Information system development in a process management environment: the dynamics of improvisation and bricolage during embedded software design

by

Wolfgang Alfred Molnar

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

University of Warwick, Warwick Business School

November 2009

"All men by nature desire to know."

Aristotle (384-322BC) Greek philosopher

Word cloud of this thesis[1]

---

1 The cloud give greater prominence to words that appear more frequently in the thesis text and was created by a Java application, derived from http://www.alphaworks.ibm.com/tech/wordcloud at 1st August 2009.

Table of Contents

Figure Listing

Table Listing

Acknowledgements

"It seems to me shallow and arrogant for any man in these times to claim he is completely self-made, that he owes all his success to his own unaided efforts. Many hands and hearts and minds generally contribute to anyone's notable achievements."

Walt Disney (1901-1966) Film producer

This dissertation is the result of the support and dedication of many people. Sincerest thanks are due the following:

To Professor Nandhakumar, my dissertation advisor and mentor of four years, much appreciation for the many hours you freely have given and for your example and commitment as a scholar. Your devotion and enthusiasm for this project have strengthened it greatly, as have your insights on the field of information system development in practice.

To my mother, the project would not have been completed without the love and compassion you have shared.

As with any scientific research, the role of empirical data is priceless. I am therefore grateful to the company I received the data from and the personnel whom I had contact with. Since the anonymity is to be preserved, no names are cited here. Without these people, however, the claims made in thesis could not have been empirically explored.

The Warwick Business School, University of Warwick, University of Bath, and American Study and Student Exchange Committee have provided financial support for this thesis. Their support and trust has made this research possible.

Declaration

I declare that, except where acknowledged, the material contained in this thesis is my own work and has not been submitted elsewhere for the purpose of obtaining an academic degree.

Wolfgang Alfred Molnar
November 2009

List of publications

1. Molnar, W., & Nandhakumar, J. (2007). *Managing a New Computer Device Development in a Creative ISO 9001 Certified Company: A Case Study*. Paper presented at the Hawaii International Conference on System Sciences, Waikoloa, HI, USA.

2. Molnar, W., & Nandhakumar, J. (2008). *Software Development Standards and Improvisation: A Case Study.* Paper presented at the SPICE 2008, Nuremberg, Germany.

3. Molnar, W., & Nandhakumar, J. (2008). *Embedded System Development Process: A Quest from Innovation to Adaption.* Paper presented at the 16th European Conference on Information Systems (ECIS), Galway, Ireland.

4. Molnar, W., & Nandhakumar, J. (2008). *Dealing with Ambiguous and Fluctuating Requirements of Embedded System Development: A Case-Study.* Paper presented at the International Conference on Information Systems (ICIS), Paris, France.

5. Molnar, W., & Nandhakumar, J. (2009). *Managing Projects in an Embedded System Development Context: An in-depth Case Study from an Improvisational Perspective.* Paper presented at the Hawaii International Conference on Systems Science, Waikoloa, HI, USA.

Abstract

The main objective of this thesis is to make a contribution to knowledge regarding the nature of improvisation and bricolage activities in the practice of embedded software design and how the tensional relationship between process management and improvisation and bricolage can be balanced. There is a lack of understanding embedded systems development in practice, and how the difficulties correspond to prescribed and emergent processes in this context. In order to address this knowledge gap I conducted an in-depth case study of an embedded system development project in the German automobile context between December 2004 and November 2008. The research adopted an interpretive approach, which involved the collection and analysis of qualitative data. Empirical data that was derived through interviews and observation revealed new insights as to how embedded systems are developed in practice. I adopt the position that emergent processes occur not randomly, but as purposeful agents that navigate through a turbulent environment of ongoing need to improvise with the items at hand. The finding indicates that the success to achieve the aims is bound to the capabilities to be continuously reflexive and induce corrective actions as appropriate. A theoretical conceptualisation disclosed measures that may enhance the capacity to be reflexive. The findings implied that process management frameworks help as scaffolding in order to practice improvisation and bricolage as a coping strategy. Moreover, improving the capabilities to cope with challenges means enhancing reflexive capabilities. The original contribution of this research is founded on rich descriptions and interpretations as to how embedded systems are developed in practice, and the theoretical conceptualisation that can aid to balance the tension between process management and improvisation and bricolage.

Keywords: Embedded system development, improvisation, bricolage, process management.

1 Introduction

1.1 Embedded systems in automobiles

"Chrysler is recalling 37˙000 Dodge Ram pickup trucks due to faulty software"[1]
"Mazda3 recalled over stability control software concern"[2]
"BMW: Software faults and incorrect Piezo injectors cause engine fail-safe program with power loss."[3]

Those and similar news stories concerning automobile recalls occur frequently and indicate that despite the efforts of many engineers, designing automobiles remains a complex task and involves the assembly of many subsystems[4]. Those subsystems are embedded in automobiles, meaning they are inserted within the larger network of multiple subsystems. Today, luxury cars have anywhere

---

1 Ramsey, J. (2009). 37,000+ Dodge Ram pickups recalled over... faulty HVAC software?!. *Autoblog* Retrieved May 12th, 2009, from http://www.autoblog.com/2009/05/12/37-000-dodge-ram-pickups-recalled-over-faulty-hvac-software/

2 Shunk, C. (2009). 2009 Mazda3 recalled over stability control software concern. *Autoblog* Retrieved June 5th, 2009, from http://www.autoblog.com/2009/06/02/2009-mazda3-recalled-over-stability-control-software-concern/

3 ADAC. (2009). BMW: Softwarefehler und fehlerhafte Piezo-Injektoren verursachen Motornotlaufprogramm mit Leistungsverlust. *Rückruf-Datenbank* Retrieved April 30th, 2009, from http://www1.adac.de/Auto_Motorrad/pannenstatistik_maengelforum/Rueckrufe/rueckruf_popup.asp?kat=PKW&hst=BMW&mod=-&page=3&katKlicked=Nein&hstKlicked=Ja&modKlicked=Nein

4 "A subsystem is a set of elements, which is a system itself, and a part of a larger system." - System. (n.d.). *Wikipedia* Retrieved September 19th, 2009, from http://en.wikipedia.org/wiki/System

between 40 to 70 different embedded systems along with their sensors and actuators and all these control modules need to exchange information with each other[1]. And this number will continue to rise, because automobiles in the future will make significant use of embedded systems and software will increasingly define what your automobile is and what it does (Broy, 2006). And that is advantageous, inasmuch as embedded systems help to support the drivers' safety and comfort. More and more embedded systems are involved in automobiles such as the anti-lock brake system, electronic stability program, and airbags.

However, as the recalls indicate, it is obviously difficult to maintain high quality, when many subsystems are embedded in automobiles. Usually, those recalls involve multi-million dollar costs that could have been avoided if the automobiles were designed properly. Therefore, a study[2] of the automobile industry states that product quality remains the most cited issue. Consequently, this includes the design of subsystems, which are embedded in automobiles. The approach to improve the fabrication of automobiles that had a profound influence within that industry was the application of process management (PM). Indeed, process management can be traced back to the early days of automobile production when Henry Ford[3] introduced the mass production. Since then, process management remained important for that industry and it spread worldwide. For example, International Organisation for Standardisation (ISO) 9001 is a process management framework that a gained great reputation

---

1 Jena, C. (2007). The IT-enabled automobile. *Express Computer* Retrieved March 17th, 2007, from http://www.expresscomputeronline.com/20070219/management01.shtml

2 KPMG International. (2008). Momentum: KPMG's Global Auto Executive Survey 2009. *RDL 1848* Retrieved February 2nd, 2009, from http://www.kpmg.com/SiteCollectionDocuments/Momentum-KPMG-Auto-Executive-Survey-2009.pdf

3 Henry Ford. (n.d.). *Wikipedia* Retrieved August 15th, 2009, from http://en.wikipedia.org/wiki/Henry_ford

and became important for many industries (Benner & Tushman, 2002). In addition, various domains established specialised process management frameworks, such as Software Process Improvement and Capability Determination (SPICE) and Capability Maturity Model Integration (CMMI) that became important for Information System Development (ISD). Those frameworks share the idea to prescribe processes and methods, in order to improve the quality and effectiveness.

Organisations following PM frameworks prescribe processes to make it easier for employees to cope with challenges, such as vagueness in daily work, high costs, and quality issues (e.g.: Dick, 2000; Rada, 1996). Therefore, those organisations work in a more standardised manner, have lower costs, and imply better product and service quality (e.g.: Dick, 2000; Rada, 1996); hence, the products and services seemingly get done more properly. Particularly the development of information systems should have benefited from that trend to apply process management, because various ISD methodologies were proposed that aim to help the developers and managers. However, the fact is that cost intense errors within embedded systems still happen as the recalls for automobiles reveal. It appears that prescribed processes are not fully accepted among practitioners (Fitzgerald, 1998), because they experience the application of prescribed processes as potentially difficult owing to unpredictable events and other conditional factors (du Plooy, 2002). In addition, inherent creative activities such as ISD can be jeopardised or hampered, because developers and managers feel constrained when they need to follow prescribed guidelines (Sawyer & Guinan, 1998).

1.2 Practice-based motivations for the study

In practice ISD is continually emergent, and it is difficult to describe ISD with predefined patterns (Truex, Baskerville, & Klein, 1999). Therefore, organisations perceive prescribed processes as challenging to follow and constraining for development activities. However, organisations want to develop novel solutions and maintain the ability to manage effectively. And the application of PM frameworks provides a sense of control, although this may not be the case in practice (cf. Nandhakumar & Avison, 1999). Because in practice, PM frameworks may operate as an illogical procedure, the enactment of which provides developers and managers with a feeling of security and effectiveness at the expense of real engagement with the task at hand (Wastell, 1996). Therefore, there is a tension between the prescriptive procedures that people should use, while practitioners actually need to cope with continually emergent situations that are difficult to determine in advance. Those emergent situations may involve improvisation and bricolage (defined as making do with the items at hand (Ciborra, 2002; Lévi-Strauss, 1966)) that are less easy to describe. Consequently, practitioners can benefit from rich descriptions of improvisational and bricolage activities that occur in a process management setting, because a better understanding of emergent processes implies that they can resolve in similar situations better.

Although the tension between process management and improvisation and bricolage prevailed over a long time, there is still little industry experience as to how to balance this tension better. Particular embedded system developments in practice suffered from a lack of industrial know how, because solutions are differently situated than non-embedded systems. Embedded systems band together the realms of software and hardware and, therefore, the development is fundamentally different to non-embedded systems (e.g. Graaf, Lormans, & Toetenel, 2003). For example, the development addresses specific constraints such as hard timing constraints, limited memory, and power usage. Consequently, proposed solutions such as various ISD methodologies are not intended for embedded system development, hence they are inappropriate to apply (Kettunen & Laanti, 2005). In addition, embedded systems that are implanted in automobiles need to fulfil high-quality standards, because insufficient quality may risk human lives in case of an accident (H. Hoffmann, Leimeister, & Krcmar, 2007). Therefore, the development of those embedded systems needs to be controlled and managed appropriately to meet the high-quality standards. However, as stated above, developers and managers find it difficult to follow precisely the prescribed processes (Fitzgerald, 1998; Nandhakumar & Avison, 1999). As a result, it is challenging to develop embedded systems adequately, without creating difficulties with the other subsystems of the automobile. If developments are performed poorly, cost-intensive recalls may be necessary. For example, to recall 500 new Mini's was estimated to cost the carmaker about £320˙000[1]. Therefore, that industry has a natural self-interest to cope better with the difficulties between process management and improvisation and bricolage.

Section 7.3 presents the implications of the research findings for practice, and so attempts to reduce some of the current difficulties.

1.3 Knowledge-based motivations

As can be seen from the literature review (section 2.2), process management frameworks such as ISO 9001 intend to increase manageability of activities within organisations. Those activities involve the development of information systems and a large part of ISD literature (section 2.3.1) concerns about prescribed methods that guideline to develop information systems systematically. However, ISD is also influenced by non-technical factors (e.g. social relations, politics and cultures)(du Plooy, 2002). Therefore, prescribed methods may be of little help, when non-technical factors interrupt the development activities (Truex, et al., 1999). Indeed, Fitzgerald (1998) describes

---

1 BBC News. (2001). BMW recalls new Mini. *BBC News Business* Retrieved July 3rd, 2009, from http://news.bbc.co.uk/2/hi/business/1523086.stm

process management frameworks as too general without contextual considerations and an overemphasis on technical aspects at the expense of social aspects, individual creativity, or learning while doing. The literature review (section 2.3.2.2) indicated that only few researchers highlight those limitations of prescribed processes in the context of ISD in practice. The practice of ISD involves improvisation and bricolage activities (e.g. Ciborra, 2002), which are contrary to prescribed processes. Although there are studies available concerning improvisation and bricolage in the context of ISD (e.g. Ciborra, 2002; Nandhakumar & Avison, 1999), scholarly understanding of improvisation and bricolage in the complex setting of embedded system development is relatively thin. Therefore, it is of interest how developers and managers practice improvisation and bricolage when they develop an embedded system. In addition, the research setting involved the collection of empirical data throughout one development project that involved the opportunity to analyse empirical data as to how improvisational and bricolage activities may change over time. Consequently, more knowledge about improvisation and bricolage activities open avenues to discuss those activities from the basis of new insights. In addition, the challenge of embedded system development is that it combines the realms of software and hardware (e.g. Graaf, et al., 2003). Embedded system development is physically engaged and it is therefore different to the abstractions of non-embedded ISD knowledge (Kettunen & Laanti, 2005). Consequently, knowledge of non-embedded ISD activities is of little help for the development of embedded systems. This thin spread of knowledge concerning embedded systems development in practice exposes a knowledge gap and my doctoral study addresses this deficiency.

From the literature review (sections 2.3.1 and 2.4), it can further be seen that the thin spread of knowledge concerning embedded systems development involves less aids for developers and managers, which helps them to understand the process, master challenges, and make use of opportunities to design innovative embedded systems. This is a problem, because organisations want to understand the process, cope with challenges, and make use of opportunities to develop novel embedded systems (sections 2.2.3.3 and 2.4.1). Inappropriate process management frameworks provide less procedural help to support or manage the dynamics of improvisation and bricolage during embedded system development. For example, Turk et al. (2002) claim that a 'silver-bullet' does not exist for the complexity of ISD. Hansen et al. (2003) argue that process management frameworks are adjusted in action and no universally applicable framework exists. However, organisations apply process management frameworks, because they want at least a symbol that provides the fiction of being capable to manage system development activities (Nandhakumar & Avison, 1999). Nandhakumar and Avison (1999) continue that in practice the applied framework was too formal and structured to be of any help in the development process. Therefore, the knowledge deficit about measures that help the development activities in practice needs to be addressed. Particularly interesting are measures that balance the tension between a process management environment and development activities that involve improvisation and bricolage. That is, because improvisation and bricolage activities are involved in the practice of ISD (e.g. Ciborra, 2002) and organisations want to share the benefits of the process management framework (sections 2.2.3.3 and 2.3.2.1). This thesis provides a theoretical conceptualisation based on the empirical findings of this research, which answers what measures help to balance the tensional relationship in the context of embedded system development.

I describe the identified knowledge gaps in the literature and research question in section 2.5. In addition, section 7.2 presents the implications for theories and models, and so addresses the knowledge gaps identified in the literature review.

1.4 Outline of thesis

The thesis is structured as follows: chapter two presents a review of the literature concerning process management, ISD methodologies, ISD in practice, and the development of embedded systems. The review indicates knowledge gaps in the literature, which determine the focus of this research with detailed research questions. Chapter three describes my research approach, data collection and analysis approaches. Furthermore, chapter three provides a summary of the fieldwork conducted and describes the applied theoretical lenses that helped to analyse the empirical data. Chapter four provides an in-depth case study about the development project of an embedded system in a process management environment. The case study analysis in chapter five applies various analytical topics to examine the empirical data from different perspectives. Chapter six discusses the research findings, answers the research questions, and relates findings to previous research findings found in the literature review. Chapter seven describes implications for theories and practice, opportunities for future research, and it states concluding remarks.

"It is impossible to design a system so perfect that no one needs to be good."

T. S. Eliot (1888-1965) American-born British Poet, Critic, Playwright

2 Literature Review

2.1 Introduction

The goal of this chapter is to review the literature that is concerned with the management and human / social aspects of the Information System Development (ISD) process, as well as the context within which they are situated. A solid basis for processes that imply organisation seems to be provided by process management practices and techniques of international standards, such as the International Organisation for Standardisation (ISO) 9001 (Benner & Tushman, 2002; Guler, Guillén, & MacPherson, 2002). That process management framework implies that improvement of processes may increase the quality of the process results, such as products and services (ISO, 2008b). In section 2.2, I assemble a short review of the current literature concerning process management by emphasising quality management and the ISO 9000 family of standards for quality management systems.

The management aspect of the ISD process refers to the body of knowledge of ISD methodologies collected since the early days of the waterfall process model (Royce, 1970). There are several approaches in the ISD literature available, which differentiate between various process models (Avison & Nandhakumar, 1995; Iivari, Hirschheim, & Klein, 2004). The goal of section 2.3 is to provide a brief review of the current ISD literature by focusing on the formal process models and methods of software development. Consequently, section 2.3 also yields profound understanding as to how ISD methodologies are used in practice. Section 2.4 follows a review of the research literature concerning a sub-theme of ISD, namely embedded system development. It focuses on the application and contextual challenges of embedded system development.

Finally, in section 2.5 research aims and questions are expressed. The motivation of this research is based on two identified knowledge gaps in the literature, which are stated in section 2.5.1. Those gaps confined the focus of this thesis: embedded software development in practice. Moreover, this thesis is interested in the dynamics of improvisation and bricolage during embedded software development. The research questions are stated in section 2.5.2 and try to address the identified knowledge gaps in the literature.

2.2 Process management

2.2.1 Introduction

Pentland and Feldman (2005) state that a deliberate attempt to capture or prescribe the actions and practices of human beings are formal rules or standard operating procedures. Process Management (PM) frameworks (e.g., Six Sigma, Total Quality Management, ISO 9001, and so on) may cover some formal rules or standard operating procedures (Benner & Tushman, 2002; van der Wiele & Williams, 2000; Wang, et al., 1997; Yusof & Aspinwall, 2000). In fact, early attempts to enhance efficiency and standardisation can be traced back to Taylor's scientific management from the late nineteenth and early twentieth centuries (Boje & Winsor, 1993). However, the goal of this section is not to narrate the history of PM frameworks, but instead to briefly review the literature about the scope of PM, with quality management and ISO 9001 in the context of development activities.

Benner and Tushman (2002) describe the concept of PM as focusing on variance reduction and increased process control that improves both speed and organisational efficiency. PM frameworks initially focused on the area of manufacturing goods, but it has spread into processes of other areas, such as administrative, procurement, development, and so on (Benner & Tushman, 2002). Especially, the potential of PM frameworks to affect development enterprises directly has also increased, because programs such as ISO 9001 and Six Sigma extended their scope into product design and development activities (Hoerl, Snee, Czarniak, & Parr, 2004). Therefore, it is of interest as to how PM frameworks might affect development enterprises (Sutcliffe, Sitkin, & Browning, 2000), and in section 2.2.2 I will elaborate this issue.

In section 2.2.3, I will examine more about the PM framework of ISO 9001 and emphasise basic issues in order to promote a better understanding of ISO 9001. From time to time, the ISO 9001 standard undergoes some updates, and the current version is from 2008. The organisation from which the data were retrieved is certified on the previous version of ISO 9001 from 2000. However, it managed to accomplish ISO 9001:2008 certification in 2009, when I followed up the developments of that organisation. The changes between 2000 and 2008 are only marginal (ISO, 2008a, 2008b), but I will base the literature review on the version from 2008. For a better understanding, I will enclose a summary of the changes between ISO 9001:2000 and ISO 9001:2008 in section 2.2.3.2.2.

2.2.2 Development activities and process management

2.2.2.1 Extent of process management

There is no doubt that creativity plays an essential role in development activities, but it is also necessary for managing any business for future challenges (Kondo, 1996). PM, on the other hand, draws attention to variance reduction and increased process control (Benner & Tushman, 2002). Although both elements of creativity and PM are considered important for organisational management, they are thought to be mutually exclusive, because the space left for inventive activities is decreased along with the progress of PM (Kondo, 1996). For example, Benner and Tushman (2002) claim the greater the extent of PM activities in an organisation, the smaller the number of inventive activities. As a more practical example, Nanda (2003) claims that key issues of PM frameworks, like documenting activities, are problematic in software development because code is a peerless, but unhandy, piece of documentation for many people. However, time spent on documentation for fulfilling PM regulations is time lost for other important developer tasks. Because of reasons such as these, implementation of prescribed processes has faced troubles in many development departments (Kautz, Hansen, & Jacobsen, 2004).

However, the success of PM frameworks is remarkable. For example, the quality management standard ISO 9001 had approached 1˙000˙000 registrations in some 130 countries by the end of 2007[1]. The massive effort of organisations to implement PM frameworks may educe that the advantages prevail over the disadvantages and concerns. However, it must be stated that one of the motivations for companies to be certified is because customers request it (R. Jones, Arndt, & Kustin, 1997). Consequently, organisations may get beneficial ratings, awards, and other forms of recognition, enhancing their prestige (Dick, 2000). With so many other organisations that strive for certification, there is certainly some pressure to behave similarly to competitors.

2.2.2.2 Tasks of process management

The literature review reveals three main activities of PM that reduce variances and increase process control. First, vigorous effort is required to map or document routines that serve the performance of an organisation's good or service (Benner & Tushman, 2002; Harrington & Mathers, 1997). Next, the documented routines are then liable for process improvement (Benner &

---

1 Eden, C. (2007). Why is ISO 9001 so successful? *Quality Matters* Retrieved March 31st, 2007, from http://www.quality-matters.com/blog/2007_03_01_archive.html

Tushman, 2002; Harrington & Mathers, 1997). Process improvement efforts are aimed at gradual change so that organisational processes become more efficient (e.g. ISO, 2008b). Finally, the process management efforts rely on the continual comparison of process effectiveness and efficiency (e.g. ISO, 2008b). For example, quantifiable measurements can be taken from a number of products, time to market for new developments, and so on. In conclusion, PM efforts strive to improve activities by favouring determined routines and eliminating vagueness. However, many organisations face a constantly changing economic and technological world (Orlikowski, 1995). Constant change is a guiding principle, and organisations need to cope with those circumstances to survive the competition. In particular, when organisations develop high-tech services and goods, the ability to remain flexible is important. However, organisations are required to enforce changes consistently and efficiently. The organisational leaders owe such consistent and efficient management not only to their employees, but also to other stakeholders of the organisation, such as clients and shareholders.

In this literature review, I focus on a commonly accepted instrument to satisfy PM standards. Many PM frameworks are based on the continuing performance and evaluation of assessments and reviews to examine processes for their effectiveness and efficiency. In addition, improvement efforts are guided through a constant analysis of changes. For example, ISO 9001 (ISO, 2008b) states,

> The organisation shall continually improve the effectiveness of the quality management system through the use of the quality policy, quality objectives, audit results, analysis of data, corrective and preventive actions and management review. (p. 14)

It is interesting to note that the realm of the agile ISD methodology emphasised this constant self-monitoring (Turk, France, & Rumpe, 2005):

The agility in agile processes is achieved through continuous self-examination of the processes used and corresponding adaption of the process. (p. 73)

2.2.2.3 Comparison of important process management frameworks and development activities

The relevance and importance of PM frameworks tends to increase, and this section provides a brief comparison of PM frameworks and their relatedness to development activities. The different demands for development activities reasoned a flourishing market for various process management frameworks, such as lawfully prescribed process management frameworks (e.g. German commercial code), ISO 9001, situated in-house solutions, SPICE, CMMI, and so on. However, because of manageability, I want to focus at this stage on two PM frameworks that have an influence on organisations in the German jurisdiction. Table 2-1 provides a comparison of the German commercial code concerning development activities and the quality management standard ISO 9001. In the domain of ISD, PM frameworks also had some influence, such as the Software Process Improvement Capability Determination (SPICE) and the Capability Maturity Model Integration (CMMI) (cf. Table 2-2).

