om afscheid te nemen met legacy IS verouderingsfactoren, zowel in het exploratieve onderzoek als het validatie onderzoek. Bijvoorbeeld, hoge kosten werd als een legacy IS verouderingsfactor gezien, maar ook als een reden (een trigger) om afscheid te nemen van een legacy IS. Daarom is geconcludeerd dat de verouderingsfactoren accumuleren met een trigger om afscheid te nemen. Het is echter altijd één trigger die een organisatie doet besluiten om afscheid te nemen, dit is geïllustreerd in Figuur 25 (p. 91). De 38 triggers zijn verder gecategoriseerd in interne triggers die beïnvloedbaar zijn door een organisatie (27), en externe triggers die niet beïnvloedbaar zijn door een organisatie (zeven) en triggers die deels beïnvloedbaar zijn door een organisatie (vier).

Voor wat betreft de 38 geïdentificeerde triggers om afscheid te nemen, zijn sommige triggers overlappend. Geïnterviewden noemden bijvoorbeeld een kosten-baten afweging een trigger om afscheid te nemen, anderen noemden weer alleen hoge kosten een trigger om afscheid te nemen. Uiteraard ligt hier een relatie. Eén ambivalente trigger "complexiteit" werd genoemd door geïnterviewden als reden om juist wel of juist niet afscheid te nemen van een legacy IS.

In de besluitvorming om afscheid te nemen van legacy IS-en, zijn drie besluitvormingsperspectieven overheersend, een functioneel perspectief, een technisch perspectief en een economisch perspectief. Er is daarom geconcludeerd dat bij de besluitvorming om afscheid te nemen van legacy IS-en vertegenwoordigers vanuit deze perspectieven moeten participeren. Besluitvorming door een groep van vijf tot 15 stakeholders leek te volstaan in alle organisaties. In alle organisaties is de IS eigenaar (de bedrijfsdirecteur) eindverantwoordelijk voor de gemaakte beslissing om afscheid te nemen van een legacy IS. Verder is de waarneming dat bedrijfsbelangen belangrijker zijn dan IT-belangen bij het afscheid nemen van legacy IS-en.

Volgens geïnterviewden is de financiële impact van een beslissing om afscheid te nemen van een legacy IS minder duidelijk dan van de beslissing om te investeren in de ontwikkeling van een nieuw systeem. Deze observatie is tegenstrijdig met de aanname vooraf, dat er veel financieel historische informatie bekend is van bestaande legacy IS-en. In tegenstelling tot nieuw te ontwikkelen IS-en, waar nog geen enkele historische financiële informatie beschikbaar is. Een verklaring hiervoor kon niet worden gegeven, verder onderzoek wordt hier voorgesteld.

Het besluit om afscheid te nemen van een commercial off-the-shelf systems (COTS) is makkelijker dan een zelfgebouwd legacy IS. Dit heeft te maken met het feit dat er minder binding is met een COTS in vergelijking met een zelf gebouwd legacy IS. Bovendien hadden vaak andere organisaties al ervaringen met het afscheid nemen van een vergelijkbaar COTS en was er kennis en ervaring in de markt aanwezig.

Het besluit om afscheid te nemen van een legacy IS dat een secundair proces ondersteunt (bijvoorbeeld een intern HRM-systeem), is makkelijker dan een legacy IS dat een primair proces ondersteunt. Secundaire legacy IS-en hebben doorgaans minder impact op de primaire processen. Er zijn vaak meer mogelijkheden voor vervanging, omdat deze pakketten vaak ook in andere organisaties gebruikt worden en er daarom sprake is van schaalvoordelen en concurrerende aanbieders. Tenslotte hadden secundaire legacy IS-en vaak minder stakeholders.

Het besluit om afscheid te nemen van een legacy IS dat is geoutsourcet, blijkt makkelijker dan een "in-house" legacy IS. Dit heeft te maken met het feit dat er minder persoonlijke binding was met geoutsourcete legacy IS-en en dat er geen medewerkers boventallig raken.

Belangen van stakeholders kunnen groot zijn bij het afscheid nemen van legacy IS-en (bijvoorbeeld managementfuncties, span of control, omzet voor leveranciers). Afscheid nemen van dergelijke grote belangen is risicovol. Eén overheidsorganisatie informeerde vooraf de politiek over de beperkingen van de

bestaande legacy IS-en, om te borgen dat er geen wetgeving werd gemaakt die legacy IS-en niet konden uitvoeren.