Table 2-1 Comparison of two PM frameworks

	German commercial code	**Quality management – ISO 9001**
Refer-ences	("Handelsgesetzbuch," 2008)	(ISO, 2000a, 2000b, 2005, 2008b; ISO/IEC, 2001)
Content	The "Handelsgesetzbuch" is the German commercial code. It includes the regulations for corporations in performing reports and closing books. Organisations within German jurisdiction need to follow this law.	The ISO 9000 family of standards has been developed to assist organisations, of all types and sizes, to implement and operate effective quality management systems.
Influence on development activities	This law (§ 289 exp. 1) is concerned about the report on the situation of the business performance in such a way that an appropriate picture is arranged for the actual conditions of the organisation. It involves the prospective development with its substantial chances and risks that are to be judged and described in the report. The basis of the assumptions needs to be indicated in the report. Although this regulation has no direct influence on the development activities of an organisation, the indirect consequences are important. For example, managers need to perform an analysis of risk concerning developments as part of their development activities. It is interesting that the ISD methodology called "spiral model of the software process" (Boehm, 1986) explicitly recognises the importance of managing risks. In conclusion, the main consequence of this regulation is the necessity of risk analysis of development activities.	This standard is mandatory for many businesses, like the automobile industry. Consequently, ISO 9001 has a significant influence on development activities, because the developers need to accomplish a variety of tasks in order to be part of an ISO 9001-certified organisation. For example, the organisation needs to plan and control the design and development of products. Components of this are the determination of design and development stages, appropriate reviews, verification and validation tasks, and responsibilities for the design and development.

Table 2-2 Comparison of PM frameworks in the domain of software development

	Software Process Improvement Capability Determination (SPICE) – ISO 15504	**Capability Maturity Model Integration (CMMI)**
Refer-ences	(Automotive-SIG, 2005; Fabbrini, Fusani, Lami, Sivera, & Sivera, 2007; ISO/IEC, 2003, 2004a, 2004b, 2004c, 2006, 2008a, 2008b; Stienen, 1999)	("CMMI® for Development, Version 1.2," 2006; Paulk, Konrad, & Garcia, 1995; Rout, 1998)
Content	ISO 15504 is an international standard and serve as a reference model for maturity determination of organisational capabilities for delivering products (software, systems, IT services).	The basis of CMMI is to yield guidelines for improving organisational processes and the ability to manage the acquisition, development, and maintenance of products or services.
Influence on software development activities	This standard serves particularly the software engineering environment, because the IS0 9000 family was not originally designed for this domain. However, SPICE follows a holistic approach in order to touch upon every possible process, which might affect the development activities. For example, this involves processes in acquisition, supply, operation, management, and so on. The standard describes processes from all aspects during the development, such as requirements elicitation, requirement analysis, architectural design, software design, software construction, software integration, software testing, system integration, system testing, and so on. SPICE emphasises the need for continual improvement activities of the organisational processes to remain effectiveness and efficiency and align with the business needs. Accomplishment of SPICE processes leads to different capability levels that are determined by independent organisations. The performed level of an organisation may serve as an indicator of organisational maturity. For example, an organisation that does have the ISO 9001 certificate does usually fulfil SPICE level 2.	The idea of the CMMI model is relatively similar to SPICE, but some processes of SPICE are seen to be either fully or partially outside the scope of CMMI (Rout & Tuffley, 2007). In addition, some aspects of CMMI are not directed in ISO 12207 (Standard for Information Technology – Software life cycle processes) (Rout & Tuffley, 2007). Interesting to note that an ISO 9001 fulfilment would achieve most level 2 goals of CMM and many of level 3 (Paulk, 1995).

2.2.3 Quality management – ISO 9001

2.2.3.1 Scope of quality management – ISO 9001

The influence of the quality management standard ISO 9001 is enormous as earlier stated in section 2.2.2.1. It is interesting that the literature review reveals a whole variety of definitions for quality (Geiger & Kotte, 2005). For example, Crosby (1979) describes quality as "conformance to requirements", and according to Juran and Godfrey (1998), quality set out "fitness for use." The American Society for Quality[1] describes quality as a subjective term, and individuals or business fields may have different interpretations. Generally speaking, quality is understood as a characteristic of goods or services, regarding their relevant values (Geiger & Kotte, 2005). The ISO 9000 family (ISO, 2005) defines quality as the "degree to which a set of inherent characteristics fulfils requirements," whereas characteristics are described as distinguishing features, and requirements are needs or expectations that are expressed, generally implied or compulsory (ISO, 2005).

The structure and scope of the ISO 9000 family documents are composed of four documents (ISO, 2005):

- ISO 9000 describes fundamentals and vocabulary;
- ISO 9001 specifies requirements for a quality management system;
- ISO 9004 provides guidelines for the continuous improvement of a quality management system; and
- ISO 19011 defines guidelines for quality and/or environmental management systems auditing

The conceptual idea of these standards is to assist organisations, of all types and sizes, to implement and operate effective quality management systems

---

1 American-Society-for-Quality. (2009). Quality. *Glossary* Retrieved February 25th, 2009, from http://www.asq.org/glossary/q.html

(ISO, 2005). This assistance is considered necessary in helping to manage and control organisations through structured and clear processes. A process in ISO 9001 is a managed series of activities using resources, thus enabling a transformation of inputs into outputs. On many occasions, the result of one process is the requirement of another process, and therefore a system of processes forms within an organisation. The application of a system of processes, together with the determination and connections of these processes and their management to get the desired results, can be referred to as the process approach, which is an important aspect of ISO 9001 (ISO, 2008b). Beyond this aspect, to manage and control prosperous organisations not only involves implementing and maintaining PM, but continually improving it as well. For this reason, ISO 9001 describes four different steps within every process (ISO, 2008b):

a) comprehension and meeting of requirements;
b) reflecting on the process concerning the added value;
c) acquiring measurements of process effectiveness and efficiency; and
d) continuous improvement of the processes based on the acquired measurements.

Figure 2-1 depicts the conceptual idea of ISO 9001, but does not show processes at a detailed level.

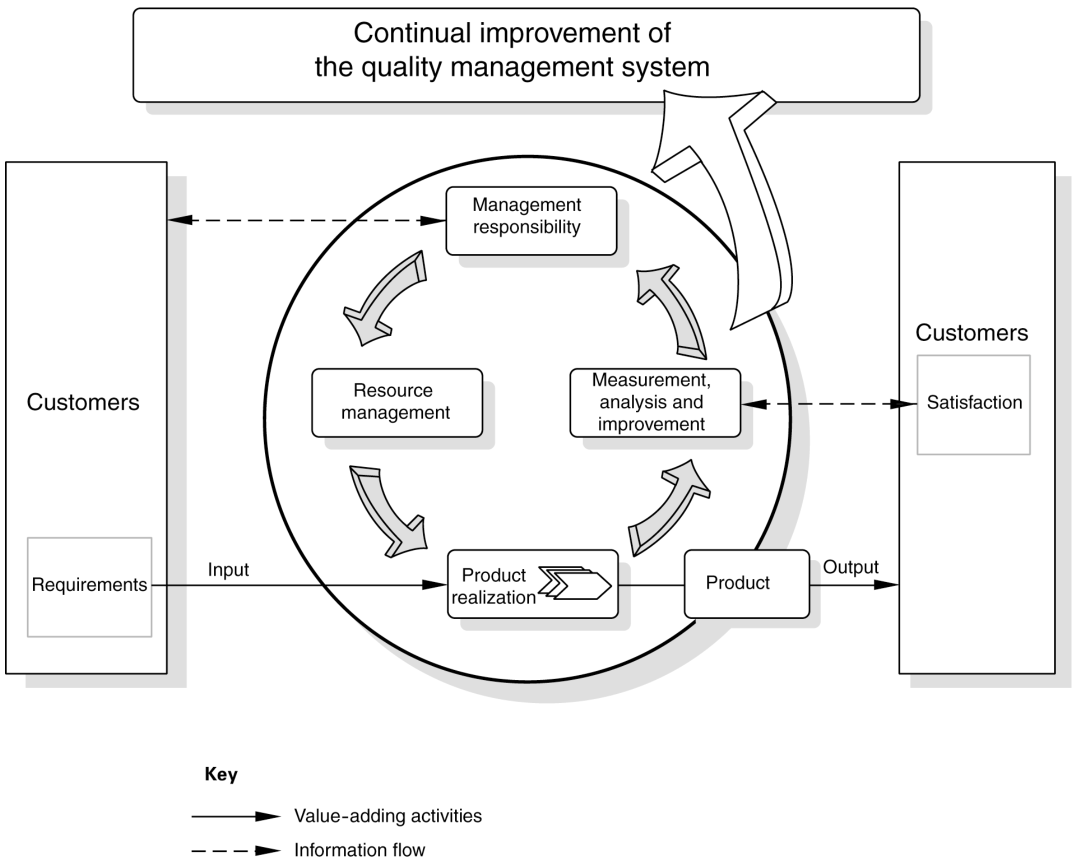

Figure 2-1 Model of a process-based quality management system (ISO, 2008b)

However, the figure above also shows that customers have an important role in the concept of ISO 9001. Customers determine requirements for the product or service they desire; the output product or service affects the customers' satisfaction; information flows between the customers and the organisation clarify the responsibilities of the management and deliver data for measurement, analysis and improvement of organisational processes. Therefore, the feedback of customers is a vital part of the continuous improvement of the quality management system. With this important role of the customer, it is valuable to explain who the customer is. ISO 9001 (ISO, 2005) defines the customer as an "organisation or person that receives a product" (p. 10). This might be consumers, clients, end-users, retailers, beneficiaries, and purchasers (ISO, 2005). Accordingly, customers can be internal or external to the organisation (ISO, 2005), and this understanding stays abreast of changes in marketing.

2.2.3.2 Content of ISO 9001

A number of other topics are important in ISO 9001 (ISO, 2005) in addition to the *process approach*, *continual improvement* and maintaining *customer focus*. *Leadership* in ISO 9001-certified organisations is also important, because leaders are supposed to establish homogeneity of purpose and provide the direction of the organisation. Leaders should initiate and support the internal setting in which individuals can participate in achieving the organisation's goals.

To *involve people at all levels* is another principle of ISO 9001. An organisation will benefit if all members of the organisation are allowed to fully participate and use their abilities to help achieve the organisation's goals. Another integral part of ISO 9001 is the *system approach to management*, because determining, comprehending and managing interrelated processes as a system induces the organisation's effectiveness and efficiency in realising its goals. Furthermore, using the *fact-based approach to decision-making* is the core of ISO 9001, since comprehension and analysis of data enables effective conclusions. Also in ISO 9001, *supplier relationships* should be mutually beneficial, for the reason that organisations and their suppliers are interdependent, and a beneficial relationship advocates the full potential of both partners to create value. The ISO 9000 family identified these eight quality management principles so that managers can apply them in order to improve the performance within their organisations. Moreover, the various rules and guidelines of the ISO 9000 family may be derived from these principles.

Structural element as referred to in ISO 9001 is recorded information in the form of documents, such as the quality manual, organisational procedures, specifications, records and so forth. A valuable item of the documents is the quality manual, which involves the scope of the quality management system, reference to the established and documented procedures, and an explanation of the interaction between the processes of the quality management system. Organisational procedures are documented with the help of descriptions and flowcharts in order to familiarise new employees with the procedures. Record documents are used to keep track of the process progress and allow the organisation to review the original data. To control documents and records is another important issue, because they serve as the basis for improvement efforts of the organisational PM. Finally, the scope of the quality management system documentation varies in form depending on the organisational business and size, the complexity of processes, and the expertise of the employees (ISO, 2008b).

2.2.3.2.1 Monitoring of ISO 9001

Organisations need to undergo regular audits, which are structured and independent inspections, in order to remain ISO 9001-certified. Internal audits are part of the continual improvement effort, because during these audits, quality managers will inspect an organisation's conformity to the organisational PM. In addition, external audits are conducted by external, independent organisations to prove the fulfilment of the ISO 9001 requirements. In case of a divergence between the ISO 9001 requirements and the observed actions and practices within an organisation, corrective actions will be necessary. ISO 9001 (ISO, 2008b) states, "corrective actions shall be appropriate to the effects of the nonconformities encountered" (p. 14). In addition, the standard suggests establishing a procedure with distinctive steps for helping the organisation to prevent further repetition of these nonconformities.

As described in the previous section 2.2.3.2, the *structural element* of ISO 9001-certified organisations is its recorded information in the form of documents.

Those are important, because auditors used to decide about certification of organisations on basis of those documents. Moreover, a member of the top management shall have the responsibility to ensure that the quality management system is not only established and implemented, but maintained as well. This top manager is committed to enhancing improvement efforts of the organisational PM and shall ensure that appropriate communication processes are initiated to communicate the quality management system. In conclusion, the monitoring of ISO 9001 regulations and guidelines in an organisation is threefold: documents containing information concerning the quality management system; top manager commitment to manifesting the quality management system; and, communication processes geared toward educating the rest of the organisation about the ISO 9001.

2.2.3.2.2 Changes between ISO 9001:2000 and ISO 9001:2008

For a variety of reasons, ISO standards are revised from time to time. The current ISO 9001:2008 is the fourth edition and replaces the third edition ISO 9001:2000. From a user's perspective, the changes are minimal, because the main goal with this update of ISO 9001 was to enhance the compatibility with ISO 14001 (standard for environmental management systems). In addition, some clarifications to the requirements of ISO 9001 were introduced, but no additional requirements and no change of intent of ISO 9001:2000 were implemented in ISO 9001:2008. For that reason, an organisation certified to ISO 9001:2008 does not imply an improvement, and organisations that are already ISO 9001:2000-certified should be able to receive a new certification to ISO 9001:2008 as well.

All details of the modifications between ISO 9001:2000 and ISO 9001:2008 are given in the standard ISO 9001:2008, Annex B (ISO, 2008b). Moreover, ISO provided a description of the transition between those two standards (ISO, 2008a)(see Figure 9-1 Implementation timetable for ISO 9001:2008). This literature review leaves it at this point, because the case study narrates a story about an ISO 9001:2000-certified organisation. The goal of this brief summary of the changes between ISO 9001:2000 and ISO 9001:2008 is to show the potential for a possible transfer of the research contribution towards ISO 9001:2008-certified organisations.

2.2.3.3 Positive aspects about ISO 9001

The literature review concerning PM revealed obvious positive aspects about the idea of PM and about the ISO 9001 in particular. Actually, the whole idea is very fundamentally simple and basic, because it involves only using common sense (Smith, 2002). This common sense was identified in the eight quality management principles of the ISO 9000 family (cf. section 2.2.3.2: customer focus, leadership, involvement of people, process approach, system approach to management, continual improvement, factual approach to decision making, mutually beneficial supplier relationships). Having written documents on hand pertaining to these principles is also a reasonable method an organisation can use to attain their goals, because the ever-changing nature of business circumstances can be very turbulent and complex.

In addition, the literature review highlighted some practical advantages of an ISO 9001 certification (Dick, 2000; R. Jones, et al., 1997; Jovanovic & Shoemaker, 1997; Rada, 1996; Singels, Ruël, & van de Water, 2001):

a) *Better proximity to customers* – Processes may mature, because ISO 9001 encourages continuous development. Even sensible customer relationships may benefit from continuous developments and a focus on increased customer satisfaction.
b) *Competitive advantage* – When used as a part of marketing initiatives, ISO 9001 certification may imply better product and service quality.
c) *Increased profitability* – The costs for achieving an ISO 9001 certification are usually more than compensation through decreased costs and less waste.

d) *Prevention against product liability* – Continuous record keeping of attributes during the product or service creation may help in the minimisation of liability risks.
e) *Requirement to enter some markets* – Some markets require a PM certificate (e.g. ISO 9001) in order to participate (e.g., the automobile market).
f) *Process standardisation* – This final definition of the procedures of an organisation avoids the vagueness in daily work and therefore is an advantage for managing an organisation. Likewise, a study by Benner and Tushman (2002) determined that using a greater number of process management activities in a firm results in a greater number of innovations that are highly exploitative.

In conclusion, the benefits of a PM framework such as ISO 9001 make it extremely valuable for organisations.

2.2.3.4 Difficulties with ISO 9001

A complete review of the literature revealed some conflicting viewpoints concerning the positive issues of PM frameworks such as ISO 9001 (Dick, 2000; R. Jones, et al., 1997; Jovanovic & Shoemaker, 1997; Singels, et al., 2001):

a) It requires time and effort into the ongoing process of communicating with customers.
b) Creative and critical thinking might be discouraged, because employees are forced to work according to prescribed procedures and rules and therefore, certified organisations have a smaller number of explorative innovations (Benner & Tushman, 2002).
c) Higher bureaucracy from the certified organisation is required to fulfil the requirements of ISO 9001.
d) Seeking certification could incur extra costs, which would make the attempt to be compliant with a PM framework a “hollow achievement”.

To summarise, to initiate and maintain a PM framework within an organisation is neither beneficial nor disadvantageous, and thus it needs to be put into perspective. A common misconception is that ISO will produce higher levels of product and service quality (Motwani, Kumar, & Cheng, 1996). ISO certification gives no promise that the quality of the products or services of a certified organisation will be better than the quality of non-certified organisations. However, ISO certification assures the consistent quality of products, services and processes (Singels, et al., 2001).

It is also important to note that the motive for seeking certification is considered to be an important factor for later performance of the implemented practices and techniques (Dick, 2000; Huarng, Horng, & Chen, 1999). For example, one of the motivations for companies to be certified is that it has been requested by the customers (R. Jones, et al., 1997). One study revealed, organisations that cited client constraint as their motivation for pursuing certification were less likely to report improvement as a result of certification than those which gave other reasons for establishing an PM framework (Dick, 2000).

2.2.4 Summary

The previous sections provide a brief literature review about how process management influence development activities. PM frameworks have enjoyed tremendous success, because many organisations want to adopt an approach to control better their work activities and processes. In doing so, ISO 9001 has been favoured by many organisations worldwide as a framework to certify organisational processes.

In section 2.2.3, I elaborate on the PM framework of ISO 9001, by describing the content. In addition, ISO 9001-certified organisations need to undergo regular audits that are accompanied by constant monitoring activities of the processes within that organisation. The ISO 9001 standard experiences from time to time updates, and the current version dates to the year 2008. The literature review highlights the practical advantages of an ISO 9001 certification, but reveals also the challenges involved with ISO 9001. Benefits of an ISO 9001 certification may include for example competitive advantage, increased profitability, and prevention against product liability (e.g.: Dick, 2000; Rada, 1996). Challenges with an ISO 9001 certification may involve for example higher bureaucracy, extra costs, and discouraged creative thinking (e.g.: Dick, 2000; Rada, 1996).

Besides those aspects of PM activities and in particular of ISO 9001, the idea to prescribe processes had also profound influence in the realm of ISD. Hence, the literature review in section 2.3 reveals various processes, which prescribe how to develop information systems.

2.3 Information system development processes

The metaphor of methodologies is the conventional view of researchers concerning ISD (e.g.: Avison & Fitzgerald, 2006; Forsberg & Mooz, 1997; Ghezzi, Jazayeri, & Mandrioli, 1991; IEEE, 1998b; Sommerville, 2007). Methodologies involve a systemised perception of different stages during the ISD process. This literature review describes some distinctive methodological approaches of ISD and a summary of some variants is presented in section 2.3.1. Section 2.3.2 describes the rationales and limitations of using ISD methodologies in practice.

2.3.1 Information system development methodologies

2.3.1.1 Preliminary remarks

ISD methodologies helped to structuralise the development process of information systems from being a poorly organised pursuit towards a more controlled and stable creation process (Nandhakumar & Avison, 1999). Researchers claim, as a result of using ISD methodologies, developers and managers were enabled to deliver more applications with better quality and lower costs (Nandhakumar & Avison, 1999). Those ISD methodologies comprehend a directed perspective that du Plooy (2002) argues the "development process can be considered a largely rational process, undertaken with the help of various tools and techniques, based principally on the so-called 'systems approach' taken from classical (natural) science" (p. 115). Furthermore, it is argued that the systems approach results of the widespread view of information systems as technical systems, being technically developed and maintained (du Plooy, 2002). Following this directed perspective, Avison and Fitzgerald (2006) defined ISD methodologies as:

> A collection of procedures, techniques, tools and documentation aids which will help the systems developers in their efforts to implement a new information system. A methodology will consist of phases, themselves consisting of sub phases, which will guide the systems developers in their choice of the techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate information systems projects. (p. 24)

In addition, it is important to realise that an ISD methodology is usually based on a specific philosophical perspective that emphasises certain aspects, such as the human being embedded in ISD processes, or the scientific solution-finding process, and so on (Iivari, Hirschheim, & Klein, 1998; Pressman, 1997). The complex task of ISD certainly presents the need for having defined methodologies. Organisations and individuals want to achieve better products through better development processes because they need to operate sustainably. However, having established an effective ISD methodology in an organisation is not a silver bullet (e.g. Pressman, 1997; Sommerville, 2007). Organisations themselves are different, and there is no standard organisation where one ISD methodology will be forever suitable. Even within the development departments, one ISD methodology may not be suitable, owing to the fact that people are different and teams of people are different.

Maybe because of the variety of organisations and their development teams, there are many different methodologies available in the academic literature. In fact, numerous articles and books are available in order to target the different wants and needs of various people concerning ISD methodologies. Some methodologies were derived from previously existing methodologies. Therefore, some people tried to identify patterns within the ISD methodologies with the goal to compare and contrast the different models (e.g. Abrahamsson, Salo, Ronkainen, & Warsta, 2002; Avison & Fitzgerald, 2006; R. Brown, Nerur, & Slinkman, 2004; Cockburn & Highsmith, 2001; Iivari, et al., 1998; Larman & Basili, 2003; McConnell, 1996; Sauer & Lau, 1997; Sommerville, 1996, 2007). For example, those patterns involve investigations concerning ease of management (Sorensen, 1995), suitability for different development activities (Sommerville, 1996), primary objectives (Boehm, 2002), people factors (Turner & Boehm, 2003), and so on. In this literature review, I follow Kettunen and Laanti's (2005) approach to classify the many ISD methodologies. Kettunen and Laanti's (2005) work classified methodologies concerning embedded software development projects, which is the focus of this thesis. This classification approach is a useful and necessary simplification of the available ISD methodologies and it summarises three main strands, namely: specification-based, evolutionary development, and agile development methodologies.

2.3.1.2 Specification-based methodologies

The first published methodology of an ISD process was derived from more general system engineering processes and is called the waterfall model (Royce, 1970). Its name and elementariness is based on the stepwise execution of the different steps from the system life cycle. The different stages of the system life cycle are executed sequentially, and that is the key concept of this methodology. For example, all requirements must be collected at the initial stage of the development for preparing a final specification for the ISD. This procedural method suggests also that the 'plan, control and monitor testing' phase is at a very late stage of the development process and thus it tends to be very risky regarding time and budget of the enterprise. Inevitably, this led to some criticism of the waterfall methodology and the development of new software development methodologies (e.g. Avison & Fitzgerald, 2003, 2006; Boehm, 1986; Sommerville, 1996; Townsend, 2007). A newer specification-based methodology was described as incremental and based on the waterfall methodology. However, the new methodology provided an overlapping relationship of practices and activities between the life cycle stages of the ISD (Mills, O'Neill, Linger, Dyer, & Quinnan, 1980). For example, it emphasised that the quality of the process outcome must be built into designs and cannot be inspected in or tested in. The system functionality is separated into several increments, and these are developed and handed over to the customer one by one. Parts of the specification may change, but specified parts of the incremented pieces of development cannot change. Therefore, it is possible to negotiate and experiment with the users significantly longer by using the original waterfall methodology. Typically, specification-based methodologies are applicable to ISDs consisting of various subsystems. However, the specifications and architecture of the system need to be defined relatively early in the development process. The straightforward nature of the specification-based methodology keeps the challenges for project management on a moderate level. Nevertheless, the fundamental difficulty with this methodology is that stakeholders have to articulate their requirements early and in advance, and this characteristic prohibits any late changes during the ISD process.

2.3.1.3 Evolutionary development methodologies

In order to overcome the late change rigidity, evolutionary development methodologies evolved (Sommerville, 1996). Evolutionary development methodologies are based on the idea of developing an initial implementation, introducing this into the stakeholder discussion and refining it through many versions until the evolution degree has achieved a satisfactory information system (Boehm, 1986; Sommerville, 1996). There are three generic stages in an evolutionary process; first is the specification of the scaffolding of the information system enterprise. This scaffolding specification may guide the developers concerning what the system should do and therefore, the initial specification does not need to be complete or consistent. Second, the information system is created, assembled and coded based on the scaffolding specification. Third, the system is continually modified until the stakeholders of the system agree about its satisfactory performance. This evolutionary development methodology put many emphases on the wants and needs of the customer. If the customer does not understand well enough, it might be better to throw away the prototype of the initial evolution in order to get a better result at the end of the ISD process. Some development undertakings may have the objective to explore an ISD by adding new features on a known information system. Critics of these kinds of evolutionary development methodologies claim that serious organisational requirements may not be given adequate priority (Sommerville, 1996). Change management becomes a critical task throughout the ISD process, and the project management is more difficult, particularly in developments of larger information systems (Sommerville, 1996). A good example of an evolutionary development methodology is the spiral model of the software process (Boehm, 1986). This methodology is developed around that evolutionary development idea, with an explicit recognition of risk. Conducting risk analysis throughout the different spiral cycles will help to reduce the negative influences on the ISD process.

2.3.1.4 Agile development methodologies

The increased need for speed and flexibility required a new methodology to develop information systems. In the 1990s, a wide range of publications dealt with the so-called agile development methodology, which was unlike any methodology that had been used previously (Abrahamsson, et al., 2002; Baskerville & Pries-Heje, 2004; Cockburn, 2001; Cockburn & Highsmith, 2001; Katayama & Bennett, 1999; Manhart & Schneider, 2004). The key concept is summarised in the 'Agile Manifesto' (Fowler & Highsmith, 2001) and it involves four main aspects: First, the requirement elicitation process is an ongoing process with high involvement of stakeholders (customer, business people, and developers) and open communication channels. Second, the code and its structural settings are based on simplicity and technical excellence. Third, the work output is the primary measure of progress, and the highest priority is to satisfy the customer with valuable software on a frequent basis. Finally, agile development needs a supportive environment to motivate the individuals and reflect on the effectiveness of actions and practices at regular intervals. This structural foundation of the 'Agile Manifesto' requires several practiced routines by the software developers and other stakeholders, such as project managers. Probably the best-known agile method is Extreme Programming (XP) (Beck, 1999; Beck, Fowler, & Kohnke, 2001). The methodology is called extreme because recognised good practices, such as iterative development, pair programming (Cockburn & Williams, 2001), and customer involvement, are carried out to 'extreme' levels. For example, some core practices of XP are as follows: the customer defines application features with user stories; development teams frequently revise and edit the overall code design (a process called refactoring); a customer representative remains on-site throughout the development. Something that agile development methodologies have in common is that the distinctive system life cycle steps are eliminated, such as lengthy requirement definition stages. In addition, agile development methodologies remain in constant communication with the customer and provide work output early. However, there are also some weaknesses by using agile development methodologies, such as limited support for distributed development environments, subcontracting, building reusable components, development involving large teams, and development of safety-critical systems (Turk, et al., 2002). Therefore, Turk et al. (2002) state that agile development methodologies may be not suitable for developing large and complex information systems.

2.3.1.5 Comparative table of ISD methodologies

The relevance of varies ISD methodologies changed over time, in order to cope with different environmental challenges. To remain consistency in this thesis, I will keep following Kettunen and Laanti's (2005) approach to classify ISD methodologies and describe whose unique procedures to initiate, execute, and complete ISD projects. Table 2-3 provides a comparison of those procedures of the three main strands of ISD methodologies.

Table 2-3 Comparison of ISD methodologies

ISD methodology	Specification-based development	Evolutionary development	Agile development
Key references	(Avison & Fitzgerald, 2003, 2006; Kettunen & Laanti, 2005; Mills, et al., 1980; Royce, 1970; Sommerville, 1996; Townsend, 2007)	(Boehm, 1986; Kettunen & Laanti, 2005; Sommerville, 1996)	(Abrahamsson, et al., 2002; Baskerville & Pries-Heje, 2004; Beck, 1999; Beck, et al., 2001; Cockburn, 2001; Cockburn & Highsmith, 2001; Fowler & Highsmith, 2001; Katayama & Bennett, 1999; Kettunen & Laanti, 2005; Manhart & Schneider, 2004; Turk, et al., 2002)
Phase: initiation	Specification-based methodologies remain on the specification phase, until the project goals are defined. The definition involves a comprehensive clarification of problems that the development faces. Specification-based methodologies cannot move to the next phase until the phase initiation is completed and defines project milestones. Therefore, specification-based methodologies assume a mature, stable environment. However, fluctuating environmental circumstances makes it difficult to pre-plan everything, particular in situations with high change.	Evolutionary development methodologies start working on the known parts, and define the rest later. Therefore, the level of detail evolves throughout the evolution of the development project. The first evolution steps involve simpler version and continue to become more sophisticated versions. Uncertainties and areas with high risk should be taken care of early, implying a focus on the feasibility risks first. The initiation, implementation, and testing cycles follow each other. The number of cycles needed is difficult to estimate because changing circumstances may require additional evolution stages. Major delays can be avoided because the evolutionary methodology emphasis is on early critical issues to avoid extensive reworking.	Agile development methodologies establish a strong connection with the customer. The developers work in continuous coordination with the customer frequent delivery of product samples. This iterative product development welcomes an uncertain and fluctuating environment. Planning sessions followed implementation, and testing phase. All iterations involve evaluating and re-planning of the completed release, in which the customer is always involved. Agile development methodologies are appropriate in manageable projects, e.g. extreme programming suggest not more than 10 programmers. The embedment of the customer in the development open avenues for open communication between the customer and developers. Hence, problems may be tackled outright.
Phase: execution	By model definition, the execution starts, when specifications were clarified. Uncertainties should be resolved at the first stages. Therefore, any changes after the phase of initiation would create trouble and are difficult to manage in the development process. Specification-based methodologies frankly integrate the whole system at once, which may reasons integration problems. In addition, the product becomes available at the final stage of the development process. Any progress can be monitored through the increasing documentations, produced at several stages of the development, and the fulfilment of the different milestones, specified at the initial stage of the development. Although the amount of documentation increases considerably, tacit knowledge might be difficult to transcribe and pass from one development phase to another.	Critical areas intended to be identified early and changes can be incorporated for next cycles. Therefore, the goal of evolutionary development methodologies is a focus on finalise the requirements timely in order to set the development boundaries and keeps it focused. Risks are reduced with every evolution, and each cycle completion is a milestone. Furthermore, the execution phase involves continuous integration and testing at the end of evolutionary steps. Evolutionary development methodologies entail an evolving documentation.	Agile development methodologies strive for a strong connection with the customer, and therefore any progress becomes visible instantly. The progress may be determined through completed features, which serve as milestones. Changes are an aspect of the development process and are incorporated in short development cycles. Agile development methodologies are based on frequent communication between the various stakeholders; consequently any communication gaps in the development process are ruinous. In addition, documentation is not as important as working software.
Phase: completion	Specification-based methodologies define end-criteria at the initiation phase. Part of the integration stage is fully tested software, in order to release software according to the specification. Depending on the quality of that specification, the final product is appropriately.	The goal of evolutionary development methodologies is to create products with growing quality after each evolutionary step. This involves a potentially difficult situation, because in many cases, the product could always be improved. Therefore, certain end-criteria need to be specified in order to meet satisfactory quality with adequate business goals.	Agile development methodologies try to improve the product quality gradually. Development activities are conducted in continuous coordination with the customer, and the customer decides if a product meets the desired features and quality attributes.

2.3.2 Information system development methodologies in practice

2.3.2.1 Rationales for using a methodology

A great part of ISD literature concerns methodological development and numerous articles and books provide many different methods specifying prescriptive instructions on how to develop information systems (Kautz, et al., 2004). Avison and Fitzgerald (2006) summarise a number of arguments for adopting and using ISD methodologies. They identified three main categories of arguments:

a) A better end product,
b) A better development process, and
c) A standardised process.

Developers and managers may want a methodology to improve the end product of the ISD process; that is, they want better information systems. Therefore, advocates of methodologies want to improve the quality of information systems, but the quality aspects of an information system that should be improved varies between organisations and individuals. Indeed, some methodologies work against each other and therefore, the most appropriate ISD methodology needs to be chosen (Avison & Fitzgerald, 2006). The second argument for adopting a methodology concerns the development process. To improve the development process on the basis of a defined ISD methodology results in stronger management and project control (Avison & Fitzgerald, 2006; Avison, Powell, & Adams, 1994). The third argument proposed by Avison and Fitzgerald (2006) states the rationales of standardised processes. Indeed, adopting and using an ISD methodology can result in benefits that are very similar to those of process management frameworks (Avison & Fitzgerald, 2006)(cf. section 2.2.3.3). Such benefits may involve a common knowledge basis as to how ISD is conducted effectively and efficiently throughout the organisation.

2.3.2.2 Limitations of using methodologies

Although methodologies may help to develop information systems better, the literature review seems to indicate that in practice the indented outcome are not always achieved. For example, Nandhakumar and Avison (1999) describe how an ISD methodology served as a symbol to provide the fiction of system development as a manageable process, but in practice it was too formal and structured to be of any assistance in the development process. Likewise, Wastell (1996) stated that the use of a methodology served as a social defence against involvement in the real and complicated task of developing information systems. Moreover, research indicates about the practice of ISD methodologies that practitioners do not fully accept proposed methodologies and there is evidence of problems with the use of these methods (Fitzgerald, 1998; Nandhakumar & Avison, 1999). Therefore, ISD methodologies in practice seem not to provide universal approaches to assist the management and development of information systems, because in practice, the development process is influenced by unpredictable events and other conditional factors. Therefore, methodologies are unable to deal with unpredictable conditions. In addition, methodologies are also used for purposes that are not intended by the methodology authors.

Those unpredictable events and other conditional factors occur in turbulent environments, where the application of a structured ISD methodology may fail to be successful (du Plooy, 2002). Moreover, agile methodologies that seem to cope in turbulent environments better, provide only limited support for the challenges in many development environments (Turk, et al., 2002). Du Plooy (2002) states that only in very closed organisations can standards and exact methodologies be enforced to develop information systems accordingly. However, ISD methodologies tend to be too rigid, while an unbroken chain of previous events does not causally determine the products of those development efforts. Consequently, in turbulent environments it may be hardly possible to achieve to overcome the difficulties of constantly changing circumstances. Because of those tendencies in ISD, advocating and using process management frameworks and traditional ISD methodologies have not done much to reduce the failure rate of information systems (du Plooy, 2002).

Furthermore, many researchers (e.g. Fitzgerald, 1997; Nandhakumar & Avison, 1999) call for more field studies that focus on the process of ISD in its real organisational context and how ISD methodologies are practiced. One of the few studies that have been conducted is the research of large ISD projects by Curtis et al. (1988), which concludes that prescribed development tools and practices had a small effect, because a number of shortcomings. Those shortcomings involve less application domain knowledge, fluctuating and conflicting requirements, and communication difficulties during the ISD process. Fitzgerald (1998) investigated the adoption of ISD methodologies and found that the same methodologies are applied differently in varying projects by developers and managers. Moreover, Fitzgerald (1998) states that methodologies are too general without any solid foundations and contextual considerations, and too strong emphasis on technical aspects at the expense of social aspects, individual creativity, or learning over time.

The emphasis on technical aspects is generated through an overemphasis on various tools, techniques, and methodologies (du Plooy, 2002; Newman & Robey, 1992). Steen (2007) states that software development is an inherently human practice, which cannot be reduced to the adherence to methods and use of techniques. However, following the conclusions of some ISD researchers and understanding ISD as social activities - culture, social relationships, decision processes, and so on – would suggest that ISD is continually emergent, and it would follow no predefined pattern (Truex, et al., 1999). That emergent character is underpinned by the inherently human development activities in analysing, designing, and programming of complex systems, when for example developers and managers need to change something. Ciborra (2002) described those activities that involve change and unfolding as bricolage (defined as: making do with the items at hand (Ciborra, 2002; Lévi-Strauss, 1966)). Moreover, emergent activities in that context are also described as improvisation (e.g.: Bansler & Havn, 2005; Ciborra, 2002; Nandhakumar & Avison, 1999; Verjans, 2005). Naturally, creativity is an intrinsic aspect of improvisational and bricolage activities. In addition, bricolage and improvisation do not imply remoteness concerning the effective and efficient performance of an organisation. In fact, Weick (2003) describes a relationship between organisational reliability and improvisational and bricolage activities (cf. section 3.7.1).

Creativity is important in all aspects of ISD (e.g.: Cooper, 2000; Couger, Higgins, & McIntyre, 1993; Fayad, 1997; Fielden & Malcolm, 2006; Gallivan, 2003; Glass, 1995; Gu & Tong, 2004; O. Hoffmann, Cropley, Cropley, Nguyen, & Swatman, 2005; Plsek, 1997). However, Aaen and Pries-Heje (2004) claim that traditional ISD methodologies can jeopardise or hamper creativity and innovation, because they do not address social issues. For example, writing code is a task with many avenues that can lead to a similar goal. This indefiniteness is a natural aspect of a developer's creativity and is contradictory to the mentality of managing an organisation and its members (Crutchfield, 1962). This obvious tension between the efforts to manage and to be creative has been described in the literature extensively (e.g.: Ciborra, 2000; Crutchfield, 1962; Cummings, 1965; Cunha & Cunha, 2006; D'Agostino, 2006; Dvir & Lechler, 2004; Eaglestone, Ford, Brown, & Moore, 2007; Edmonds & Candy, 2002; Gallivan, 2003; Sawyer & Guinan, 1998). In addition, Kirton (1976) found two different styles of creativity: *innovators*, who tend to possess high originality, low efficiency, and low rule-conformity; and *adaptors*, who have less originality, high efficiency, and high rule-conformity. Each style has its advantages and disadvantages (Gallivan, 2003), but it is certainly the adaptor who is more concerned about the commonness of innovations within an organisation (Benner & Tushman, 2002). In coping with this concern, the adapting developers can rely on prescribed processes that help to distribute innovations effectively. Innovators, with their low rule-conformity, might be of little help in the effort to increase effectiveness and efficiency in the entire organisation. Therefore, an organisation does not necessarily require that all employees be innovators (Gallivan, 2003).

Truex et al. (2000) questioned the assumptions behind the prescriptive methods, because they do not resemble how system development is conducted in practice. Therefore, Truex et al. (2000) describe the amethodical system development to present the limitations of the literature concerned with methodological ISD. Amethodical system development involves the management of ISD without completely predefined structure, sequence control, rationality or claims for universality. However, this does not imply chaotic, anarchic, and methodless approaches to ISD. Truex et al. (2000) understand the amethodical ISD as a unique, negotiated and opportunistic process driven by accident. Empirical evidence that describe ISD activities without using formalised methodologies or where only certain aids are utilised, support the point of view by Truex et al. (Fitzgerald, 1997, 1998; Nandhakumar & Avison, 1999; Truex, et al., 2000; Wastell, 1996). In addition, Fitzgerald (1998) report that developers adapt and apply methods in a pragmatic way and that the "most evident contribution seems to be as a framework for the use of tools and techniques, as they do not appear *per se* to affect the allocation of time to various development activities" (p. 11).

Because of the limitations of ISD methodologies in practice, Madsen and Kautz (2002) doubt whether it is feasible to incorporate all activities performed in systems development in one methodology. Moreover, Hansen et al. (2003) state that methodologies are adjusted in action and no universally applicable methodology exits. That finding confirms Brooks' (1987) conclusion that a 'silver-bullet' does not exist for the complexity of ISD. It is interesting that 15 years later, researchers obtained the same conclusion concerning the limitations of agile methodologies (Turk, et al., 2002). Although those findings about the ISD methodologies in practice indicate many difficulties, it is also suggested that methodical as well as amethodical system development views must be kept in mind, when engaging in development activities (Kautz, et al., 2004; Truex, et al., 2000). Madsen (2002) states that the challenge is to find an equilibrium between the methodical and the amethodical aspects of ISD. That is supported by findings of Stolterman (1991), he concludes that the irrationality of developers practice must be addressed by new methods that will be accepted and understood by practitioners (Kautz, et al., 2004). Moreover, the symbolic use of methodologies gives developers and managers at least some comfort, confidence and belief to be capable of developing large and complex information systems (Kautz, et al., 2004; Nandhakumar & Avison, 1999; Wastell, 1996), because methodologies provide distinctive sets of instructions and guidelines for the stakeholders of ISD undertakings that help with complex ISD processes and requests.

In summary, the literature review indicated various limitations of traditional ISD methodologies. Mainly because organisations are not mechanistic, but are influenced by other factors (social relations, politics, and cultures) and ISD projects may be influenced through imponderability of business circumstances (e.g.: hardware proclamation, changing software tools, and so on). Accordingly, to achieve the full compliance with managed processes, methodologies, and techniques in the realm of ISD turn out to be extremely difficult.

2.4 Development of embedded systems

The development of embedded systems combines the realms of software and hardware and therefore is a relatively complex sub-domain within the ISD domain (Graaf, et al., 2003; Kettunen & Laanti, 2005; Lee, 2000, 2002; Taramaa, et al., 1998). First, application examples of embedded systems are explained in order to familiarise the reader with the significance of this domain. Second, contextual challenges of the embedded system development area are described based on a brief literature review of this field.

2.4.1 Applications of embedded systems

All embedded systems are dedicated to one or more specific tasks; these can include the entire technological spectrum, from deeply embedded applications in specific hard real-time systems to software intensive applications. For example, embedded systems are included in today's consumer electronics (mobile phones, personal digital assistants, cameras), cars, airplanes and audio equipment. Especially in vehicles, customers expect more gadgets than ever before (e.g.: Kibler, Poferl, Böck, Huber, & Zeeb, 2004; Powers & Nicastri, 2000). Most of these software applications have taken over what mechanical and dedicated electronic systems used to do (Goossens, et al., 1997; Lee, 2002). In addition, many systems have moved from analogue to the digital domain, such as the control systems of cars (Cena, Valenzano, & Vitturi, 2005). Embedded systems generally also have several modes of operation, must be able to respond quickly to actions, and must possess a great deal of concurrency (Gajski & Vahid, 1995).

It is interesting to note that Mr Al Gore[1], former vice president of the United States of America and current environmentalist, encouraged embedded system developers during the Embedded Systems Conference in 2007. Gore explained in his keynote address that greater intelligence in embedded systems – from car engines to decentralised power stations – could contribute crucially to the "healing of our planet." Higher achievement and efficiency with lower energy consumption must become the central construction principle. He stated that the collective intelligence of these smart embedded systems would contribute

---

1 Bonnert, E. (2007). Al Gore feuert Embedded-Entwickler an: "Helft dem Planeten!". *Heise Online* Retrieved April 4th, 2007, from http://www.heise.de/newsticker/meldung/87831

decisively to the improvement of the environmental balance of the information technology installed worldwide.

2.4.1.1 Automotive environment

Complex embedded systems have become an important part of modern automobiles (Crotty, 2005; Mello, 2005; Powers & Nicastri, 2000). Premium cars have anywhere between 50 to 70 different electronic control modules along with their sensors and actuators and all these control modules needed to exchange information with each other[1]. Electronic bus systems (e.g.: Controller Area Network (CAN), Media Oriented Systems Transport (MOST), Flexray) are perceived as promising solutions that have been used in the automotive context (Cena, et al., 2005; Kibler, et al., 2004). Several business analysts (Crotty, 2005; Frank, 2005) state, infotainment solutions (navigation / GPS systems, Bluetooth devices, DVD entertainment systems) become increasingly central in the product design of automobiles. For example, integrated solutions to connect MP3 devices to a modern car are nowadays part of the expectations of many customers (Miel, 2005; Morgan, 2005; Prasad, 2005). Embedded development in the automotive environment faces not only technical challenges, but also the management of the manufacturing lifecycle is a difficult exercise (H. Hoffmann, et al., 2007). In addition, automotive safety must be considered in the design of embedded systems (H. Hoffmann, et al., 2007). Consequently, the integration in the network of embedded systems is the actual challenge in developing an automobile, because embedded systems involve different pieces of software (Broy, 2006). Researchers[2] state, standardised operating systems in the automobile environment may contribute in integrating embedded system

---

1 Jena, C. (2007). The IT-enabled automobile. *Express Computer* Retrieved March 17th, 2008, from http://www.expresscomputeronline.com/20070219/management01.shtml

2 Löwer, C. (2008). Einheitliches Betriebssystem für alle Autos. *Handelsblatt* Retrieved April 18th, 2008, from: http://www.handelsblatt.com/technologie/technik/einheitliches-betriebssystem-fuer-alle-autos;1364233

better into the network of the automobile, in order to increase speed and quality, and decrease costs (Marsh, 2005).

2.4.2 Contextual challenges of embedded systems development

Embedded system development is fundamentally different to non-embedded systems because it addresses specific constraints such as hard timing constraints, limited memory and power use, predefined hardware platform technology, and hardware costs (Graaf, et al., 2003; Kettunen & Laanti, 2005; Lee, 2002). For example, embedded systems often have to perform tasks within a time slot defined by, e.g., a telecommunication standard (Ronkainen & Abrahamsson, 2003). Embedded system developers must understand interdisciplinary product application domain knowledge (Curtis, et al., 1988) and physical requirements such as memory, power, and real-time issues (Graaf, et al., 2003; Lee, 2000, 2002). In addition, many business constraints must be taken into account in industrial new product development environments (Manhart & Schneider, 2004). For example, changing requirements often cause delays in the project schedule, which can cause bottlenecks in embedded system development projects, where for example the hardware development is incomplete and when the software is ready for testing (Shehabuddeen, Hunt, & Probert, 2002).

Embedded systems are usually complex; change in one component (hardware or software) usually causes a change in another (Shehabuddeen, et al., 2002). From a development viewpoint this means that the requirements for components cannot be frozen before development work begins (Ronkainen & Abrahamsson, 2003). Therefore, the development method, by necessity, has to have some kind of mechanism to cope with changes in requirements during development (Ronkainen & Abrahamsson, 2003) and those changes involve technical (hardware and software) and social issues (Coulin, Zowghi, & Sahraoui, 2006; Jiao & Chen, 2006; Nuseibeh & Easterbrook, 2000). Clearly, requirement elicitation and management is highly dependent on the specific project, as well as organisational and environmental characteristics and circumstances (Christel & Kang, 1992). Consequently, some ISD methodologies are less easy to apply on embedded system development, because the hardware requirements and design are incomplete at the beginning of the project, which leads to requirement changes on software created by hardware (Mäkäräinen, 2000; Ronkainen & Abrahamsson, 2003). For example, the recommended guideline for software requirement specifications by the Institute of Electrical and Electronics Engineers (IEEE)(1998a) provide for embedded system development only limited help, because issues beyond those identified have to be addressed. Therefore, the changes in the hardware design and interfaces are often adapted with software, since changing the software is cheaper and easier (Greene, 2004). After the hardware is ready, fixing and tuning the system are made with software as the needs are discovered, which again means unexpected requests and changes (Greene, 2004). Consequently, functionality has steadily shifted from hardware to software (Lee, 2000). This paradigm shift from hardware to software has several consequences. For example, it is possible to compile late software specification in response to fluctuating requirements. The use of software also facilitates the reuse of previously-designed functions if the code was written at a processor-independent, abstract level (Goossens, et al., 1997).