Swanson en Dans (2000), vonden dat grotere legacy IS-en een hogere overlevingskans hebben ten opzichte van kleinere legacy IS-en. In dit onderzoek worden hiervoor verschillende redenen gevonden:

1. Grote legacy IS-en ondersteunen vaak meerdere producten of diensten (soms wel 25). Indien er één product wordt afgestoten, blijven er in dergelijk grote legacy IS-en nog 24 over.
2. Grote legacy IS-en hebben doorgaans meer gebruikers en vaak ook belangrijkere stakeholders in vergelijking tot kleine legacy IS-en.
3. In het geval van een fusie, waarbij twee gelijksoortige legacy IS-en zijn (één groot IS met veel gebruikers en één klein IS met enkele gebruikers), is het goedkoper om enkele gebruikers van een klein legacy IS te trainen voor het gebruik van een groot legacy IS.
4. Grote legacy IS-en vormen vaak het hart van de organisatie en hebben een groter risico om afscheid van te nemen dan kleine legacy IS-en.
5. Specifieke functionaliteit van een klein legacy IS was makkelijker in een groot legacy IS in te brengen dan andersom.
6. Grote Legacy IS-en konden relatief gemakkelijk extra productievolumes van kleine legacy IS-en aan, terwijl kleine legacy IS-en problemen kunnen hebben om grote extra hoeveelheden van grote legacy IS-en aan te kunnen (in het geval van een fusie waren grotere legacy IS-en beter schaalbaar dan kleine legacy IS-en).
7. Grote legacy IS-en zijn moeilijk te herbouwen en te implementeren, het kost veel resources en risico's en kosten zijn hoog.
8. Het is soms onmogelijk om een groot legacy IS opnieuw te bouwen, door het feit dat ontwikkelen misschien wel 5 jaar heeft gekost en er 10 jaar aanvullend onderhoud is gedaan, daarnaast hebben organisaties vaak de benodigde kennis niet meer in huis.
9. De desinvestering bij het afscheid nemen van grote legacy IS-en is vaak erg groot.

Tenslotte zijn er 37 "good practices" ten aanzien van besluitvorming omtrent het afscheid nemen van legacy IS-en gevonden.

Hoe realiseren organisaties in de praktijk het afscheid van legacy IS-en?
De laatste fase in de levenscyclus van een legacy IS, is het in de praktijk realiseren van het afscheid van een legacy IS. In deze fase is een goede repository met informatie over alle IS-en essentieel en is communicatie met de stakeholders belangrijk. Verder moet onderhoud in de periode voordat het afscheid wordt gerealiseerd, tot een minimum worden beperkt. Een fallback scenario is noodzakelijk. Het in de praktijk realiseren van het afscheid van legacy IS-en kan veel stress opleveren bij stakeholders, het management moet hier aandacht aan besteden. In het geval dat een legacy IS wordt opgevolgd door een nieuw IS, is een succesfactor om stakeholders van het oude legacy IS te betrekken bij het nieuwe IS. Indien er stakeholders worden ontslagen als gevolg van het realiseren van het afscheid van legacy IS-en, is een goede afvloeiingsregeling noodzakelijk.

Verder is de IS eigenaar (bedrijfsdirecteur) altijd eindverantwoordelijk voor het in de praktijk realiseren van het afscheid van legacy IS-en; verantwoordelijkheid voor realisatie kan worden gedelegeerd naar een project of een programma. Indien een organisatie een fusie of consolidatie achtergrond heeft, heeft het de voorkeur een specifiek programma/project te hebben dat het afscheid van legacy IS-en realiseert (redenen zijn efficiency en het opdoen en borgen van specifieke kennis en ervaring).

Vaak bleek extra werk nodig om het afscheid van een legacy IS in de praktijk te realiseren, bijvoorbeeld als er essentiële data moest worden opgeslagen. Ook bleek documentatie vaak niet adequaat of zelfs afwezig. De werkzaamheden om het afscheid van legacy IS-en te realiseren vereisen analytisch vaardige

medewerkers met uithoudingsvermogen, die willen graven in “oude” informatie en het leuk vinden om met “oude” technologie te werken.

Tenslotte zijn er 60 “good practices” geïdentificeerd ten aanzien van het in de praktijk realiseren van het afscheid van legacy IS-en.