2.5 Research aim and questions

The literature review concerning process management and information system development activities reveals a complex picture. Section 2.2 reviews the literature about process management and development activities, and to sum up, process management efforts aim to shift the ad hoc processes to a more manageable approach. This thesis focuses on the quality management framework of ISO 9001, because it is an important standard in process management and relevant for the later analysis of the case study. Subject of section 2.3 are ISD processes. A large part of ISD literature concerns methodological development that aims to provide guidelines to develop information systems systematically, similar to the efforts of process management literature. However, ISD is also influenced by non-technical factors (e.g. social relations, politics and cultures). Therefore, it is extremely difficult to follow those guidelines to develop information systems and only few researchers highlight those limitations of ISD methodologies in practice (e.g.: du Plooy, 2002; Nandhakumar & Avison, 1999; Truex, et al., 1999; Wastell, 1996). The practice of ISD involves improvisation and bricolage activities, which are opposite to prescribed processes. An interesting and complex topic within the ISD domain is the development of embedded systems. However, the development of embedded systems has benefited less from the well-researched concepts of ISD, because the physically engaged world of embedded systems is different from the abstract way of thinking in the non-embedded ISD literature. Therefore, guidelines that aim to structuralise development activities have not done much to improve the embedded system development in practice.

2.5.1 Identified knowledge gaps in the literature

The literature review concerning the development of embedded system indicated that this sub-topic within the ISD domain has received less attention on non-technical issues. The thin spread of available in-depth investigations of embedded system development settings was reasoned to conduct more research in that area. Particularly the research on embedded software development lacks detailed studies and gives rise to this thesis. Moreover, the literature review identified a knowledge gap within the field of embedded software development. Given that available literature emphasises how embedded software *should* be developed, we may well expect the empirically grounded writings focus on how it *is* developed. However, my literature review revealed little research that aims at the particular situation of how embedded software gets developed in practice.

Certainly, to develop an embedded system successfully and systematically is a difficult task and requires a better understanding to support organisations in coping with the difficulties. In addition, development activities tend to involve improvisational and bricolage actions and practices that are hardly capable of being controlled and managed. This is a challenge, because organisations want to maintain the ability to supervise their employees in order to be effective and efficient, hence they try to apply process management frameworks. But those process management frameworks seem to be of little help. In addition, the present literature provides less procedural help to support or manage the dynamics of improvisation and bricolage during embedded system development. Owing to the fact that embedded systems do more and influence people’s daily life more, its development circumstances should be better understood than the current state of literature can provide.

In summary, there are two knowledge gaps identified in the literature:

a) A description about how embedded software gets developed in practice.
b) A theoretical conceptualisation that aids developers and managers to understand the process, overcome difficulties, and make use of opportunities to produce innovative embedded software.

2.5.2 Focus of this thesis

The previous section revealed a knowledge gap in the literature concerning how embedded software development is practiced. In addition, the literature review indicated that development practices may involve improvisation and bricolage activities, which tend to oppose prescribed guidelines. Moreover, the literature review of process management (section 2.2) and ISD processes (section 2.3) suggest that given guidelines have a tendency to be too directive, formal, rigid, and structured. Therefore, I seek to answer following research question in trying to close the first identified knowledge gap:

Research question 1: How do developers and managers practice improvisation and bricolage in the context of process management during embedded software design?

The relation between process management and improvisational and bricolage activities is contradicting (section 2.3.2.2). Therefore, proposed practices of process management frameworks might be of little help for those organisations that involve many improvisational and bricolage activities in their day-to-day processes. In addition, the second identified knowledge gap in the literature point out that guidelines how to overcome practical difficulties during embedded system development are not available. In fact, I found only two references that concern about issues how to improve and manage improvisation, but both miss the contextual relationship to ISD and embedded system development in particular. Weick (1998) refers to issues how to improve improvisation, but lacks in suggestions how to manage and control improvisation. Vera and Crossan (2004) refers to lessons learned how to manage improvisation in an organisational context. However, those findings were not investigated in the embedded system development and therefore, the second research question is:

Research question 2: What measures help to balance the tension between process management and improvisation and bricolage activities in the context of process management during embedded software design?

Because this thesis tries to address the revealed knowledge gaps in the literature by answering research questions 1 and 2, it is worthwhile to reflect on the findings and derive implications for theories and models of system development. The related research question is:

Research question 3a: What are the implications of the findings for theories and models of system development?

This thesis is founded on the analysis and discussion of empirical data from a development project in an organisation in the automobile sector. Hence, the findings have some relevance for people, who are involved in that economic sector. However, beyond that setting, this thesis offers implications of the findings that are significant for the development of embedded system, by asking the research question:

Research question 3b: What are the implications of the findings for the management of embedded system development processes?

By addressing these questions, I want to enrich the understanding of improvisation and bricolage in the context of embedded ISD.

2.5.3 Summary

The literature review has:

a) Identified two knowledge gaps in the literature on the subject of scholarly research in how embedded system development is practiced;
b) Adjusted the focus of this research by articulating the research questions; and
c) Reviewed topics in the literature (PM: ISO 9001; ISD processes: methodologies, ISD methodologies in practice) that substantiate the analysis and discussion of this research.

The literature provides the perspective of this investigation, which will be developed during the theory manifestation of this PhD research based on the topics drawn from the data.

3 Research Approach

3.1 Introduction

This chapter describes the research approach that was used in the design of this study and in the collection and analysis of data. First, it provides an overview of the literature supporting my epistemological assumptions and my choice of methodology, including techniques of data collection and analysis. Section 3.2 then begins with a description of the epistemological perspective of interpretivism and its application in ISD research. The interpretive approaches that I have adopted are illustrated in section 3.3, and this is followed by an explanation of the approach to data collection in section 3.4. An introduction to the case study site is given in section 3.5. Chapter 4 details the case study data. Section 3.6 describes the modes of data analysis used in this research. The theoretical lenses that I used to guide the analysis and interpretation of the data are detailed in section 3.7. Section 3.8 explains the format of data presentation in the case study chapter. Finally, section 3.9 summarises this chapter into a brief overview of my research approach.

3.2 Epistemological position

The information systems research paradigms may adopt three different epistemological positions (Chua, 1986; Klein & Myers, 1999; Orlikowski & Baroudi, 1991): a) Positivist stance; b) Critical stance; and c) Interpretive stance.

I adopted an interpretive position, because this epistemology emphasises the complexity of sense in any given situation or event (Klein & Myers, 1999). Moreover, this epistemology emphasises on the socially constructed reality and understanding in practice from the participants' perspectives (Klein & Myers, 1999) - which is an important part of my research. Interpretive researchers comprehend these situations and events, and their social outcomes, as not permanent, but part of a social process that reflects individual awareness and understanding (Burrell & Morgan, 1979). Interpretive research seeks to understand both the meanings that people assign to phenomena, and how participants contribute to social processes (Orlikowski & Baroudi, 1991).

According to many surveys (e.g. Chen & Hirschheim, 2004; Nandhakumar & Jones, 1997; Orlikowski & Baroudi, 1991) the interest in interpretivism is growing for information systems research. However, information systems researchers often adopt a positivist stance (Chua, 1986; Klein & Myers, 1999; Orlikowski & Baroudi, 1991). In positivist research, researchers typically test theories that can be used to predict general effects of a variable or variables. Accordingly, any knowledge claim or scientific explanation must be derived from quantifiable data and relationships between variables are comprehended as objective. In addition, some information system researchers adopt a critical stance (Chua, 1986; Klein & Myers, 1999). The critical stance seeks to questions the status quo by uncovering inherent, structural conflicts within social organisations. Then, the critical stance seeks to overcome those contradictions, by eliminating the causes of the conflicts. However, the research objective of this thesis does not try to question the status quo or to obtain underlying conflicts (Orlikowski & Baroudi, 1991). Rather, the objective of this research is to develop a bottom up conceptualisation, derived from the gathered understanding through the collection of meanings from research participants. Therefore, the interpretive stance is more appropriate for the research issues to be explored in this thesis (dynamics of improvisation and bricolage during embedded software development; measures that balance the tension between improvisation and bricolage activities and process management; implications of the findings for theory and practice of embedded system development).

The research aim previously stated in section 2.5 involves studying the practices of improvisation and bricolage, and what helps to balance the tension between process management and improvisational and bricolage activities. In order to address the research questions, it is essential to study the embedded software development process in-depth (cf. Geertz, 1973). In doing so, I develop a bottom-up conceptualisation that is obtained from the understanding by research participants. The interpretivist paradigm can provide an in-depth understanding of the examined phenomenon, through comprehension of the internal processes associated with social action (Chen & Hirschheim, 2004; Hirschheim, 1985; Lessnoff, 1974) and by retrieving the meanings that research participants have assigned to them (Orlikowski & Baroudi, 1991; Walsham, 1995). Walsham (1993) describes the interpretive method of research as: "producing an understanding of the *context* of the information system, and the *process* whereby the information system influences and is influenced by the context" (p. 4-5). Orlikowski and Baroudi (1991) discuss the interpretive position that individuals construct and connect their own personal and interpersonal meanings as they cooperate with the environment. That cooperation takes place in the form of an iterative research approach, which includes critical reflection on the social and historical circumstances of the research setting. Thereby, researchers can place the current situation in its cultural and temporal contexts. The iteration occurs when the researcher adjusts their perception (based on reflection on these contexts) of the situation or event under investigation. New knowledge gathered by the researcher may require adjustments to be made that improve interpretations. Indicators of progress are sophisticated interpretations that contribute to awareness of the content and meaning of the research phenomenon (Guba & Lincoln, 1994). The derivation of potentially valuable interpretations must take into account the environment that is under study. This 'contextual grounding' is a feature of the approach that provides it with higher practical relevance than the other epistemological foundations (Khazanchi & Munkvold, 2003).

This 'contextual grounding' enables such investigations to be much more inclusive of circumstantial features of the studied phenomena than other research paradigms. In addition, the researcher has a different relationship with the research context during the gathering of interpretive data. For example, the researcher may collect data as a participant-observer, an interviewer, a consultant, and so on (cf. section 3.4). Appropriate research findings can be derived from contextual grounding that involves the researcher in the data, through a process of abstraction and generalisation (Klein & Myers, 1999). This process has its roots in the phenomenology of Heidegger (1962) and Husserl (1931), who extracted general concepts from everyday experiences. Such attempts to abstract and generalise the data are an important feature of the interpretivist approach (Klein & Myers, 1999). Moreover, interpretive researchers can apply theoretical frameworks as "sensitising devices" or "theoretical lenses", in order to view the world from a certain perspective (Walsham, 1993). In the realm of information systems research, interpretive researchers tend to generalise to social theories such as Structuration Theory (cf. section 3.7.2) (Klein & Myers, 1999).

In conclusion, the interpretive approach emphasises the following aspects in relation to the research described in this thesis:

a) An in-depth understanding of the social activities of developers and managers in the organisational context.
b) Greater comprehension of the social processes (such as creative activities, non-technical processes, and inherent human practices) of developers and managers in the embedded development setting.
c) Filtering of individual differences between research participants in order to comprehend the full variety of social activities.

d) A contextually grounded study of the social activities of embedded system developers and managers in a structured process management environment of ISO 9001.
e) Sophisticated abstraction and generalisation that may be based on a social theory.

3.3 Interpretive research approach

The interpretive research approach of this thesis involved a level of immersion in the field in order to familiarise myself with the wider contextual forces. For example cultural ethnography involves that level of immersion in the field, where long periods of observation provide rich data (e.g.: Evans-Pritchard, 1940; Malinowski, 1922). Ethnographic research is also a comprehensive method in information system research (Myers, 1999). The ethnographer is able to understand the actions of the individual, group, and organisation within the broader context. Using the ethnographic research approach, the researcher is able to collect information on the human, social and organisational aspects of information systems (Myers, 1999). In addition, ethnographic researchers may present their data using case studies. Walsham (1993) introduced the notion of the 'in-depth case study', in which data are collected over a given period through direct participant observation. These studies focus on analysis of the context of the process. At a local level, social structures are another valuable source of data for the context/process researcher.

The ethnographic in-depth case study research approaches allowed me to focus on the complexity of human activities as situations and processes in the research context emerged. Indeed, some complex situations necessitate the study of human activities as they occur and the ethnographic in-depth case study method supports this. The study of previous events or situations (such as improvisation and bricolage) in the business environment is distorted by retrospective justification and the actors' reconstitution of previous organisational circumstances (Cunha, 2005). The research approach accommodated my understanding of the wider context of the complexity of embedded development activities in the German automobile business.

3.4 Data collection approach and process

The decision to utilise interpretive ethnographic and in-depth case study research approaches necessitates field research at an appropriate location. In this case, I needed to conduct research at an organisation with embedded development activities and an established process management framework. It was also essential that I have direct access (e.g. through interviews, observation, and so on) to the actors involved in the development activities, such as developers and managers. In addition, it was necessary to investigate process management in order to comprehend the desired target procedures of the organisation. I found an organisation that was able and willing to accommodate my research. Therefore, I was able to conduct in-depth interpretive research with sufficient access to the knowledge and perceptions of actors (Altheide & Johnson, 1994; Nandhakumar & Jones, 1997).

My level of access to research-relevant data allowed me to utilise a variety of data collection methods. Nandhakumar and Jones (1997) have identified the various interpretive data collection methods and described the different levels of engagement between the researcher and the subject of research for each (cf. Figure 3-1). Distant methods refer to those techniques with no direct interaction between the researcher and the research phenomenon. Analysis of published statistics, textual analysis, postal surveys, and structured interviews are all distant methods (Nandhakumar & Jones, 1997). Studies with direct interaction between the researcher and the research participants are defined as engaged methods and include consultancy, action research, participant observation, passive observation, and unstructured or semi-structured interviews (Nandhakumar & Jones, 1997; Rubin & Rubin, 2005). My research approach involved several forms of data collection: participant observation, passive observation, and interviews (semi-structured, structured, and unstructured). In addition, the contribution of company norms to the process management framework was investigated, and company documents, including meeting reports, guidelines, handbooks, published information, as well as customer documentation (specifications, customer-meeting reports) were studied.

Distance — **Engagement**

- Analysis of published data
- Textual analysis
- Survey
- Interview (structured > semi > unstructured)
- Passive observation (and lab experiments)
- Participant observation
- Action research
- Consultancy

Figure 3-1 Different data-gathering methods (Nandhakumar & Jones, 1997)

I employed these forms of data collection in different ways throughout the research process. The organisation where this research was conducted faced the usual economic challenges and competition (section 4.2.2) and its primary goal was economic success. The fact that research was not the primary goal of the research site moderated my access somewhat. Nevertheless, this provided the opportunity, to collect rich data in a natural setting of embedded developers. I was able to study one specific project, entitled *Theta* (cf. section 4.3), at several stages of its development. During the initial stage, I spent about two months, from December 2004 until January 2005, as a passive observer of the developers and project managers of project *Theta*. During that stage, I was able to develop my first impressions of the nature of embedded development within the company. Moreover, I established first contact, which helped me to collect richer data at a later stage of the research. The major feature of this first phase of observation was to overcome problems of access to the various developers and managers at a promising research site.

Before I conducted the first phase of interviews with developers and managers involved with project *Theta*, I developed my understanding of the wider context of embedded development activities in the German automobile business. I familiarised myself with the literature on process management (cf. section 2.2.2), information system development processes (cf. section 2.3), and improvisation and bricolage (cf. section 3.7.1). Attention was paid to issues relevant to the research setting, such as the process management framework of ISO 9001 (cf. section 2.2.3). This preliminary literature review developed my knowledge of embedded system development settings through a process of critical reflection on the circumstances of the research setting and the available literature. In addition to this review of public material, I also reviewed unpublished organisational documentation (company norms, guidelines, handbooks, and so on) as previously indicated. This comprehensive literature review was important for the progress of my research.

Based on my newly acquired knowledge, I prepared the questions for the first batch of interviews, which were conducted in April 2006. I drew upon the guidance of Radice et al. (1985) to inform my semi-structured interviews with developers and managers. I began with two group interviews involving all developers and managers involved in project *Theta* in order to introduce myself and share rudimentary information about this research project. I assured them that their statements would remain anonymous and they would not be explicitly linked to their company or product. Therefore, names of the organisation, its departments, products, and technology applications have all been anonymised. After these group interviews, I conducted fourteen semi-structured interviews (Rubin & Rubin, 2005) at interview phase 1 with every hardware and software developer, project manager, and middle manager (cf. questionnaire guide in section 9.2.1). These interviews were concerned with the collection of data about the project's progression since my initial observation period. In these semi-structured interviews I used open-ended questions that gave me control over the interview and the opportunity to collect rich data through extensive responses (Rubin & Rubin, 2005). These interviews provided a better comprehension of the social processes of developers and managers in that particular organisational setting of embedded development. In addition, I was able to note some individual differences between research participants. To support the research process, I taped and transcribed most interviews, which was helpful for later analysis. The most important gain from the phase 1 interviews was the enrichment of my understanding of the context and processes at the research site. Furthermore, because the interviews were conducted during the project, interviewees were able to describe the full flavour of social activities without offset.

In addition to the first phase of interviews in April 2006, I was also able to collect relevant data from the customer side of project *Theta*. Between February 2006 and January 2007, I observed the industrial customer, who specified the product and was capable of influencing the software developers' actions through their feedback on the final product (Rautiainen, Vuornos, & Lassenius, 2003). The final products were used by a number of clients. Observing these people from different backgrounds (engineers, test drivers, students, and project leaders) provided me with a better understanding of the meaning with which customers imbue the product. My remoteness from the research subject during this observation stage warranted an adequate distance between the research subject and me. Too close involvement in the research phenomenon may hinder the researcher from keeping appropriate distance. However, this distance is required to ensure impartiality when dealing with the collected data. The opportunity to observe the process from both the broad customer perspective and the point of view of the software developers of the embedded system allowed both the necessary distance, and simultaneous engagement.

Preliminary analysis of the collected data from the first phases of observation and interviews indicated that another observation and interview phase with the developers and managers was necessary to derive sufficient data. The observation period also needed to take place over a longer period. Through an arrangement with the development site managers, I was able to observe the developers and managers from February 2007 until November 2008. I observed the software developers through immersion in the Software Development department where the investigated product was created. I was able to collect very rich data from this immersion. Most observation was passive (Leidner & Jarvenpaa, 1993) apart from a more participatory two month period (Mouakket, Sillince, & Fretwell-Downing, 1994; Pentland, 1992). My active observation involved a comparison of the developers and managers' project results with the engineering processes of a process management framework namely ISO 15504 (SPICE). By engaging in both participative and passive observation, it was possible to understand the nature of the process more fully and to identify local meanings and their effects. Therefore, the collected data delivered a multitude of new insights into micro-level factors, such as individual differences and non-technical aspects of the process, as well as the influences upon the respective stakeholders.

Figure 3-2 depicts the different stages of the research and the course of project *Theta*, which ended in mid-2007. Accordingly, the second phase of developer and manager observation started the advanced stage of this project. In addition, follow-up development activities were examined, such as necessary maintenance of *Theta*, and other projects, new and old. Although this might have been expected to hinder the collection of relevant data, I think that the collected data consisted of meaning that was rich enough to shift this research undertaking positively. Moreover, the company authorised me to conduct research in the manner described. My chosen methods have the potential to produce problems of researcher intervention, which must be handled with care (Goffman, 1959). While observing, I avoided drawing attention to this research through either engagement with the customer or interference with the duties of developers and their managers. Therefore, I was provided with the same access to secure data as company employees and was accepted as a normal member of the respective teams in the environments of both the customers and the developers. Although a relatively large amount of the observation of developers and managers took place after completion of project *Theta*, it is important to understand that all internal stakeholders (developers, managers, marketing people, and quality managers) remained in their positions. Therefore, it was easy to discuss emerging questions with the relevant people during the research process.

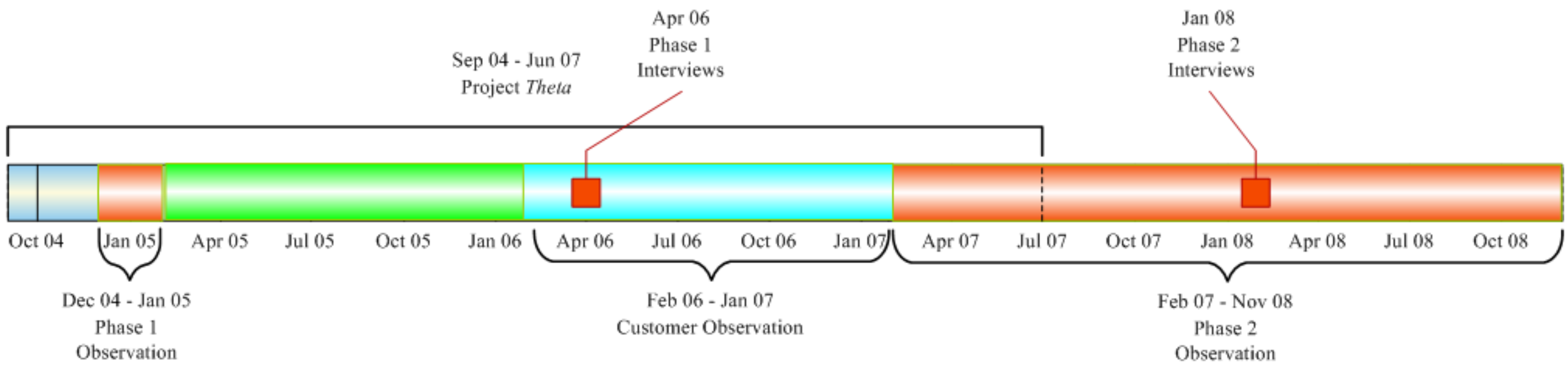

Figure 3-2 Data collection process schedule

Figure 3-2 indicates that I performed a second phase of interviews with the internal stakeholders of project *Theta*. These semi-structured interviews were performed with a good understanding of the nature of the research subjects (cf. questionnaire guide in section 9.2.2). Individual differences contributed to my advanced filtering of the interviewees, enabling enhanced understanding and completion of the chain of evidence from various pre-analysed research results. While the questionnaire was designed to overcome the problem that some phenomena would exceed the period of observation (Nandhakumar & Jones, 2002), the second phase of interviews further assured that this was allowed for (Rubin & Rubin, 2005).

In summary, the primary data sources for this in-depth case study were interviews with the internal stakeholders, conducted over two interview phases. Participants were assured that their statements would remain anonymous and they would not be explicitly linked to their company or product. Most structured and semi-structured interviews were taped and transcribed, and the interviews were between 80 and 100 minutes in duration. In addition, detailed notes were taken of the work activities observed during each visit. Table 3-1 and Table 3-2 summarise the interviews and other data collection methods.

Table 3-1 Summary of interviews

Number of interviews	**Type of interview**	**Position of interviewee**	**Responsibility of interviewee**
Internal stakeholders			
3	Group interview	Group of developers	Study of *Theta*
1	Structured interview	Senior manager	Development, Finance
1	Semi-structured interview	Senior manager	Development, Finance
2	Semi-structured interview	Quality managers	Quality
4	Semi-structured interview	Middle managers	Software
1	Semi-structured interview	Product manager	Marketing
4	Semi-structured interview	Project managers	Study of *Theta*
16	Semi-structured interview	Associates (software and hardware developers)	Study of *Theta*
Customers			
7	Unstructured interview	Users	Engineers, test-drivers, students
3	Unstructured interview	Middle managers	Project leaders with responsibility for the requirements of *Theta*
Other external stakeholders			
1	Structured interview	ISO 9001 Auditor	
∑ 43	**Interviews**		

Table 3-2 Summary of data collection process		
Number/ time	**Type of data collection**	
43	Interviews	
30	Group meeting observation	
22 months	Development department observation	Dec 04 – Jan 05 Feb 07 – Nov 08
12 months	Customer observation	Feb 06 – Jan 07

3.5 Case study site

As mentioned earlier, I found a research site that was willing to accommodate the research process. To encourage open communication from site employees, I assured them that their statements would remain anonymous and they would not be explicitly linked to their company or product. Names of the company, its departments, projects and products, and technology applications were all anonymised. A more detailed description of the case study site may be found in Chapter 4.

The pseudonym of the research site is Engineering Company (EC). The site is a part of the German automobile sector, which is the source of a great deal of economic activity in the country. In addition to the large automakers, there is a supplier industry of international weight. These suppliers' innovations are shown at international automobile events and exhibitions. The company's reputation was established on reliable and high-tech products. EC consisted of several departments, chief among which are the Service Supplier and the Development Department. The Service Supplier associates work is embedded based on the needs of customers. The associates of the Development Department work at the EC offices. EC customers demand high-tech, high quality products; thus, quality is important to the nine-year-old EC. The organisation has experienced significant growth since its inception. Beginning with only four people, the organisation consisted of more than 200 personnel at the time this research was undertaken. This tremendous level of growth has changed the entire organisation, inside and out. For example, internal communication paths have lengthened, decisions require more time, and more people are involved in the decision making process. Communication with customers has also changed. As their market share and experience has grown, more and more customers have noticed EC. These customers expect the quality of EC services and products to be of a consistently high standard. In addition, the German law demands that distributed products in the automobile sector comply to quality standards, such as ISO 9001. However, there were also important internal reasons for EC to acquire ISO 9001 certification. An established ISO 9001 framework creates a more well-structured organisation. Therefore, the EC has been established as an ISO 9001-certified organisation for six years. This certificate shows that EC has established a working process management framework throughout the organisation. The certificate itself is provided by independent organisations, which are able to measure and evaluate organisational process management frameworks. Another characteristic of this organisation is its strong focus on continuous improvement of overall performance. They have introduced the Japanese philosophy of 'Kaizen' as their improvement philosophy. This focuses on steady development and improvement at all levels, from individual to organisational.

The EC has a four level management hierarchy in its Development Department: senior management, middle management, project leaders, and associates. This research is largely focused on the work practices of middle managers, project managers, and associates of the Development Department (cf. Table 3-3).

Table 3-3 Key individuals in this research

Name (pseudonym)	Role	Responsibility
Michael	Senior manager	Development, finance
Gabriel	Middle manager	System architecture, software development
Martin	Project manager	Studied development project
Raphael	Associate	Software

Michael, senior manager and co-founder of the EC, directs hardware and software development. Gabriel (middle manager, software) is responsible for software development within EC. Martin was the project manager for the development project studied in this research. Half of his time was spent with the customer as a service supplier and the other half with the associates. This allowed EC to have first-hand contact with its customer. Although this situation is new, it was a key factor in the acquisition and contribution of information for this project. Raphael's contribution to the project was outstanding. The EC associates portrayed him as a 'fire fighter', due to his ability to help and solve problems. Other important individuals will be described in Chapter 4.

The development of a new embedded system, *Theta* (a pseudonym), at EC was the focus of this investigation. In particular, I was interested in the processes and context of this project. This system was an electronic device (based on Linux) that used software to connect with the various bus systems of a vehicle or test rack. Its purpose was to record all transmissions on these bus systems and store them on a hard disk drive. It was possible to connect to this device on a personal computer using its client software for a later readout and analysis of the data.

3.6 Data analysis

Since the border between data collection and data analysis is not as clear in qualitative research as it is in quantitative, it is maybe more precise to speak of "modes of analysis" instead of "data analysis" (Myers, 1997). My approach to the analysis of the data gathered was based on Miles and Huberman (1994). Several distinctive types of coding were used to support data analysis. Those were applied iteratively as my understanding was developed by new knowledge. New information was collected almost continuously over the duration of the field research (cf. Figure 3-2), through interviews, observation, organisational documents, and so on. The data were pre-selected to avoid data overload from the acquisition of material unrelated to the research questions. I thereby minimised the problem of data retrieval in later analysis throughout the research process, although the collected data load was still immense. Almost concurrently with the data collection process, I tried to sort the obtained data in order to obtain some understanding of the data load. Accordingly, the first action was to deposit data with some meaning in order to expose the various activities, events, and incidents. Miles and Huberman (1994) describe this initial step of data analysis as descriptive coding. This early, ongoing analysis enhanced my critical reflection upon the circumstances of the research setting. In addition, it influenced data collection by causing me, on occasion, to re-adjust my perspective and further focus during the research process. Moreover, early, ongoing coding revealed previously unnoticed sources of bias, and helped me to detect partial and vague data that could be clarified during the field research. This ongoing mode of gathering data and analysis through descriptive coding demonstrates the hermeneutic circle in practice. The significance of field research lies in its live revelation of actions and practices. Methods that intensify examination of such actions are particularly beneficial, but must be flexible (Miles & Huberman, 1994). Miles and Huberman (1994) state:

> The ultimate power of field research lies in the researcher's emerging map of what is happening and why. So any method that will force more differentiation and integration of that map, while remaining flexible, is a good idea. Coding, working through iterative cycles of induction and deduction to power the analysis, can accomplish these goals. (p. 65)

Descriptive coding is then followed by interpretive coding, which derives meaning from the data, puts it into context, and clarifies its significance. To develop my understanding of the complex research setting, I wrote short descriptions in table form, dissecting the obtained data (cf. analytical tables in section 9.3). With new data, previous interpretations were adjusted and developed. During the initial stage of this research, a number of surprises emerged from the data. For example, social structures and norms appeared to differ over time, and this both permitted and hampered the research participants' activities. In addition, some field data needed to be interpreted differently during the ongoing data collection, because some codes failed to apply and others decayed in relevance. Lincoln and Guba (1985) have categorised the elements in this process of continuous refinement in data coding (Miles & Huberman, 1994). These are:

a) filling in: adding codes, building upon a consistent scheme as new situations develop and new ways of looking at the research data become apparent;
b) extension: returning to previously coded data to cross-examine them in a new way in light of freshly obtained data;
c) bridging: building new or previously unnoticed relationships within units of a given category;
d) surfacing: determining new categories.

I applied these elements of continuous refinement during my research process and found that it was natural to adjust and refine my understanding through new knowledge frequently. This stage of interpretive coding was a very important aspect of the data analysis. The analytical tables in section 9.3

(appendix) summarise the emergent themes (structured processes, improvisation and bricolage, reflexivity and human agency) from the case analysis.

Usually, it is not enough to name and describe what we perceive. This research searched for patterns in the data in order to gain a better understanding of the research phenomena. Therefore, the final step in understanding the various meanings is described as pattern coding (Miles & Huberman, 1994). Patterns, themes, leitmotivs, or causal links are sought later in the course of data collection, as they become more visible. Pattern coding can include noting plausibility, clustering (what goes with what), counting meanings, noticing contrasts or comparisons, noting relations between meanings, building logical chains of evidence, and finding observable conceptual coherence. For example, I was able to note dynamics between improvisation and bricolage activities and structured processes (see Figure 6-8). In addition, pattern coding led interpreted data to become more inferential and explanatory as they were detected in situations of the research setting. The detection of "repeatable regularities" (Kaplan, 1964) also allows previous narratives to be grouped into a smaller number of analytic packages (Miles & Huberman, 1994). Following interpretive coding, pattern coding then improved my continuous fieldwork by provoking comprehensive reflection on the research setting.

My data analysis included a variety of tasks, such as selecting, condensing, and transforming obtained data. Section 3.8 describes how these data were presented in an organised way. Conclusions were drawn using an approach based on several distinctive types of coding as described in this section (Miles & Huberman, 1994). Finally, checking for representativeness, weighting the evidence, and obtaining feedback from informants, such as research colleagues, verified conclusions. This feedback in particular, gave me confidence in the validity of the meanings that I had inferred from the case-study data using this analytical approach.

3.7 Theoretical lenses

When I described my epistemological position in section 3.2, I mentioned, that interpretive researchers tend to apply theoretical frameworks as “sensitising devices” or “theoretical lenses” through which to view the world from a certain perspective (Orlikowski & Robey, 1991; Poole & DeSanctis, 2003). Potential theoretical lenses may include ISD methodologies, such as specification-based, evolutionary, and agile development methodologies. However, those structured approaches are less helpful in getting a better understanding of procedures that are unstructured, or situated in emergent contexts. Therefore, I applied improvisation and bricolage (section 3.7.1) as the first sensitising device. However, that sensitising device does not possess descriptions of the human actions and the motives of individuals, regarding why those processes occur. The *concepts of reflexivity and human agency* (section 3.7.2) help to explain human actions and the motives of individuals and because of that they make those vital aspects of social life better comprehensible. Section 3.7.3 describes how the sensitising devices are applied and how they helped to analyse the case study.

3.7.1 Improvisation and bricolage

Research on the phenomenon of improvisation and bricolage has received increased recognition within a variety of contexts, such as new product developments and entrepreneurship (Akgün & Lynn, 2002; Boccardelli & Magnusson, 2006; S. Brown & Eisenhardt, 1995; Hmieleski & Corbett, 2006; Kamoche & Cunha, 2001; Moorman & Miner, 1998b), organisational change (Baker, Miner, & Eesley, 2006; Crouch, Streeck, Whitley, & Campbell, 2007; Cunha, 2003, 2005; Cunha & Cunha, 2003; Cunha, Cunha, & Kamoche, 1999; Kamoche, Cunha, & Cunha, 2003; Miner, Bassoff, & Moorman, 2001; Mirvis, 1998; Moorman & Miner, 1998a; Orlikowski, 1995; Weick, 1993a, 1998), management (Barrett, 1998; D'Agostino, 2006; Lewis, Welsh, Dehler, & Green, 2002; Leybourne & Sadler-Smith, 2006; Montuori, 2003; Stahl, 2005; Vera & Crossan, 2005), and ISD and application (Bansler & Havn, 2003). However, for a long time, it has always been associated with a dysfunction, an "unintended outcome or as an organisation design failure" (Lewin, 1998; MacKenzie, 1986; March & Simon, 1958).

The goal of this section is to reveal the concept of improvisation and bricolage. The gathered understanding of improvisation and bricolage forms the first sensitising device of this thesis. This sensitising device helps to analyse the data load more comprehensively by interpreting procedures that are difficult to describe, control and manage, or that are situated in emergent contexts.

The term's origin of improvisation and bricolage is detailed in section 3.7.1.1. Section 3.7.1.2 comprehends improvisation on the basis of six paradox pairs with related theoretical constructs found in the literature. Section 3.7.1.3 elaborates the differences between improvisation and bricolage. Furthermore, examples of improvisation and bricolage activities are described in section 3.7.1.4.

3.7.1.1 Latin word origin

To get a better understanding about the origin of the terms improvisation and bricolage, each will be discussed in this section. The term improvisation is derived from the word 'proviso', meaning to provide something in advance. The prefix 'im' gives it the opposite sense: without previous planning or conditioning (Weick, 1998). 'Not seen ahead of time' might be another adequate description of 'improvisus' (Ciborra, 2002). In addition, in Latin, improvisation is paraphrased as 'extemporalis actio', hence action outside of time, or outside the normal flow of time (Ciborra, 2002). The word 'improvise' concerns the unforeseen and unprovided-for, which describes Tyler and Tyler (1991) as "the negation of foresight, of planned-for, of doing provided for by knowing, and of the control of the past over the present and future" (p. x). The act of improvisation is therefore unguided by 'how-to' and other descriptions for preparations that happened in the past, and this perspective is the opening for an art that is neither a craft, nor a technology capable of being mastered (Tyler & Tyler, 1991).

Bricolage (from the Latin 'bricola', which means catapult) is adopted from the French and paraphrases the meaning of 'to fiddle, to tinker' or 'making do' with the items or resources at hand (Ciborra, 2002; Lévi-Strauss, 1966). Bricolage is about leveraging the circumstances as defined by the situation (Ciborra, 2002). Items or resources are going to be used by the bricoleur (person who engages in bricolage) regardless of their initial purpose; as an interesting side note, in France do-it-yourself home improvement stores are called 'bricolage markets'.

3.7.1.2 Comprehension of improvisation

Improvisation is understood as the convergence of design and execution (Baker, et al., 2006; Moorman & Miner, 1998b), and those activities occur outside organised routines or formal plans (Miner, et al., 2001). Improvisation involves to some extent creation (composition) but of a specific sort (Baker, et al., 2006), and it occurs practically at the same time. For example, improvisation is described as "real-time composition", "reading and reacting in parallel", "thinking in the midst of action" (Moorman & Miner, 1998b), and "the negation of foresight, of planned-for, of doing provided for by knowing, of the control of the past over the present and future" (Weick, 1998).

These different nuances indicate how complex the comprehensions of improvisations are. Following Mirvis (1998), Ciborra (1999), and Zheng's et al. (2007) work, I resume six different paradox pairs of improvisation with related theoretical constructs found in the literature (Table 3-4).

Table 3-4 Paradoxes of improvisation

Improvisation-paradox	Related theoretical constructs	Key references
Pragmatic creativity	Environmental turbulence	(Ciborra, 1996; Moorman & Miner, 1998b)
	Task uncertainty	(Dahlbom & Mathiassen, 1993)
	Unplanned-for occurrences	(Miner, et al., 2001)
	Task complexity	(Hutchins, 1995; Weick & Roberts, 1993)
	Drop your tools	(Weick, 1993b)
	Visions	(Hatch, 1999; Hutchins, 1991; Mintzberg & McHugh, 1985; Weick, 1993a)
	Situated change	(Orlikowski, 1995; Weick, 1998)
Oriented drifting	Convergence of planning and execution	(Moorman & Miner, 1998b)
	Mixing the pre-composed and the spontaneous	(Weick, 1998)
	Magnetic fields	(Weick, 1993b)
	Minimal structure	(Cunha, et al., 1999)
	Plan to improvise	(Miner, et al., 2001)
	Artful planning	(Baskerville, 2006)
	Chosen voluntariness	(Dehlin, 2008)
	Utilisation of the known	(Dybå, 2000)
Retrospective order	Retrospective sense-making	(Weick, 1993a)
	Ex-post interpretation	(Lanzara, 1999)
	Transient constructs & Persistent structure	(Lanzara, 1999)
	Emergent order	(Miner, et al., 2001)
Managed serendipity	Organised anarchy	(Cohen, March, & Olsen, 1972)
	Collateral structure	(Cunha, et al., 1999)
	Experimental culture	(Cunha, et al., 1999)
	The aesthetic of imperfection	(Weick, 1999)
	A sense of urgency	(Crossan, 1998; Hutchins, 1991; Mirvis, 1998)
	Previously established understanding	(Miner, et al., 2001; Ryle, 1979; Vera & Crossan, 2005; Weick, 1998)
Collective individuality	Facilitative leadership	(Crossan, 1998)
(Mirvis, 1998)	Trust and kinship	(Crossan, 1998; Weick, 1993b)
	Emotional communication	(Hatch, 1999)
	Hanging out	(Barrett, 1998)
	Fluid communication	(Miner, et al., 2001; Orlikowski, 1995)
Anxious confidence	Individual skills and creativity	(Hutchins, 1991; Moorman & Miner, 1998b; Orlikowski, 1995)
(Mirvis, 1998)	Formative context	(Ciborra & Lanzara, 1994)
	Organisational memory	(Moorman & Miner, 1998b)
	Moods	(Ciborra, 2002)

Those paradoxes identify the whole ambivalence of improvisation, because it is difficult to characterise it with certain attributes and to describe the full flavour of improvisational activities. Moreover, these paradox pairs seem to emerge genuine through the interplay of contrary weights (Mirvis, 1998). Accordingly, the emergence of new solutions through improvisation involves the limited ability to predict what will result (Mirvis, 1998).

Pragmatic creativity: Improvisations may be very innovative and differ more from existing constructs, and other improvisations may only involve less creativeness (Moorman & Miner, 1998b). Weick's (1998) research describes various instances of improvisation and declares activities that alter, revise, create and discover to be purer instances of improvisation. He claims that these instances inhabit the full improvisational spectrum and involve smooth transitions. In contrast, activities that shift, switch or add are weaker instances of improvisation and are the result of isolated improvisation. In other words, these are improvisations that occur situated and involve pragmatism. Orlikowski (1995) describes also the contradictions of improvisation, which may involve creative and radical improvisations, incremental improvisation, and pragmatic everyday improvisations.

Oriented drifting: Improvisation deals with the unplanned, and it happens without prior conditioning (Weick, 1998). Moorman and Miner (1998b) described improvisation as a variation from existing activities and understanding, and Weick (1993a) relates improvisation to continually rebuilt processes and design. Dehlin's (2008) research findings relates negative improvisation with vagueness, ambiguity, unpredictability, and the ability to cluster events of improvisation sequentially. In contrast, positive improvisation is chosen voluntariness. In addition, Dybå (2000) differentiates between explorative and exploitative improvisational actions and practices. Explorative improvisation is a discovery of the unknown; exploitative improvisation is an utilisation of the known.

Retrospective order: Weick (1993a) and Lanzara (1999) state that meaning often become apparent from retrospective interpretation of activities, whereas those activities may seem chaotic when they occur. This retrospective order of improvisational activities is another contradiction, in describing the full flavour of improvisation. Moreover, Miner et al. (2001) describe that an evolving improvisation and reflections on it give rise to an "emergent order", which might be utilised for further activities in that setting.

Managed serendipity: Previously established understanding shapes improvisational activities, because it determines some contours of development and excludes others (Ryle, 1979; Weick, 1998). Therefore, gathered comprehensions and skills influence improvisation in important ways (Miner, et al., 2001; Vera & Crossan, 2005). Hence, improvisation does not occur randomly or accidentally (Miner, et al., 2001; Weick, 1998). Instead, improvisation is founded on a sophisticated fundament of experience and advanced understanding that can be managed and therefore influenced in a positive way. Consequently, to improve improvisation is to improve that fundament through a higher level of experience and higher maturity of understanding (Weick, 1998).

Collective individuality (Mirvis, 1998) concerns improvisation when it is performed in a team. Many improvisational activities are the result of collective efforts to strive for a reasonable output, but at the same time, some individuals may have contributed a larger portion of work than others. Those improvising stars of an undertaking may be differently rewarded than the rest of the team through the fragmentised organisational reward system with group and individual rewards. For example, good sport teams usually require playmakers (individuals) and 'ordinary' teammates (collective) to respond to complex situations in the world of sport.

Anxious confidence (Mirvis, 1998): Final, anxious confidence relates to the optimal balance point between excitement and angst beyond the area of poor solutions (Csikszentmihalyi, 1990; Mirvis, 1998). Pure excitement and confidence are encouraging aspects for individuals and teams to strive toward better solutions as previously found. In contrast, pure angst and anxiety lead to devastating results. Performing improvisation entails both to a certain degree, because a person may be confident in the emergent solutions of the improviser but may be worried about the unknown of the newly emergent solution.

In conclusion, various samples of improvisation involve different aspects of improvisation, but these do not imply that some types of improvisation are better than other types.

3.7.1.3 Comprehension of bricolage

Baker (2007) stated, the comprehension tying bricolage to improvisation is compelling. Improvisational activities may involve bricolage, but bricolage also occurs when planning precedes execution (Baker, et al., 2006; Cunha, 2005; Miner, et al., 2001; Moorman & Miner, 1998b). Therefore, bricolage occurs outside of improvisational activities and represents an advantageous theoretical and practical concept for research on business processes (Baker, et al., 2006). Baker and Nelson (2005) offered a definition of bricolage as "making do by applying combinations of the resources at hand to new problems and opportunities" (p. 333). Furthermore, the literature review indicates that different perspectives sometimes highlight more improvisation and sometimes more bricolage terminology. In comparison, both terminologies have their right to exist, because there are two main differences found in the literature. First, bricolage may involve gathering and creating something with the items or resources at hand (Lanzara, 1999; Lévi-Strauss, 1966; Louridas, 1999; Weick, 2001). This stresses the physical nature of bricolage and makes it different from improvisation. For example, it is possible to improvise in the domain of composing music, but the idea of bricolage is not applicable to that domain because it does not involve any physical aspect. In contrast, using dishes to improvise drums might be called bricolage, because it does involve some sort of instruments, which of course are physical objects. Second, bricolage may occur with pre-planning, whereas improvisation occurs (often spontaneously) in the absence of planning (Lanzara, 1999; Louridas, 1999; Weick, 2001). For example, building a tree house for your child making use of whatever materials are at hand implies bricolage and may be improvisation. However, bricolage does not imply improvisation. Either way, the ability to create new instances is fundamentally proof of how capable the bricoleur is (Baker, et al., 2006). These differences, and the point that improvisation may involve bricolage, educe the understanding that improvisation and bricolage are two overlapping phenomena (Figure 3-3).

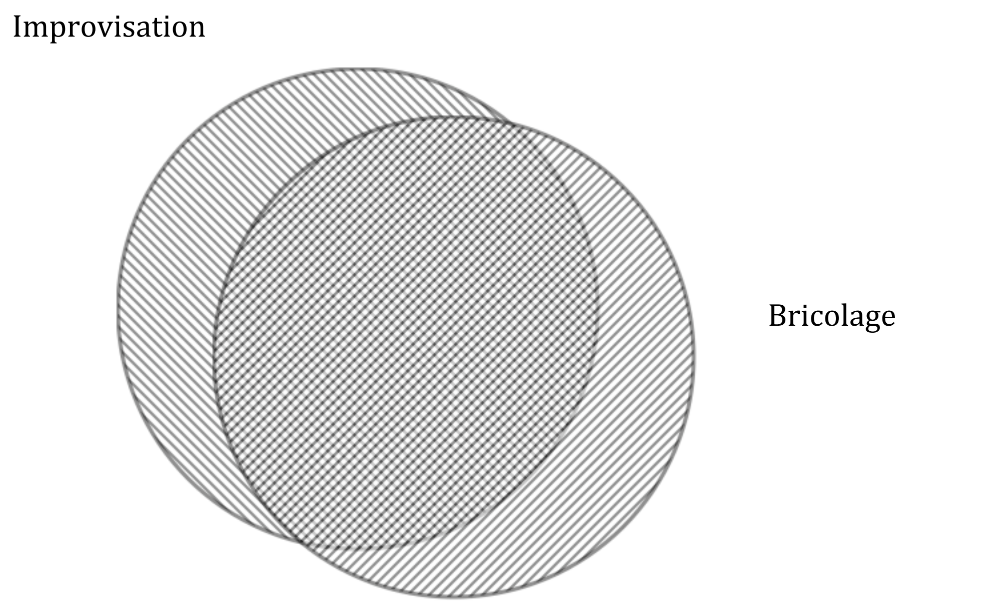

Figure 3-3 Understanding of improvisation and bricolage (Molnar & Nandhakumar, 2008b)

In order to reveal the characteristics of bricolage, some aspects may be similar to improvisation, but this is due to the indistinguishability of the two overlapping phenomena. Similar to an improviser, the bricoleur (someone who performs bricolage) is able to perform a variety of tasks and unlike the engineer, he or she does not fragment tasks to check the availability of tools and resources to meet the requirements for that task (Coppola & Elliot, 2005). Furthermore, the bricoleur is capable of re-interpreting and re-combining available resources (Boccardelli & Magnusson, 2006). However, the engineer may restrict himself to a subset of resources and tools that are required to complete the tasks (Hmieleski & Corbett, 2006; Lévi-Strauss, 1966). Bricoleurs remain innovative in troubled times, because they are used to operating in seemingly chaotic circumstances and get along somehow with whatever resources are available and adapt to possible routines (Ciborra, 1996). Both the improviser and bricoleur respond to this chaotic circumstance on an ad hoc basis (Hatton, 1989), and form a course of action where the actors become involved in paths that they try to shape in real time (Garud & Karnøe, 2003). In turn, these courses of action begin shaping the actors over time (Garud & Karnøe, 2003).

The literature review reveals that bricolage may be distinguished to different degrees. For example, Butor and Guynn (1994) describes two levels of bricolage, that is, the use of tools that are provided and the invention of material. The use of tools is the offensive aspect, but more interesting is the invention of material, because this involves the gathering and recovery of items – in other words: recycling (Butor & Guynn, 1994). Ciborra (1996) has determined three dimensions. First, 'tinkering' or 'fiddling' with available objects; second, 'trial and error' actions to strive for a reasonable solution (Ben-Ari, 1999); third and final, forming an entity. This entity, as Ciborra (1996) puts forth, needs to be a form of creative expression and the result of putting resources together in ways that are not usually done (or ways that are not 'supposed' to be done). This kind of entity as one aspect of bricolage was also described as a nonlinear and holistic attack on a problem that ends in some reasonable creation (Innes & Booher, 1999).

This brief section about the comprehension of bricolage summarise the relation to improvisation and reveals characterisations and different degrees of bricolage. The next section 3.7.1.4 describes various examples of improvisation and bricolage.