Een methode om afscheid te nemen van legacy IS-en
Een methode om afscheid te nemen van legacy IS-en is gebaseerd op de ervaringen van de vijf case-organisaties. Allereerst is geconcludeerd dat er een nieuwe methode nodig is om afscheid te nemen van legacy IS-en. Het hiërarchische niveau van deze methode is het IS, de methode moet zowel financiële als niet financiële evaluatie elementen bevatten, er moet sprake zijn van een pragmatisch aantal besluitvorming attributen, het moet een portfolio aanpak bevatten en verschillende stakeholders die verschillende disciplines vertegenwoordigen (functioneel, technisch en economisch) moeten worden betrokken, de IS eigenaar (bedrijfsdirecteur) is altijd eindverantwoordelijk.

Een methode bestaande uit tien stappen wordt voorgesteld (zie paragraaf 8.3, p. 144). Vooralsnog is het bestaan van een dergelijke methode ons niet bekend. De methode is gevalideerd door middel van argumenten, echter in de praktijk is de methode nog niet gevalideerd. In vervolgonderzoek zou de methode in de praktijk moeten worden getoetst.

De primaire onderzoeksvraag
Dit proefschrift is er op gericht te komen tot een verbeterd begrip over legacy IS-en en inzicht te krijgen in het proces om afscheid te nemen van legacy IS-en en om een methode voor te stellen om afscheid te nemen van legacy IS-en. De primaire onderzoeksvraag was:

Hoe identificeren organisaties legacy IS-en en hoe managen ze het afscheid nemen van legacy IS-en?

Op basis van de laatste opvolgende fasen in de levenscyclus van een legacy IS, beschrijft dit proefschrift uitgebreid de laatste fase, de “end of life” van een legacy IS. De literatuur was beperkt en de sociale relevantie was hoog. Achtereenvolgens werden het verouderingsproces van legacy IS-en, het besluitvormingsproces om afscheid te nemen van legacy IS-en en vervolgens het proces om in de praktijk het afscheid van legacy IS-en te realiseren onderzocht. Dit leverde richtlijnen op voor een methode om afscheid te nemen van legacy IS-en.

Er zijn vijf organisaties onderzocht, die afscheid hebben genomen van 2.020 legacy IS-en. Er zijn 51 interviews afgenomen, de geïnterviewden werkten tussen de vijf en 30 jaar in de ICT-industrie en hebben vaak meerdere werkgevers gehad. Deze studie geeft daardoor inzicht in de veroudering van legacy IS-en, de beslissing om afscheid te nemen van legacy IS-en en de realisatie van het afscheid van legacy IS-en in de praktijk.

Externe validiteit
Dit onderzoek is uitgevoerd bij vier grote publieke organisaties en één grote private organisatie (tussen de 1.200 en 18.000 medewerkers), in totaal zijn 51 interviews uitgevoerd, waarbij de geïnterviewden tussen de vijf en 30 jaar ICT werk ervaring en vaak meerdere werkgevers hebben gehad. Alle geïnterviewden zijn gevraagd naar verouderingsfactoren van legacy IS-en, naar triggers om afscheid te nemen van legacy IS-en, naar het besluitvormingsproces om afscheid te nemen van legacy IS-en en naar het proces om in de praktijk het afscheid van legacy IS-en te realiseren. Indien de laatste Case E niet was toegevoegd, was er geen verouderingsfactor gemist en slechts één trigger om afscheid te nemen van legacy IS-en. Daarom wordt verondersteld dat het toevoegen van andere cases geen significante veranderingen zal opleveren. Het was niet mogelijk om de resultaten te generaliseren naar middelgrote (tussen 1.000 en 100 werknemers) en kleine organisaties (<100 werknemers). Vervolgonderzoek wordt hier geadviseerd.

Tenslotte zal een gecentraliseerde IS repository ook waardevolle informatie leveren voor de investeringsbeslissing van nieuw te ontwikkelen IS-en en het onderhoud op bestaande IS-en. De 149 geïdentificeerde verouderingsfactoren kunnen helpen bij beslissingen ten aanzien van onderhoud en re-engineering. Verder zullen een groot aantal "good practices" ten aanzien van de beslissing om afscheid te nemen van legacy IS-en ook waardevol zijn bij besluitvorming ten aanzien van investeren in te ontwikkelen IS-en en besluitvorming rondom het onderhouden van IS-en. De dominant aanwezige perspectieven (functioneel, technisch en economisch) zullen ook waardevol zijn voor besluitvorming rondom investeringen in nieuwe IS-en en besluitvorming rondom het onderhouden van bestaande IS-en.