3.7.1.4 Examples of improvisation and bricolage activities

Many studies have examined improvisational activities in performing arts such as the composition of music and theatre (Barrett, 1998; Berliner, 1994; Crossan, 1998; Kamoche & Cunha, 2001; Lewin, 1998; Orlikowski & Hofmann, 1997; Peplowski, 1998; Weick, 1998). Those metaphors may have some limits, but they provide greater insights into the phenomenon of improvisation (Hatch & Weick, 1998; Kamoche, et al., 2003). Mangham and Pye (Mangham & Pye, 1991) presented a reasonable description that compares managers to jazz musicians (Weick, 1998):

> Like jazz musicians, managers simultaneously discover targets and aim at them, create rules and follow rules, and engage in directed activity often by being clearer about which directions are not right than about specified final results. Their activity is controlled but not predetermined. (p. 79)

Certainly a well-known bricoleur of popular culture is MacGyver, the main character of the television show of the same name, which was broadcasted in the late 1980s and early 1990s (Bogost, 2006). MacGyver was a paragon of a bricoleur when he applied his scientific knowledge in practical ways through the inventive use of common items at hand – along with his ever-present Swiss Army knife[1]. Another interesting example of bricolage is when it is used in the context of research (cf. Kincheloe, 2001, 2005). Bricolage describes an interdisciplinary research approach, when a research process crosses

---

1 MacGyver. (n.d.). *Wikipedia* Retrieved September 5th, 2007, from http://en.wikipedia.org/wiki/MacGyver

boundaries and the researcher employs the analytical frames of more than one discipline. Certainly, the multidisciplinary approach offers limitless possibilities for conducting research; however, those bricoleur researchers need to clarify what concepts are used in order to contribute to the body of research. Particularly in social research, this approach might be useful when researchers explore blended domains. Kincheloe (2005) states:

> The bricolage exists out of respect for the complexity of the lived world. Indeed, it is grounded on an epistemology of complexity. (p. 324)

Some results of improvisational and bricolage activities are very innovative, such as the rescue of Apollo XIII as part of the National Aeronautics and Space Administration (NASA) (Moorman & Miner, 1998b); other improvisations and bricolage outputs are the everyday improvisations and slippages or incremental improvisational and bricolage activities by organisations (Orlikowski & Hofmann, 1997). As occurred in the case of the Apollo XIII, improvisation and bricolage activities are one possibility to respond to environmental turbulences. When an organisation faces many change, it has the possibility to ignore the turbulences and continue with already planned activities; it can try to accelerate iteration cycles in order to respond to changes adequately; or it can try to cope through improvisation and bricolage (Moorman & Miner, 1998b).

Indeed, the ability to improvise and perform bricolage was also associated with organisational reliability, because it is very challenging for organisations when environmental turbulences are unexpected and thus demands dynamic reactions (Amabile, 1988; Cunha, 2005; Eisenhardt & Tabrizi, 1995; Weick, 2003). However, it was also discovered that improvisation and bricolage can cause harm, and therefore it cannot be generalised that improvisation and bricolage is always good (Miner, et al., 2001). It is also important to emphasise that organisations may not develop an organisation-wide proficiency to perform improvisational and bricolage activities. Organisations as such vary with their departments, groups and individuals, and therefore they may only develop improvisational and bricolage proficiency in very specific subdivisions (Miner, et al., 2001).

3.7.2 Structuration Theory

In the realm of information systems research, Structuration Theory has been widely accepted and used. For example, between 1986 and 2002, 225 information system articles have been identified that referenced Structuration Theory (M. Jones & Karsten, 2003). Indeed, numerous researchers have underlined the value of Structuration as a theoretical lens through which a variety of information system issues can be studied and have sought to increase its relevance to empirical research (Pozzebon & Pinsonneault, 2005). In addition, a number of themes of Structuration Theory fit into this study's interest in directed approaches (process management cf. section 2.2, ISD processes cf. section 2.3) as enabling and constraining social structures, and the dynamics of social life such as improvisation and bricolage activities (cf. section 3.7.1). I refer in particular to the Structuration Theory concepts, such as reflexivity and human agency, which elaborate on Giddens (1984) work. I use those concepts as a sensitising device, because they help to explain human actions and the motives of individuals; therefore, they make those important aspects of social life better understandable. Moreover, using Structuration Theory as a sensitising device helps to abstract and generalise the data, in order to derive sophisticated answers to research questions (cf. section 3.2).

It might be useful at this point to briefly summarise some of the key ideas contained in the Structuration Theory. A detailed review of Structuration Theory can be found in M. Jones (1999). Giddens (1984) Structuration Theory helps to reconcile structural dualities in order to avoid describing how actions or structures are conceived. Structures refer to rules and resources that are recursively involved in the reproduction of social systems. As a result of the need for reproduction, Giddens (1984) states “structures exist only as memory traces, the organic basis of human knowledgeability, and as instantiated in action” (p. 377). Thus the demand for reproduction of social systems, structures may be perceived as both constraining and enabling for social actors (M. Jones, 1999).

All human beings are knowledgeable social actors, therefore in that they know a great deal about the conditions and consequences of what they do in their daily life activities (Giddens, 1984). This knowledge is partly individual and partly situational. Knowledgeability may operate on different levels of consciousness. Practical consciousness contains knowledge about actions that social actors cannot express discursively. Actors are also able to describe what they do and why they do it. Nevertheless, actors do not usually concern themselves about what level of consciousness their knowledge of chosen actions is embedded in. However, if activities become confused (e.g. refuse to conform to given procedures) rationalisation of conduct may be revealed if the actors are asked why they acted as they did. The knowledgeability of social actors is also connected with unconscious, unacknowledged, and unintended consequences of action. This complex diversification of knowledgeability and consequences of action means that any analysis of the actions and practices within organisations should be based on observations made over time, and should include study of the context or of the contextual features of social interaction. Context refers to the time-space boundaries around pieces of interaction (when and where), anything involving how interaction occurs, and the knowledgeability or level of consciousness influencing or controlling the interaction process. Another important theme of Structuration Theory are routines that serve as a key feature of the duality of structure due to the continuity of social life. In fact, the predominant types of daily social activities are those with no direct motivation that are linked to routines (M. Jones, 1999). The engagement in predictable routines and encounters cultivates a sense of ontological security. Ontological security is the confidence that the social world conforms to our beliefs about it.

3.7.2.1 Reflexivity and human agency

Giddens (1984) emphasis that social structures do not exist independent of human action, nor are they material objects. He describes them as 'traces in the mind' and argues that they exist only through the action of humans. This supports a view of human beings as beings in a continuous mode of reflexive monitoring of their social life (M. Jones, 1999). Thus, reflexivity is an important theme of Structuration Theory (Giddens, 1984). As indicated, knowledgeability of social agents can influence or control ongoing interactions. The reflexive form of knowledgeability is especially involved in the recursive control of social practices. The flow of practices presumes reflexivity, but reflexivity, in turn, is conceivable because all social agents repeat the flow of practices in an identical way. Therefore, social agents are purposive, have reasons for doing what they are doing, and are able to explain those reasons. Practices are a continuous flow of conduct, just as cognition is a continuous flow of understanding. Purposive actions are embedded in that ongoing flow. Accordingly, social agents involve reflexivity in the continuous monitoring of actions and expect that other agents behave similarly. Reflexive monitoring is dependent on the competence of social agents, in terms of their capacity to rationalise ongoing social life.

Reflexive monitoring of activity is an ongoing process and involves not only the monitoring of oneself but also of others (Giddens, 1984). Social actors maintain a continuing understanding of the reasons for their activities, hence rationalising their actions. While this is not necessarily related to the ability to explain the reasons for a specific behaviour, the capacity to do so is an important criterion of competence in day-to-day conduct. The stratification model (see Figure 3-4) distinguishes the reflexive monitoring and rationalisation of action from its motivation. Reasons are concerned with the grounds for an action; motives are concerned with the wants that stimulate it.

The motivating stimulus is a potential of action, in contrast to the ongoing activities of reflexive monitoring and rationalisation by social actors. Only in relatively uncommon settings, when routine behaviour is not applicable, do purposive actors seek to behave differently. The predominant types of daily social activities are linked to modes of reflexive monitoring and rationalisation; hence, they are not directly motivated. Those types of social activity (reflexive monitoring, rationalisation, motivation) may be related to different levels of consciousness. Discursive consciousness describes the social agents' ability to express social conditions, including the conditions of their own actions. Practical consciousness is the social agents' ability to perform despite an inability to express how or why they are doing so. We therefore, Giddens (1984) argues, know more than we can say (M. Jones, 1999). Interestingly, Giddens (1984) did not wish a rigid distinction to be drawn between discursive and practical consciousness. A clear distinction can, however, be drawn between the unconscious and conscious realms. Unconscious motivation and cognition are important contributors to human conduct. Those are either completely repressed from consciousness or appear in consciousness only in distorted forms. In addition, actors cannot control exactly the way in which structure is produced and reproduced. Therefore, Giddens (1984) put emphasis on the unacknowledged conditions and unintended consequences of action. The stratification model stresses the three 'layers' of cognition and motivation (discursive consciousness, practical consciousness, unconsciousness) in order to describe the processes involved in the actions of social actors (reflexive monitoring, rationalisation, motivation of action).

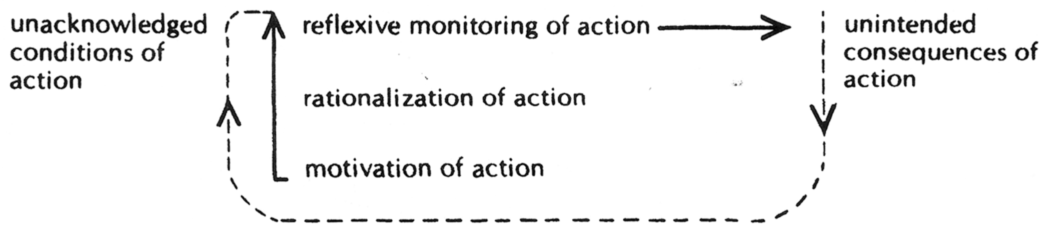

Figure 3-4 Stratification model of the social actor (Giddens, 1984)

Agency does not describe the intentions behind actions but the individual's ability to perform them. Thus, agency is about events that are driven by social actors, in which the actors could choose to act otherwise at any given moment. Being the driving force of events may lead to either intended or unintended consequences. In general, consequences are events that would not have occurred if the social actor had acted otherwise. Regardless of the intentions of the social actor, some events are not within the power of the agent to have caused.

3.7.3 Summary

As earlier indicated, I applied theoretical work on improvisation, bricolage, and concepts from Structuration Theory (reflexivity and human agency) as “sensitising devices” or “theoretical lenses” to analyse the data and discuss the gathered findings.

First, I pre-analysed the data load (field notes, interview notes and transcripts, data from company documents) through depositing data with some meaning to derive a better understanding about what happened at the case study site. Miles and Huberman (1994) describe this preliminary step of data analysis as descriptive coding. This coding occurred through iterative cycles of attaching meaning to data and collecting more data in order to re-adjust my perspective and further focus during the analysis process.

Second, the pre-analysis allowed me to further focus my research, and it became visible that the sophisticated interpretation of the data load requires three analytical topics:

a) structured processes (such as the process management framework of ISO 9001);
b) practice of improvisation and bricolage activities; and
c) reflexivity and human agency.

The process management environment of the case study site calls for an initial investigation that is concerned how project details and processes were conducted in its context. Therefore, the first analytical topic involved the study of structured processes and how research participants followed prescribed processes. However, the pre-analysis of the data revealed also processes that emerged over time and became important for the research participants. Because the analytical topic of structured processes provided less help to analyse appropriately that kind of data, a second analytical topic was used. The second topic of improvisation and bricolage (section 3.7.1) helped to analyse data related with processes that were complex and difficult to describe, control and manage, and were situated in emergent contexts. Although the application of the second analytical topic overcomes shortcomings of the first one, the interpretation of the data load required a third analytical topic to describe inherent human actions and the motives of individuals better. Therefore, I applied the concepts of reflexivity and human agency (section 3.7.2) to the data load, to make human actions and motives understandable.

In familiarising and working with those analytical topics, I was able to categorise, and position, data into context, and clarify its significance. Miles and Huberman (1994) describe this task as interpretive coding and the analytical tables in section 9.3 (Table 9-1 Analytical table: structured processes; Table 9-2 Analytical table: improvisation and bricolage; Table 9-3 Analytical table: reflexivity and human agency) summarise the intermediate result of my analysis process of this case study.

This stage of the analysis process was followed by the third step of data analysis: Detecting patterns within the data load. This task is identified as pattern coding (Miles & Huberman, 1994) and involves a comprehensive reflection on the research setting by seeking themes, leitmotivs, or causal links. Determined patterns are subject to express comprehensive narratives that are based on the case study data. Chapter 5 details those determined patterns, by following the three analytical topics of this research.

Finally, theoretical work on improvisation, bricolage, and Structuration Theory helped not only to analyse the data load thoroughly, but it also biased the thesis discussion in a positive way. The case study participants' improvisational and bricolage activities are discussed in section 6.2. That part of the discussion reveals how developers and managers practice improvisation and bricolage in the context of the case study and hereby aims to answer research question 1. Moreover, reflexivity and human agency, concepts of the Structuration Theory, are discussed in section 6.3. Indeed, reflexivity is a vital aspect in seeking procedures that help to balance the tension between process management and improvisation and bricolage activities. Moreover, the proposed conceptual model that seeks to answer the research question 2 in section 6.4 is based on the previously gathered findings. Therefore, the theoretical work of the "sensitising devices" helps to answer the research questions that are formulated in section 2.5.2 and to derive a conceptualisation.

In summary, this research benefited in three ways by using theoretical work on improvisation, bricolage, and Structuration Theory as vital elements of my analytical topics: First, the theories improved my understanding of the data gathered in analysing the data load comprehensively; second, the theoretical work enabled me to answer the research questions; and third, it facilitated me to formulate a conceptual model that is based on the research findings.

3.8 Presentation of data

Data are presented as a case study prior to the analysis chapter of this thesis. As described in Section 3.4, the data were collected through various methods, including interviews, observation, and analysis of published data and organisational documents. Instead of presenting the reader with raw data, I have formulated a narrative that describes each significant event. I have emphasised the meanings that participants gave to various incidents. This enhances the readers understanding the causalities within the described data. The narrative contains enough description of context for readers to understand the nature of the incidents. In order to offer a rich description, the case study includes some technical jargon from the field, but these are self-explanatory in the context of the description. In summary, the narrative presents an account of my engagement over time with the case study site.

3.9 Summary of research approach

Table 3-5 summarises this chapter, providing a brief overview of my research approach.

Table 3-5 Summary of research approach	
Topic	Information systems development in a process management environment
Focal level	Embedded software design in practice
Epistemology	Interpretive
Data collection methods	Participant observation, passive observation, interviews, analysis of published data
Data presentation	In-depth case study
Data analysis	Descriptive, interpretive, and pattern coding
Theoretical lenses	Improvisation and bricolage; reflexivity and human agency

At this point, I want to emphasise that my doctoral advisor, Professor Nandhakumar, supervised this research and intermediate results were considered for publication by reputable international conferences and their corresponding journals. These results were presented at the International Software Process Improvement and Capability dEtermination (SPICE) Conference, the Hawaii International Conference on System Sciences (HICSS), the European Conference on Information Systems (ECIS), and the International Conference on Information Systems (ICIS). In passing through the review processes of conferences and journals, I have received valuable feedback from commentators that has supported my work on this thesis.

„Verbastelt ist kein Qualitätsbewertung, es ist eher eine Herangehensweise. Du weißt nicht genau was Du tust und Du probierst was aus. Du hast zwar eine vage Vorstellung und der Kunde will ja was und dann macht man das, aber dann rennst Du einmal in die Richtung und dann wieder in eine andere Richtung und das wird dann so komplex und verbastelt, dass eben nicht wirklich eine echte Struktur dahinter ist."

„Tinkering is not a measure of quality, it is an approach. You may not know what you are doing and you try something. You may have a vague idea told by the customer and therefore you are doing this, but then you may do it one way followed by another way and then it is getting so complex and tinkered that a true structure is hardly to identify."

Gabriel, Development manager

4 Case Study

4.1 Introduction

The case study described in this chapter concerns the embedded system development of a German-based company called Engineering Company (EC – a pseudonym). I describe the activities of developers during the development of project *Theta* (a pseudonym) by drawing on data from interviews with stakeholders inside EC, observation notes, and company documents. The project lasted from September 2004 until June 2007 and involved the collection of data from December 2004 until November 2008 (as per section 3.4).

This chapter is structured as follows: Section 4.2 describes the wider organisational context of EC. This involves a historical perspective on what happened before project *Theta*. In addition, economic characteristics of EC's market place will be explained. The different organisational actors will be portrayed to get a deeper understanding of their background. Moreover, previous development processes will be characterised for a better understanding of EC. Section 4.3 reports on the actions and practices during project *Theta*. This consists of six distinct sections in order to get a comprehensive image of project *Theta*. First, how it all started, including details about the proof of concept, ambiguous and fluctuating requirements, and concurrent projects of EC. Second, challenges with characteristics about the technical setting, hardware and software issues, and facets about the project management. Third, the increasing complexity of project *Theta*, experienced through the necessity for documentation, higher challenges than presumed, new customers, quality challenges, and the necessary system tests of *Theta*. Fourth, I compare the prescribed company standards and compare its compliance with the actions and practices of developers and the project manager. The fifth and sixth subsections recapitulate the raised conflicts and observed solutions. Finally, the seventh subsection finishes with a brief summary of project *Theta*. Section 4.4 describes some customer opinions about project *Theta* in order to get overview about the project from the other side of the counter. Finally, section 4.5 looks at EC for follow-up projects of *Theta*.

4.2 Wider context of EC

This section presents an overview of the historical, economic, personnel and process settings in which EC's embedded system development was situated.

4.2.1 Historical setting of EC

EC was founded in June 2000 by two entrepreneurs – I call them Michael and Mariah. The idea of this enterprise was a threefold specialisation in the environment of the automobile industry. EC concentrated its operational field in the technology areas of information, entertainment and telemetry. Two divisions accomplished this threefold specialisation: the development department and the service supplier department. Since then, EC kept growing, because its products and services gained a good reputation in the marketed environment. Within the first year, EC developed a 16-bit processor platform for the automobile industry. In 2002, EC developed the first product according to customer specification and the 16-bit processor platform. In addition, the other division of the service suppliers had to quickly recruit in order to satisfy customer needs. In 2003 and 2004, the 32-bit processor platform was developed to cope with the increasing demand of processor speed in the automobile industry. The first product, Eta, on that 32-bit processor platform was launched in late 2004. Subsequently, EC gained some knowledge about the development of a 16-bit processor and 32-bit processor platforms.

Soon after the foundation of EC, the company expanded its business operations to two offices, 200 kilometres apart. The central domicile includes the business activities of administration, financial bookkeeping, logistics, and human resources. In addition, a part of the software development and project management was based at this location. The second office included the areas of software and hardware development and project management, quality management, information technology, procurement and marketing. Both offices served as an address for the service suppliers; however, most of them were directly involved in their duties at the customer site.

It is worth introducing some of the attributes of EC's potential customers. The current situation of the automobile manufacturer market in Germany demands high quality, innovative products because some automobile manufacturers see product quality as their trademark. Most of these big organisations employ more than 100˙000 people, hence their research and development departments employ several thousand people. In many cases the sub-divisions operate relatively independently and as a consequence some issues may remain inside the sub-division. For particular projects that involve business contact with external development partners, such as EC, business secrets remain within the sub-division. Before project *Theta* began, EC completed several of these projects with their 16-bit and their new 32-bit platform. During these projects, the EC developers referred to the customer as the whole automobile manufacturer, although the actual customer involved only a relatively small sub-division of these big organisations.

4.2.2 Economic setting of EC

The founders of EC, Michael and Mariah, were previously involved in the development of electronic platforms for a Japanese supplier of the automobile industry. The acquired habits and attributes of that employer were influential for the economic arrangements of EC. Michael and Mariah put emphasis on the importance of the vision for EC. In particular, during interviews with potential new employees, the introduction of EC involved the vision of a "full service supplier". This goal would have involved the development and production of a device, which would have been delivered to the assembly lines of the automobile manufacturer. Michael and Mariah equated this vision with the image of building a cathedral. In that image, four persons were depicted. The first person worked on a single stone with a hammer and chisel. The second part of the image depicted a builder building a wall from stones. The third part of the image showed a person who worked on an arch. Finally, the fourth part of the image exhibited the entire building process. Following this image, Michael and Mariah tried to relate their vision of a full service supplier with that cathedral, which might not be as easy to accomplish, but a vision is needed to set goals. Subsequently, the goal of the founders of EC was not only to supply a service to customers in the automobile industry, but to develop and produce products for that industry. Michael and Mariah were fond of that goal, although entering that competitive market was an ambitious goal for new companies.

Whereas the development department of EC involved a manageable eight employees during project Eta in mid-2003, the service supplier department of EC already consisted of more than 40 employees. Dealing with that imbalance between the service supplier and development department created some tensions. Through the economic success of the service suppliers, they asked for more comprehensive products from the development department. However, the setup of a growing number of service suppliers, mostly within the environment of the customers in the automobile industry, narrowed the relationship between EC and its customers. As a result of that relationship, EC gained more knowledge about the specific needs of some research and development departments in the automobile industry. The growing capabilities reflect the positive development of EC's turnover since its foundation in 2000 (Figure 4-1).

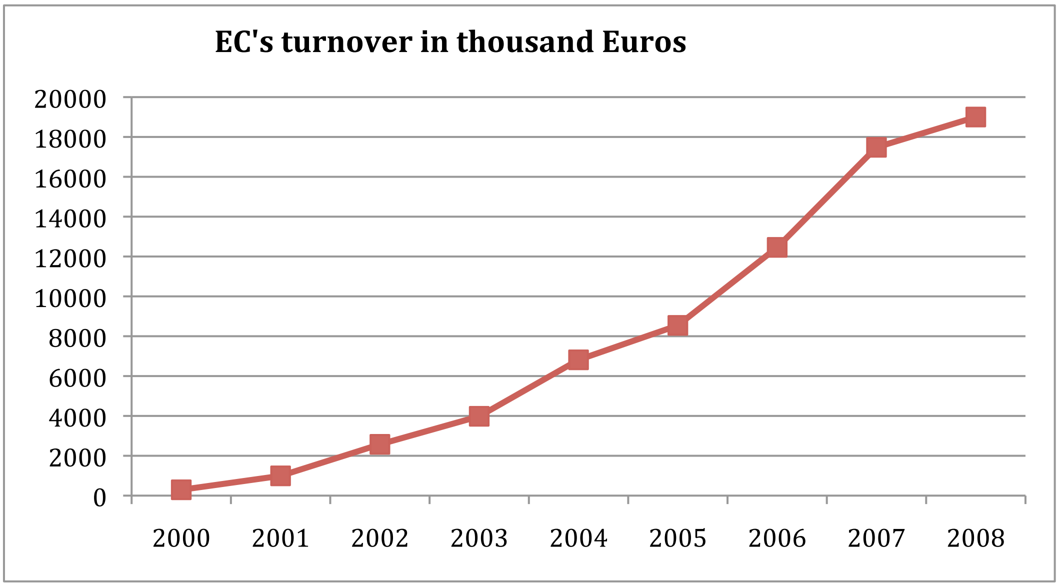

Figure 4-1 EC's turnover (2000 – 2008)

As previously mentioned, EC developed various electronic platforms, such as 16-bit and 32-bit processor platforms, in order to build upon those platform customised solutions. Michael (head of the development department) introduced this approach to EC in order to deal with the economic situation in the automobile industry. This situation was based on the demand of many smaller projects with less than 250 units for the research and development departments within the industry. These smaller projects required comprehensive solutions, which needed to be embedded in the electronic systems of the customers. Lucrative projects were sought by EC and its competitors. Therefore, EC tried to get a competitive advantage by adopting a strategy which involved the development of various platforms.

Before *Theta*, project Eta was, with its 32-bit platform, the biggest project of the EC development department. Eta was based on a Power PC (32-bit) processor with an embedded Linux as the operating system. In addition, a processor for the communication with the automotive environment was required. The products of customers used the Controller-Area Network (CAN) to interconnect the various devices inside a car. A third processor on that platform was used to create a video output signal. In summary, the biggest project of EC before project *Theta* started was a multiple processor platform, which was based on a 32-bit processor. The platform was designed to fulfil the relatively high standards of an automotive environment. Compared to consumer electronic devices, automotive requirements are significantly higher. For example, end-users would also like to use their electronic car equipment on a hot and sunny day, when the temperature inside a car may easily reach more than 75°C. The same electronics are required to work without malfunction on a cold day at less than -25°C. However, product Eta was only a small project, because its use was limited to the research and development department of an automobile manufacturer and the number of produced units was less than ten.

4.2.3 Personnel setting of EC

EC has seen significant growth since its foundation and employed about 100 individuals at the beginning of project *Theta*. As previously stated, the number of employees who were engaged as a service supplier was significantly higher than the number of developers. At the start of project *Theta* there were about nine individuals involved in development activities of EC. However, at the start of *Theta* only six people were involved. Michael, one of the founders of EC, was the head of the development department. During the initial stage of project *Theta* Michael had significant influence (see section 4.3.1). The distributed departments of EC required commuting between the two locations. Gabriel was responsible for the architecture of the new embedded system and he was also manager of the software developers. The distributed location of the software developers necessitated commuting between the two offices, as for Michael. Subsequently, Gabriel spent two days of the week at one office, and the rest of the week at EC headquarters. Michael and Gabriel knew each other long before EC was founded.

Raphael was a senior developer and the key feature of project *Theta* was developed by him. It is interesting to know that Raphael and Gabriel also knew each other before Raphael joined EC. In an interview, Raphael stated:

> "EC is something like a family to me."

The seemingly messy workplace of Raphael is depicted in Figure 9-2. Other developers contributed to project *Theta* also. However, I will introduce other people as their role becomes more important as the case description unfolds. It only remains to mention that besides Michael, Gabriel and Raphael there were a number of junior developers, who had less experience, because they had just finished their university degree. A workplace of one of the junior developers is depicted in Figure 9-3 and provides an obvious contrast to the workplace of Raphael. The nature of EC's usual projects, relatively small projects with a manageable hardware and software architecture, enabled them to be involved quickly in the day-to-day business activities of the development department. The biggest project was the 32-bit platform (Eta) before project *Theta* began.

In addition, other people at EC were influential for the study of the activities of the developers during project *Theta*. David's responsibility was to manage the compliance of the ISO 9001 standard through internal audits. I will go into detail in section 4.2.4, to elaborate on David's role and influence. The role of the project manager – I call him Martin – was vital for the success of *Theta*. Martin spent several years as a service supplier with this customer and knew the circumstances of the people and processes inside the research and development departments of that customer. At the customer site, Martin was involved with the development of a high performance head-unit. One of the technological attributes of a head-unit is the interconnected network of electrical buses. His task was more of a technical nature and after several years Martin sought a new challenge. He was ready to take the opportunity to manage a new development in the company. During the conceptualisation phase of project *Theta*, he spent half of his time still as a service supplier with the initiator of that new embedded system and half with the development team of this system at EC. Therefore, he had excellent knowledge about the customer needs as well as the developers' difficulties. However, project *Theta* soon required a full-time project manager, and therefore he resigned his task as a service supplier. I will go into detail in section 4.3 to expand on Martin's role and influence.

4.2.4 Process setting of EC

More than a year before project *Theta* had begun, EC was certified according to the International Organisation for Standardisation 9001:2000 (ISO 9001) in May 2003. This structured process management framework is a legal requirement if an organisation wants to sell products for the automotive market in Germany. Besides this legal necessity, it formed the foundation of the quality management system of EC. Organisational operations such as procurement, marketing, human resources, quality management, and development were made compliant with ISO 9001. The prescribed processes needed to be followed by every department. For example, the product development process was explained in detail with eight multifarious figures, 27 documentation outcomes, and 19 links to other documents. The management of the database, which kept all organisational ISO 9001 documents on file, was David's responsibility. After David helped to achieve the certification of EC, his tasks involved the regular internal audits of all departments. In doing so, the goal was to manage and support the compliance of ISO 9001 standards and to continually improve the processes within EC.

To continuously improve the processes is part of the ISO 9001 standards, but EC amplified this issue by using "Kaizen". Kaizen is a Japanese philosophy that focuses on continuous development and improvement. The founders of EC, Mariah and Michael, implemented this philosophy at EC, because they felt it to be a useful method to strengthen the quality perspective.

EC underwent an organic growth since its foundation. Before project *Theta*, the organisational hierarchy within the development department involved only three layers. Michael was a member of the board and senior manager of the development section. Gabriel was responsible for the system architecture and software development as a middle manager. The associates who were involved with development activities had Michael as well as Gabriel as contacts. However, Michael had the final say on critical issues.

4.2.5 Summary of EC introduction

This brief summary of EC before project *Theta* started in September 2004 provides insights into the historical, economic, personnel and process settings. The young company distributed its employees at two sites in Germany. Two branches evolved: a service supplier department for the automobile industry and a development department for electronic devices for the automobile industry. The service suppliers, the larger department, was involved in research and development activities directly at the customer site. The development department consisted of nine individuals, including the managers. Before project *Theta* started, project Eta was EC's biggest project and involved a 32-bit platform and embedded Linux operating system. In addition, some of the developers knew each other before they were employed by EC. Martin, the project manager for *Theta* was previously engaged as a service supplier and gained a large amount of knowledge about the needs of the research and development departments in the automobile industry. EC became certified according to ISO 9001 in order to market its products for the automobile industry in Germany. The certification is a legal requirement and involved the detailed prescribed processes of EC.

Table 4-1 provides an overview of the key individuals before project *Theta* started.

Table 4-1 EC's key individuals before project Theta started.

Name (pseudonym)	Role	Responsibility
Michael	Senior manager	Development, finance
Mariah	Senior manager	Service suppliers, marketing
Gabriel	Middle manager	System architecture, software development
Martin	Project manager	Studied *Theta*
Raphael	Associate	Software
David	Quality manager	EC quality

Table 4-2 gives an overview of EC's previous projects.

Table 4-2 Previous projects of EC

Time frame	Development name	Description
Prior to 2003	Platform and products	16-bit processor
2003 / 2004	Eta	32-bit processor

4.3 Project Theta

Project *Theta* started in September 2004 and unexpectedly became the most important development project for EC. The following description provides a detailed overview of the actions and practices of the internal key actors for this project.

4.3.1 How it all started

As section 4.2.1 explained, the various sub-divisions of big automobile manufacturers tend to operate relatively independently. One of these sub-divisions was engaged with the research and development activities of head units. Head units cover various functions for the end customer, such as radio, CD-player, navigation, and so on. From a technical perspective, head units involve various electronic bus interfaces, such as Media Oriented Systems Transport (MOST), Controller Area Network (CAN), Local Interconnect Network (LIN), and so on. These electronic bus interfaces facilitate the communication between the different devices inside a modern car. It is interesting to note that some automobiles may have more than 50 different interconnected devices. MOST is a relatively new electronic network standard, intended to interconnect various audio and video devices. CAN networks might connect engine control units and transmission, or (on a different bus network) connect the door locks, climate control, and so on. LIN is an electronic network bus system, designed to connect less complex devices, such as electronically operated window regulators. Head units may influence many devices inside a car and, consequently, are important components. The development of head units demands high management skills, in order to satisfy customer requirements. The relatively new MOST network standard caused a gap in the research and development activities of automobile manufacturers, because the market provided no equipment for integrated testing of the complex automotive environment with several different electronic bus systems.

Integrated testing was the desired way to test new developments, because all possible data of the different networks would converge and simplify the analysis for further development. This disclosed gap of available equipment was the signal for the manager of the head unit sub-division of an automobile manufacturer to contract out the development and production of about 150 devices. It was intended that these devices become part of the equipment for the employees of the head unit sub-division. The idea was to create a device, which facilitates the tests of researchers and developers. That device had interfaces with various electronic network standards, hard disk to store the recorded data streams, and a power supply according to automotive standards.

4.3.1.1 Proof of concept

Many service suppliers from EC were engaged in the sub-division issuing the contract out of the development and production of that device. Martin, who later became the project manager, was deeply involved in preparing the contract out process of the head unit sub-division. This knowledge may have contributed to the successful bidding of EC, in order get the contract with that head unit sub-division. However, this successful bid of EC was not only reasoned because Martin along with others was involved inside that sub-division of the automobile manufacturer. As section 4.2 briefly described, the development department of EC had with product Eta only one project, which had similar hardware requirements. Due to this background, the head unit sub-division demanded a model as a proof of concept for the most critical component of that development: to record the MOST signal. It was seen as critical because it was a new technology for EC. The most obvious solution to confront that problem was to negotiate with the monopolist of MOST technology a contract that consisted of the exchange of knowledge concerning that technology. Mariah and Michael, the senior managers of EC, held negotiations with the monopolist. It became clear that the business conditions were the crux of the negotiations. The negotiations broke down and EC had to look for other solutions to record the MOST signal. In a peripheral discussion between Michael and Raphael, they came up with an idea of how to achieve this. Neither was sure if their proposal would be successful. However, they thought it would be worth a try. Based on that idea, they quickly designed an adaptor board to connect a MOST network with EC's existing 32-bit processor platform. With the deadline in mind, when EC needed to show the proof of concept to record the MOST signal, the developers improvised a solution, which they thought would be good enough to show the customer that they could cope with that issue. The customer accepted this proof of concept, because the concept tests with a defined log file were successfully validated.

4.3.1.2 Requirements

The contract between EC and the head unit sub-division included a requirement specification of the proposed new device *Theta*. Considerable effort went into generating that list of requirements and, as mentioned earlier, in that initiation process some EC service suppliers were involved along with Martin. Their perspective was the user side, which implicated the high usability of *Theta*. The device should be easily connected to the automotive electronic network, and the recorded data should enable continued processing in the already available software tools. The proposed approach from the user side was that *Theta* should be a manageable plug and play device, both hardware and software. In addition, the people involved in the research and development of head units put emphasis on the new device not intruding on other parts of the electronic environment. In automotive environments, the consumption of power is a highly critical issue, because power is limited when the engine is not running. As a consequence electronic networks of automobiles have stand-by functions in order to save energy. *Theta* was urged to have stand-by functions in order to avoid any harm for the rest of the electronic network. Another technical requirement was the hard disk to store the data streams of several electrical networks simultaneously. Beside these relatively loose descriptions of requirements there were determinants from other parts of the specification of the proposed new device. For example, the connection to the CAN network was prescribed in detail for the hardware chip as well as the software core. In summary, the requirement specification was ambiguous in several parts and the specification of *Theta* also left room for interpretation. However, the initiators of *Theta* were aware of the lack of detail in some parts of the specification and Martin stated:

> "After the first rereads of the draft specification, we needed to improve it. Consequently, the list of requirements kept growing. [...] Basically, the list was a simple accumulation of attributes in the form of a Microsoft Excel file. [...] The desire was to create a relatively simple device to record all data of automobile networks. The gathering and specifying process of *Theta*'s requirements behaved like a learning curve. At the beginning, we had an idea and this idea improved into a more sophisticated requirement profile. However, the requirements kept fluctuating during the entire development phase."

4.3.1.3 Concurrent projects

During the time when the proof of concept for project *Theta* succeeded, another project, Iota, kept the developers busy. Iota was a highly confidential project, because it involved the connection of high tech products from consumers in the automobile entertainment environment. This was at that time, autumn 2004, a new technology, and required a sophisticated understanding of high tech products for consumers. New human resources tried to cope with the increasing workload. EC hired a new employee, who had recently finished a university degree; at the beginning, this new member of staff needed some assistance to become accustomed to the new environment. In addition, Martin needed to become familiar with the development department of EC. As mentioned earlier in section 4.2.3, as the assigned project manager of *Theta*, Martin spent half of his time with the EC developers. The 32-bit platform, which was already available at EC, attracted interest, because it was seen as a potentially good basis to develop project *Theta*.

The project management activities involved another interesting detail of this case study. As stated earlier in section 4.2.3, Martin became the *Theta* project manager. He seized the chance to attain a more challenging position by moving from the service supplier department to the developers' department. However, project *Theta* started as just another development project. Concurrently, Iota was another project in its initial phase of development and Eta was not finished. As a result, the developers' department, which involved ten people at that time, was very busy. With few human resources, the time management of every developer was critical. One way to cope with that problem was to prioritise tasks in order to get the most important jobs done. Developers commented:

> "Practical thinking and doing, this was our way of doing things at that time". "Skills and the knowledge from previous projects helped us enormously at the beginning of project *Theta*."

And Martin stated:

> "Learning and doing was the prevalent way, although some issues had been taken over from previous projects. The issues that had been taken over included some known problems, which we were not able to solve in the given time."

Subsequently, to synergise the efforts of the developers for the concurrent developments of Eta, Iota, and *Theta* was a demanding task for the project manager, particularly at the initial phase. In addition, there was little support from the senior management, because it was obvious that project *Theta* would not become a product to hit the assembly line of the automobile manufacturers. For the senior management as well as the developers, project *Theta* was just another project, which should result in a manageable amount of devices with limited impact on the market.

4.3.2 Challenges

4.3.2.1 Technical setting

After this preliminary introduction of the initial phase of project *Theta*, I describe the early phase. This includes the time between the accepted proof of concept and the first models which were delivered to the customer. The successfully realised proof of concept was the signal for all developers that project *Theta* had become a real project. Michael set with the platform strategy a basis that could be used to develop various devices. The 32-bit platform provided enough computing power to deal with the requirements. The CAN network connectivity required another 16-bit processor, which was requested in detail from the customer. The MOST network connectivity was implemented with a Digital Signal Processor (DSP). The central processing unit was a 32-bit Power PC, which connected the CAN processor, DSP, hard disk, and other units such as memory and input / output connectors.

The software was based on an embedded Linux operating system. So far, everything was set up as a sophisticated plan; the hardware layout was sent to a subcontractor to assemble the different units. The software architecture was largely based on the unwritten experiences of software developers and the system architect (Gabriel). In addition, the project required another piece of software, of which EC had no experience. This piece of software, called "client", was based on a Microsoft Windows environment instead of the embedded system. It was needed for the easy connectivity of the device with the systems available for the researchers and developers of the automobile manufacturers. The recorded data of the device needed to be easily transferable with the client, in order to manage and analyse the data at a different place. In addition, the user needed to be able to update the embedded system via the client. It was started from scratch, so the new developer was asked to begin with that task. It was a lucky coincidence that Martin had some experience with the development of software for a Microsoft Windows operating system. Therefore, Martin became the architect of the client as well as carrying out his duty as the project manager of *Theta*.

4.3.2.2 Hardware issues

With the delivery of first devices from the subcontractor began the proverbial brazing and soldering by the two hardware developers. The late delivery confused the schedule and EC renegotiated with the subcontractor in order to improve the relationship. However, the relationship between EC and the subcontractor was ill-fated, because the quality of the delivered hardware boards left room for improvement. As previously mentioned, the level of experience with complicated hardware boards was good enough to realise that the job of the subcontractor could have been performed better. However, their confidence in developing error-free hardware boards remained, although they kept falling behind schedule. Their response was to improve the design of the board with a new layout of the circuits and other bug fixing. In the meantime, they helped themselves and the software coders with direct wiring on the defective board by brazing and soldering. The development and later correction of the first hardware boards took a lot of time. The hardware developers commented that their activities and practices involve planning, because the alteration of boards involves the valuation of potential side effects (such as electromagnetic compatibility) and preparation in order to assemble these high tech boards. However, direct wiring on the defective board was the only way to cope with the defects and avoid further delays. The planning did not imply the understanding of detail in the design of some circuits. The hardware developers commented:

> "Some errors looked like random acts of nature, some errors have been known or foreseen. [...] With the design we first tried ordinary solutions with fundamental background. With the deadline getting closer more quick and dirty solutions occurred. [...] With most circuits we know why to design the background in this way. However, there are several parts, where I do not know why it is done exactly this way. We just trusted

> these parts of the design, which had been accomplished by somebody else."

4.3.2.3 Software issues

The software developers were anxious about the problems with the hardware board, because any source of error during that stage of development could not be detected as easily, because neither the hardware nor the software could be characterised as absolutely fault-free. To cope with the hardware issues at that critical stage of development was challenging for all developers involved with project *Theta*. To gather and process the MOST signal was the major problem during that stage of software design. What was difficult was not the connection between the different hardware units, but the interpretation of the recorded MOST signals, because it required a deciphering of the translation table between the MOST signals from a media layer into a host layer. This translation table managed the conversion between the pure signal transmissions into a sophisticated data representation. Raphael was responsible for the development of that important feature of project *Theta*. Raphael's working style may be characterised as tinkering and 'fiddling', by which I mean he was an individual who like to doctor or play with something by making small adjustments. The result of those activities and practices was an embellished understanding. This does not mean to follow a prescribed plan, necessarily, but a series of actions might imply a deliberate approach and it might also imply a completion of possible options. However, that list of options might be compiled as a result of previous tinkering and fiddling. It is interesting to note that Raphael described his activities as 'normal common sense' and he emphasised the importance of motivation. He commented:

> "Actually, I am just a simple straightforward, piddling programmer. That is my task at EC. [...] I think you may discard my Engineering degree, if you talk about my approach to technical challenges or problems. I am more of an autodidact with some experience and passion. [...] However, to tinker might be the wrong word to describe my approach. Though it is

> not a scientific approach either. I try to accomplish tasks goal-oriented, somewhere between creative chaos and structure."

In addition, other parts of the software development were also troubled during this early stage of project *Theta*. The embedded Linux kernel required adjustments to the new system. Timing issues between different units required several tries to measure and test the correct calibration. According to the developers who were involved with those tasks, learning by doing was prevalent. In contrast, Raphael described the activities of these developers more as a structured solution finding. However, the time pressure during that period of the project was difficult; in particular, the hardware errors were cause for concern. One reason for concern was the necessity of new adjustments, when new hardware would arrive on their desk for further development.

Despite these worries, the client development progressed as expected. Martin was deeply involved in the architectural foundation of the software, because he accomplished similar tasks in other environments. The new developer, who was assigned to the client development, was accustomed to that work. She got the impression of 'her' project, although Martin helped at the beginning. Finally, another developer was assigned with the programming of the CAN part of the software. The developer of that part collected some experience with previous EC developments. He commented:

> "At EC we know about the quality issue, that we should strive for the best solution. So, we take the individual developer as responsible for the code she or he programmed. [...] However, sometimes errors are unclear. As a consequence, a structured solution is not possible and therefore errors were patched, hacked or some workarounds were created to solve the problem."

It is interesting to note that the motivation was high at that phase of development. Many developers professed great enthusiasm during that time. Some developers commented that high goals with that project were challenging but also encouraging. Raphael liked the immediate sign of success or failure, if the code was implemented in the hardware. Many developers realised that the Pareto principle was generally true, and that during the first 20 percent of the time, 80 percent of the requirements might be accomplished. That instant response of individual activities may have stimulated the high ambitions of most developers at that time. However, with growing knowledge about some attributes of the hardware and software, the requirements changed frequently. These fluctuating requirements during that early stage of development were less harmful for some developers, but others expressed their concern.

4.3.2.4 Project management

Martin, *Theta*'s project manager, experienced a variety of challenges during that time. The problems of the hardware required a lot of extra work. Martin called for additional work from *Theta*'s developers; it was not as difficult at the early stages of project *Theta*, because most developers stated that they were highly motivated. During that time, the 40-hour week of developers could easily become a 60-hour week. Martin also commented that most of the prescribed ISO processes from EC were not really helpful. Moreover, the defined organisational processes did not encompass necessary tasks he was required to do during that phase. For example, the ambiguous and fluctuating requirements needed constant negotiations and explanations as to how various details should be understood by the developers. The developers did not have a feeling for the circumstances of the customers, and the list of requirements was at several points incomplete or vague in matters of detail. However, the ISO documents called for one final and complete list of requirements in order to start the development. Constant contact and renegotiations about technical details were not allowed by EC's ISO standards. However, Martin recognised the need for his approach.

Another challenge was the spread of responsibilities within EC. The expansion of EC was organically structured at the developers' side, because the individual developers and the project manager were organised as almost one united team. Despite the local distribution of developers, more or less, the developers shared the same large-scale office and a similar team spirit. However, Martin was not fond of, that Gabriel and Michael could change the priorities of the available human resources, hence to reduce personnel for the development of project *Theta*. In Martin's opinion, the project manager should have had responsibility of the human resources that were involved in the respective project. However, EC had limited human resources and several concurrent projects at that time. Nevertheless, Martin's role was very important for project *Theta*, particularly at that early phase. The reasons for that were recurring negotiations and explanations between both sides of the project that were more than appropriate. Martin was an ideal candidate to fulfil that role, because he already knew the customer very well and he became familiar with the developers very quickly.

4.3.3 Increasing complexity

4.3.3.1 Producing documentation

The progress of project *Theta* highlighted an increasing complexity of what was formerly "just another project". The requirements, collected in a simple electronic table file, kept swelling, because the details became more finely graded. Part of the development culture was the weak documentation of their work at EC. The part of the software that was ported onto the device was affected more by that circumstance, because the people involved with embedded software concentrated on coding. Though prescribed procedure at EC was to write a specification beforehand, many specifications of *Theta*'s embedded software were written just when it was necessary for the forthcoming visit of ISO inspectors. This issue will be further detailed in section

4.3.4. This context challenged the embedded software developers more than usual, and they coped with that situation with some black humour, such as:

> "The best documentation is the code itself. When you have a working device and look at the code, that is the best documentation you may get. [...] We started to realise, that project *Theta* got bigger, and as a consequence, some 'more' documentation might help. But at that time, we had no time for documentation. And as far as I understand, you may also build crap with documentation. So, we concentrated on the final product, instead of producing paper."

4.3.3.2 Higher challenges than presumed

Gabriel commented that several ambiguous parts of the requirements turned out to be far more difficult to realise than previously expected. This expectation was based on the point of view from developers, without any further experience of the customers' situation. The developers' view involved the more technical aspects of project *Theta*, such as necessary processors, interfaces, and power supply. In contrast, the customers' perspective entailed more the perceivable aspects of that new device, such as reliability, non-interference with the tested automobile environment and easy usability. These two different languages of understanding needed to communicate together, and in the fine, graded issues of project *Theta,* trouble was pre-programmed. With increasing complexity, the project manager's negotiation skills became more important. Gabriel claimed:

> "One of the problems of the requirements was the different understanding between us and the customer. Though at the beginning we thought we had a common understanding, we were disabused. In addition, I tried to reuse some parts of the previous projects, including some architectural issues. This turned out to be a constraint throughout the development. Moreover, the client issue was more severe than previously expected by myself. Martin identified this, and put some effort into the client development. I, as a developer, would have been

> happy with a text-based interface to configure *Theta* and download data via that simple interface. I am happy that the client software was so good at the end that the customer was almost surprised by its usability. A graphical interface was certainly a more convenient solution for the customer. [...] The unrealistic schedule, which had been approved by the customer and Michael, was one more factor that constrained the development. As I commented, we thought to reuse more parts from project Eta, but this hope was dashed. With the progress of project *Theta* it became obvious that the schedule could not be achieved."

Martin played another important role in the dimensioning of several ambiguous parts of the requirements for project *Theta*. As commented earlier in section 4.2.3, Martin was previously involved in the research and development activities of the head unit sub-department of an automobile manufacturer. Martin's previous placement as a service supplier was very helpful, because he knew the problems of the customer from his own experience. When it was necessary to realise ambiguous elements of the requirements, Martin contributed to the decision-making process by asking himself: What would I do in that situation as a user of *Theta*? Though the perspective of the user played an important role, at some point the managers of EC (Martin, Gabriel, and Michael) had to decide which features should be developed and which should not. These negotiations were a difficult decision-making process.