Suggesties voor vervolgonderzoek
Deze studie biedt ruimte voor vervolgonderzoek, de volgende 12 punten zijn geïdentificeerd:

1. Door gebrek aan financiële data van legacy IS-en was het niet mogelijk financiële schattingen te maken ten aanzien van legacy IS veroudering. Een dergelijke berekening lijkt erg belangrijk bij het verder exploreren van het belang en de relevantie van individuele legacy IS verouderingsfactoren en legacy IS triggers om afscheid te nemen van legacy IS-en, daarom wordt vervolgonderzoek voorgesteld.
2. "Gezond ouder worden" is misschien ook wel van toepassing op IS-en, daarom wordt geadviseerd vervolgonderzoek te doen naar de legacy IS verouderingsfactoren en het praktische gebruik hiervan in het onderhoudsproces.
3. Geconcludeerd is dat de meeste geïdentificeerde legacy IS triggers om afscheid te nemen interne triggers waren, hetgeen suggereert dat organisaties ze kunnen beïnvloeden, verder onderzoek naar de frequentie verdeling van ieder individuele legacy IS trigger wordt aanbevolen.
4. De voorgestelde methode om afscheid te nemen van legacy IS-en is gevalideerd door middel van argumentatie, maar nog niet getest in de praktijk. Aanbevolen wordt de methode in de praktijk te testen.
5. De verouderingsfactoren en de triggers om afscheid te nemen van legacy IS-en worden op het meest gedetailleerde niveau weergegeven, vervolgonderzoek kan ze aggregeren naar hogere niveaus.
6. "Good practices" betreffende besluitvorming om afscheid te nemen van legacy IS-en en het in de praktijk realiseren van het afscheid van legacy IS-en zijn geïdentificeerd. Vervolgonderzoek, kan onderzoeken of deze "good practices" ook "best practices" zijn.
7. Dit onderzoek was gericht op triggers om afscheid te nemen van legacy IS-en; één ambivalente trigger "complexiteit" werd genoemd als reden om juist wel of juist niet afscheid te nemen van een legacy IS. Een interessant vervolgonderzoek zou kunnen zijn om vanuit een continueringsperspectief te kijken naar triggers om geen afscheid te nemen van legacy IS-en.
8. Om meer inzicht te krijgen in de besluitvorming om afscheid te nemen van legacy IS-en lijkt het concept van "stakeholder salience" erg interessant, het kan gedrag van stakeholders verklaren. Verder onderzoek wordt geadviseerd.
9. Uit het longitudinale onderzoek is gebleken dat het afscheid nemen van legacy IS-en leidt tot meer afscheid, dit heeft waarschijnlijk te maken met voortschrijdend inzicht, kennis en ervaring, alsook afnemende weerstand. Het wordt geadviseerd dit fenomeen verder te onderzoeken.
10. Onverwacht stelden geïnterviewden dat van nog te ontwikkelen nieuwe IS-en meer bekend was v.w.b. de financiële impact dan bestaande legacy IS-en die al vele jaren in een organisatie aanwezig zijn, een uitleg kan zijn; dat het investeringsproces beter is. Aanvullend onderzoek wordt voorgesteld.
11. De voorgestelde methode is vooral gebaseerd op de ervaringen die opgedaan zijn door organisaties Case A, Case B, Case C, ze hadden specifieke programma's met veel ervaring. De organisaties in het onderzoek hadden veel werknemers (1.200-18.000). Vervolgonderzoek moet uitwijzen of de methode ook bruikbaar is bij kleine organisaties met weinig legacy IS-en.
12. Er zijn nu een aantal methodes ontwikkeld voor verschillende fasen in de life cycle, de ontwikkeling van IS-en, het onderhoud van bestaande IS-en en aanvullend het afscheid nemen van legacy IS-en. Geadviseerd wordt om in vervolgonderzoek te onderzoeken of er integratie mogelijk is met theorieën over de andere fasen uit de life cycle.

13 ABOUT THE AUTHOR

Arnold Commandeur was born on May 14th 1965 in Wormerveer, the Netherlands. He holds a Bachelor of Science degree in Electronics from the HTS Alkmaar (1989) and Master degree in Economics from the University of Amsterdam (2001).

In 1989 Arnold started working at IBM, where he first had a career in PC services and procurement and subsequently worked for large account sales and business consultancy.

In 2003, Arnold went to work to work in the public sector. There he was employed in various departments: architecture, CIO office, ICT business control and accountancy. In 2006-2010 Arnold held a part time research position at the University of Groningen. Today Arnold is full time employed as a business architect.