4.3.3.3 New customers

The delivery of the first models took some pressure away from the shaken project development and management team. Some developers of *Theta* expressed their satisfaction with that achievement. In several negotiations, Martin explained to the customer the reason behind the schedule delays, but the customer was excited about having the first models in their hands. Both parties knew that this success was not the end of the development, however the handing over was an important milestone of project *Theta*. The customer used them from the first day for tests on board automobiles or on test benches. A significant amount of these first users were service suppliers, who got their pay cheques from EC. The communication between these service suppliers and Martin was open and informal. They provided valuable feedback about their experiences with the usage of *Theta*, in addition to the formal contact between the customer and Martin. After praising the success of *Theta*, some critical issues were raised. Most expressed concern with the quality of the device. Several malfunctions troubled the reliable record of data inside the automobile.

Despite those malfunctions, the message of that new device and its benefits started to spread beyond the head unit sub-department. Other sub-departments became aware of this useful device and started to ask questions of the first users and their managers. These sub-departments were concerned with other areas of research and development of electronic devices inside modern automobiles, such as MOST amplifiers, MOST telecommunication units, MOST CD/DVD changers, and MOST radios without navigation. In addition, sub-departments that were dealing with verification and validation of electrical components as a whole for research and development as well as production became aware of that new product. In conclusion, the setup between the initial customer and EC of the formerly 'just another project' became the starting point to gather a lot more customers for project *Theta*.

The interest of these new customers increased and most wanted to participate in the current stage of development in order to influence the final product for their own specialised tasks, hence there were problems. Martin stated:

> "We needed to bring together the various new requests of the customers and the possibilities of the advanced project *Theta*. After the creation of the first models, [the customers] tried to implement more features. Most of these features were based on slightly different and partly new software. However, we brought them back to earth."

The departments with a slightly different focus of operations resulted in different requirements for *Theta*. Consequently, the list of requests kept growing and became more complex.

Several software and hardware developers stated:

> "The customer desired the all-in-one device suitable for every purpose. We were lucky at some point they understood that this was not possible."

Gabriel commented:

> "The customer did not really know what he wanted. So, he came out with the wise solution to integrate every possible idea. However, we needed to cap these flourishing ideas and put the features into a state that was realisable. [...] Some of the desired features were truly born in dreamland, because they were hardly or not at all achievable due to technical limitations. Using these different and partly contrasting requirements, it was Martin's responsibility to derive the baseline requirements. Martin's mature understanding of the needs and wants of the customers was certainly an advantage."

4.3.3.4 Quality challenges

Facing an increasing demand for *Theta*, EC started to realise that this was not just another project as previously thought. In order to cope with that development, EC's biggest concern was the quality of *Theta*, because the fault return rate of their devices was too high. The investigations about that problem highlighted hardware problems because of the insufficient quality of assembled boards by the hardware supplier. Compensation negotiations and corrective actions resulted in unsatisfactory quality improvements. Because of this situation, EC decided to close a deal with a new subcontractor. Potential products with higher quality were a key factor for this decision.

In addition, the new contractor was able to deliver more assembled boards. This increasing demand reflected the growth of interest of the initial customer and new customers. As noted earlier in sub-section 4.3.1, at the beginning the initial customer demanded about 150 devices. However, with the delivery of the first models, the other departments became aware of the perspective of project *Theta*. The perspective of an increased degree of automation combined with easy usability encouraged other departments in the area of research and development of automobile electronics to get involved with project *Theta*. This involvement of other departments, hence new customers, in project *Theta* resulted in increasing demand, much higher than a mere 150 devices. These new circumstances – an increased number of devices, managing several customers with the same product, and the new subcontractor – increased the complexity of project *Theta* significantly.

4.3.3.5 Testing Theta

Moreover, the client software, based on Microsoft Windows was going to be more complicated than previously expected. In order to cope with that situation, EC hired new employees to support the client development. At the same time, Martin and Gabriel raised issues about quality concerning the practiced testing methods of EC, because these had not been coordinated. Previous projects at EC were relatively small and manageable, whereas project *Theta* became more complex over time. The test methods were the responsibility of the particular coder and, therefore, every software developer tested only her or his contribution to the project. This method resulted from the different programming knowledge of the software developers, because the various pieces of software were differently transcribed, such as software for the CAN chip, DSP chip, embedded Linux software and client software. This test method has been relatively successful with previous projects of EC. However, Martin and Gabriel were concerned about the increasing complexity of the software in general and the number of new customers. With this background, Martin and Gabriel spent more time thinking about different ways of testing to raise the quality of the product through higher regimentation of the development procedures. Gabriel commented:

> "At that time, we looked for ways to get reliable information about the capabilities of that device. We required these data to decide which wheels may be adjusted to improve the entire system. Initially, developers were more concerned about their own area of activity and less about the activities of others. Martin and I kept a more holistic view, to progress the development of project *Theta*. However, the data served the purpose to make decisions on a more sophisticated level and to avoid more or less some of the decisions based on gut feeling."

In addition, Martin stated:

> "When project *Theta* became more complex, I wanted to guard against mistakes which are avoidable. For that reason, we put more effort into system testing, which had been totally disregarded. The arranged time for software, hardware and system testing was part of the schedule, but it has been misused as a time cache in case of delays during the development. However, what and how to test has not been really thought through at the beginning. Anyway, the increasing complexity messed up the initial schedule and we renegotiated these delays with the customer."

Figure 9-4 depicts one of the typical test benches of product *Theta*.

4.3.4 Company standards

The growing complexity of project *Theta* required a more sophisticated approach in dealing with challenges. As previously mentioned, many activities of developers and the project manager at the beginning of project *Theta* involved a less coordinated course of action, but were affected by individual experiences, team spirit, optimism, and procedures for finding creative solutions. However, the conformity with the company's process management framework was very weak. Some organisational standards were partly followed, whereas most activities by developers and the project manager were not influenced by the organisational set of prescribed processes. In addition, Gabriel and Martin expressed their concerns about the process quality and the lack of advanced testing for this complex device.

These thoughts were nourished by previous experiences of EC with ISO 9001 audits. These audits needed to be accomplished on a regular basis, in order to be a certified ISO 9001 organisation. David helped with regular internal audits to achieve the ISO 9001 certification; hence David's goal was to improve the conformity with the company's process management framework. Upcoming external ISO 9001 audits caused a series of actions in order to catch up with company standards. Open tasks from the previous ISO 9001 audit were at the beginning of this series of actions. These open tasks usually involved minor improvements of organisational documents of the process management framework. Second, obvious lacks of organisational processes are going to be analysed by the quality manager. It is important to know that the quality manager typically takes a more holistic approach to analysing organisational processes. The processes of development activities are a small part of the organisational set of prescribed processes. If corrections are necessary to the current set of prescribed processes, the quality manager would initiate an alteration of the respective process descriptions and flows. Finally, the quality manager would instruct the relevant employees in order to fulfil the new prescribed procedures. In conclusion, David's task to prepare an ISO 9001 audit by external examiners was manifold and required a holistic understanding of organisational processes.

4.3.4.1 Lack of conformity between company standards and practices

In this section, I want to elaborate several observations of non-conformity between company standards and practices by developers as well as the project manager.

4.3.4.1.1 Requirements of Theta

Several procedures at EC lacked conformity and needed some adjustment. The requirement elicitation and management processes were differently practiced in terms of the processes prescribed. First, I describe the prescribed processes in detail, followed by an interpretation of the nature of the requirement elicitation and management processes.

EC's database of structured processes entailed fourteen documents that relate to activities of requirement elicitation and management. First, the initial overview of requirements was gathered through an enquiry by the marketing department. An integral element of that enquiry was a business assessment (EN.72.PB-2), such as manufacturing costs per unit, costs of development, development schedule, and risk analysis. A positive decision about the start of development involved a more detailed business assessment (EN.73.PB-3). The processes of developing a product were related to the project planning. The product development procedure was sequential and encompassed six prescriptions: project control, control of development documents, project requirements, development planning, preparation of specification, and quality report (EN.71.AA-1, EN.71.PB-1, EN.72.AA-3, EN.71.AA-2, EN72.AA-4, QM.42.FB-21). In addition, other documents were requested from the customer (customer requirements) or build (list of responsibilities, project plan, project requirements, specification, specification review report, and so on), according to the stage of the product development procedure. After the customer approved the specification, the development was distributed into four main strands: hardware, software, test hardware and test software development. During this course of action there was no possibility of influencing the activities. However, there were some processes prescribed (EN.83.PB-16), when an alteration in the development was necessary. This alteration process was sequential, and involved alteration reports, customer renegotiations, specification alteration, specification review and customer acceptance. It is important to know that the alteration process dealt with changes that were necessary, because requirements or the combination of requirements would not be realisable. Finally, the four strands of distributed development entailed another series of defined processes. However, these sub-processes obtained their requirements from previous process step outcomes during the detailed business assessment. Every sub-process may involve a specification concerning the strand, such as hardware or software.

Figure 4-2 is an extract of the multifarious structured process prescription of EC's development process.

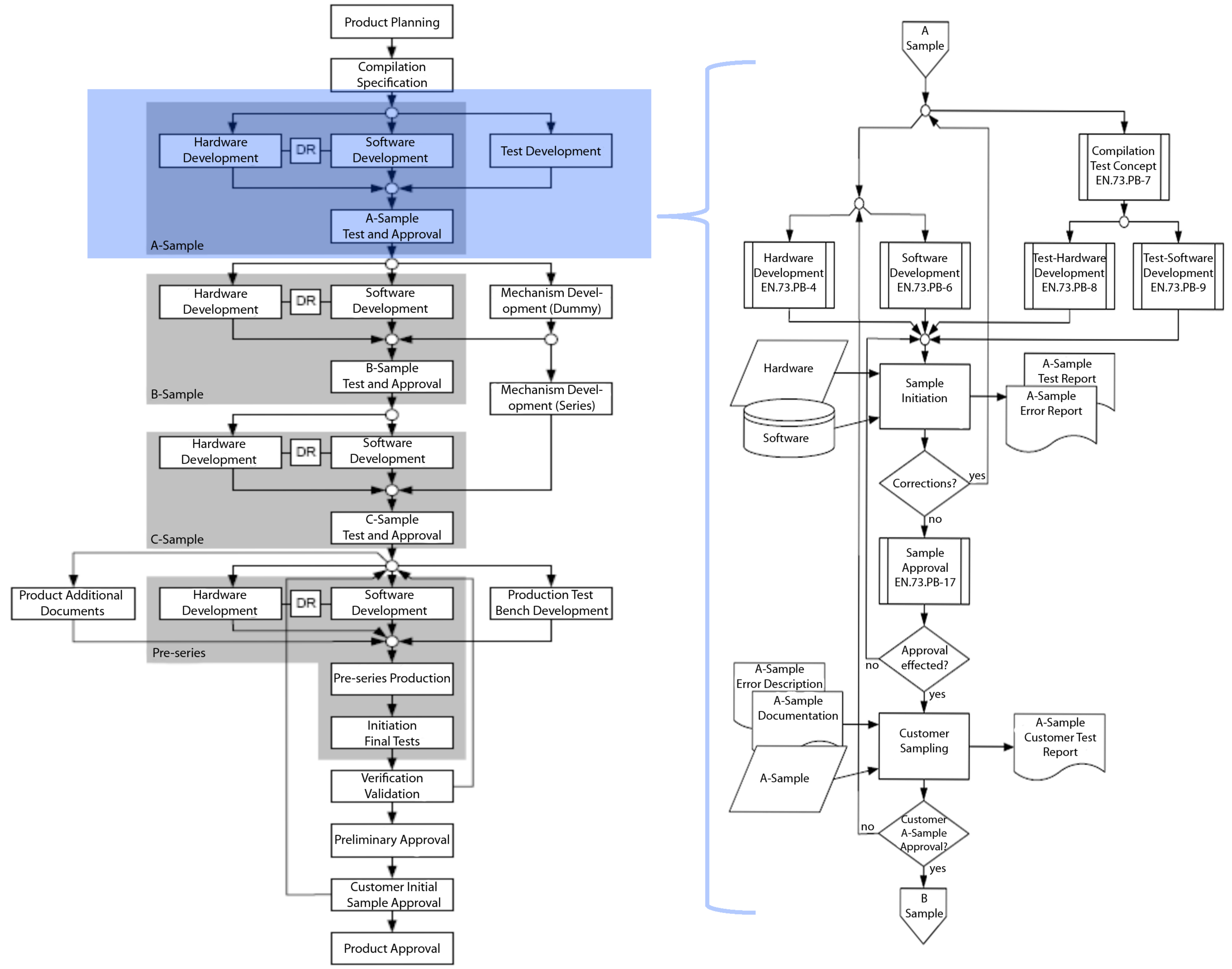

Figure 4-2 Distributed and detailed development process of EC (extract)

Although these prescribed processes may provide a detailed set of guidelines for elicitation and management of requirements, the nature of the practiced actions by developers and the project manager was different. Yet, it was fortunate that the initial overview of requirements was gathered through a business assessment, involving information about manufacturing costs per unit, costs of development, development schedule and risk analysis. However, these data were more or less a rough guess from the developers. Gabriel commented:

> "Usually I do not spend a lot of time with these assessments. On one side, it is difficult to decide properly, because when they want exact figures, the list of requirements has a tendency to be vague. On the other side, my figures are usually inaccurate, because I want to take care about every part of the requirements. Therefore, the development would take far too long to be placed in the market. So, to find the balance during this early stage is very difficult."

The positive decision of the customer in the case of *Theta* involved the development of a proof of concept, as earlier commented in section 4.3.1. Besides this unforeseen case, the product development procedure did not follow the sequential guideline. With the final approval of the proof of concept, Martin and the customer's contact people were in constant negotiations and discussions about the project. The prepared specification of product *Theta* was only a rough description of the foundation of *Theta*. It involved a vague language and in some parts absolutely no detail. Martin kept discussing the details for several months, according to the particular upcoming decision that needed to be made. This communication between EC and the customer were not based on any prescribed process, but on doing what was necessary in order to bring the project a step further. Martin stated:

> "We experienced a learning curve with the requirements. At the beginning, the fundamentals were relatively easy to accomplish. In a

> later stage of the development, we needed to discuss the issues with the customer constantly."

In addition, the specifications were not updated, because the developers and the project manager stated that they had no time for this task. Martin commented:

> "At the beginning it was relatively hard to convince developers to write documentation. In particular, the updating of documents was only done if you asked the developer to do so."

4.3.4.1.2 Developing Theta

Furthermore, the testing procedures during the development were differently practiced in terms of the defined processes. After a description of the prescribed processes I will describe the actual testing practices of developers and the project manager.

As earlier described, the development was distributed into four main strands after the customer approved the specification. The idea was to start concurrently the independent development of the device (hardware and software) and its test device (hardware and software). Unbiased development teams of the device and its test device examine the specification from different perspectives. At the end of development, the different strands merge and a better coverage of the given specification may evolve. This approach is the core idea of the distributed development according to the prescribed processes of EC. In addition, the development of the test device is described in detail. The approach covers a sequential execution of tasks, similar to the approach taken for hardware and software development. The first step involves the preparation of a test concept, with the result of a test specification of the individual project. The test concept needs to be evaluated in a test audit. Positive results of that audit of the test specification initiate the preparation of test instructions for hardware, software, test systems as such and test stations. By the time test instructions are completed in a sequential approach, an audit is needed to evaluate the results. Positive results initiate the concurrent development of test hardware and test software. However, the development approach of test hardware and test software is sequential and similar to the prescription of hardware and software of the tested device. They involve the adherence of several prescriptions and the compilation of hardware layout diagrams or software versions. With final evaluations, the test hardware and test software might be approved. With that approved test equipment, the different strands of hardware and software and test hardware and test software join for setting-up operation. With the initiation of the system and its test system, the development team needs to decide if any adjustments are necessary. If so, the development would return to the stage at which the different strands separated into hardware, software, test hardware, and test software development. If no adjustments are necessary, the sequential approach leads to an official sample approval. With positive approval, the device would be delivered to the customer in order to get feedback about the device. At that point, the test strands of test hardware and test software remain stable. In conclusion, the test development processes of EC, based on ISO 9001, are completely structured and integrated in the other development processes of EC. Similar to EC's other hardware and software development processes, the test hardware and test software processes are sequential and involve the adherence of prescriptions and the compilations of documented work products.

Figure 4-3 depicts the development process of test software according to EC's standards.

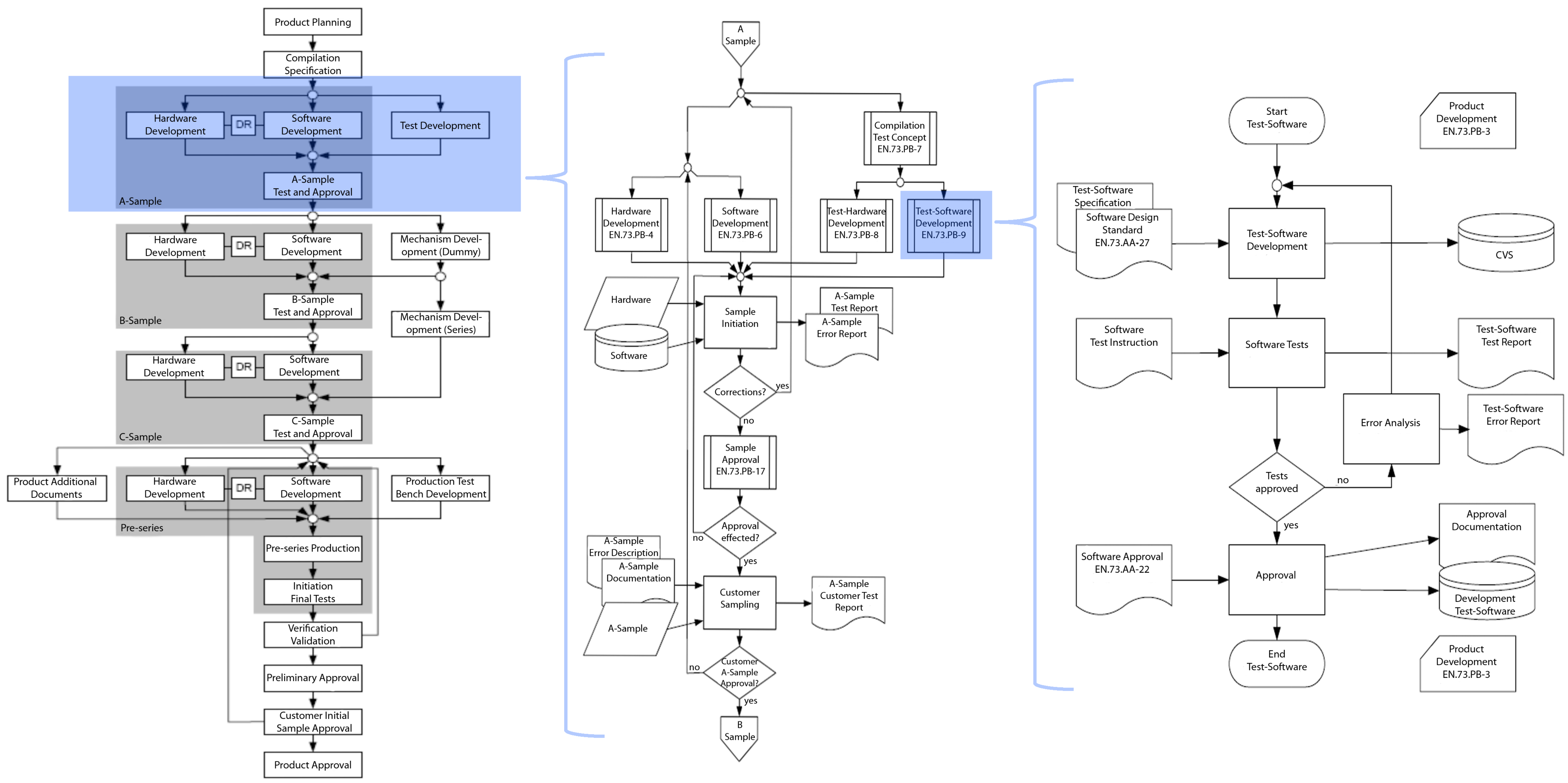

Figure 4-3 Development process of test software according to EC's standards

These defined development processes would provide a set of principles for the integrated testing approach. However, the observed and questioned actions and practices of the developers and the project manager exhibit a different picture. The entire test procedures were neglected until the project was at a very advanced stage of development. Many developers commented that the development of the test concept, test hardware, and test software was not given enough attention, although some developers were aware of the quality issue and they strived for the best solution. Individual responsibility looked to be the proper way to cope with the lack of testing. One developer commented:

> "Some of the errors were unclear. I had no explanation for them, even when I discussed possible reasons with other developers, such as Raphael. I had no sophisticated solution ready and therefore some errors were patched with some workarounds."

In addition, Martin stated:

> "The schedule for project *Theta* left no margin for error. So, at the initial stage, the validation and approval of the system was not really planned. This time slot should provide a buffer for the development in general, but a sophisticated test scenario was never part of our considerations at the initial stage of *Theta*. With the progress of that project and its unexpected increase, thoughts about testing started to rise in our heads. […] We needed to do something about this, because we experienced a lack of quality with the relatively high number of malfunctions. So we hired new people and began to increase our efforts for a more sophisticated testing of *Theta*."

In March 2005, about seven months after the start of *Theta*'s development, EC concentrated their efforts on testing. EC started to hire new developers and began to set up test hardware and test software for *Theta*. The large number of features was rather difficult to test; therefore, the test equipment development focused on regression test plans to cover the fundamental features of *Theta*. Similar to other developments of EC, the test device was set up from previous experiences of EC. A test equipment developer commented:

> "The test equipment needed to be available very quickly, because at the beginning of *Theta*'s development no one thought about testing. The basis of the test equipment was a flexible, modular structure, which was developed relatively easily. The time constraint was always there and therefore we did not have a sufficient period to validate the test equipment. Actually, all tests were conducted during the development itself. I remember that at the initial stage we had a rough concept written in a specification, but very soon we did not pay any more attention to the specification and hacked the device."

4.3.4.1.3 Documentations

Another lack of conformity to EC's prescribed processes involved the issue of documentation. It is interesting to note the six main instructions of procedures implied in all 63 different process outputs in the form of documentation such as reports, specifications, approvals, and so on. The software development consisted of five different documents: software specification, database reports (to save the source code in a system called "Concurrent Versions System" (CVS)), update of project plan, software test reports, and software error reports. These documents are required to accomplish a process output of every sequential step during the software development. However, the software developers were focused on the creation of source code to realise their part of project *Theta*. Many software developers did not recognise the value of documentation and expressed concerns about that process requirement. Raphael commented:

> "In my opinion, the code itself is the best documentation you may create. Everyone who can read the code can distinguish its purpose, without any other documents. Moreover, I can spend more time on writing proper source code instead of writing additional documents."

Other software developers shared a different understanding about the value of documentation; however, the tight schedule of project *Theta* forced them to write specifications when there was enough time available. A software developer stated:

> "Well, we all know that you should write specifications before you start coding. However, the tough schedule forced us to deliver results quickly. With that work output targeted orientation you postpone everything that might distract, such as writing documents."

4.3.4.1.4 Subversion and Bugzilla

In addition to this general lack of writing documentation compared to the suggested company's standard, software tools were extensively used, which would imply different prescribed processes, as discovered in this case study. As previously explained in section 4.3.4.1.3, a database called CVS should have been used. Instead of using CVS for project *Theta*, the software developers used a tool called Subversion. Subversion is open source software to control different software versions in a system and it is very similar to CVS. Subversion is a database system to file and structure the progress of various software modules, and helped the software developers throughout project *Theta*. One software developer commented that Subversion was a valuable tool for addressing revision control, because they needed to deliver new versions of the embedded software as well as the client software early and frequently. One software developer commented:

> "I introduced Subversion and Bugzilla to improve the software development foundation at EC. These tools are helpful and necessary to develop. Both are open source and fit within our environment at EC very well. Other issues would also be helpful, such as a knowledge databases, but time constraints inhibited that implementation."

At one site, software developers introduced Subversion just at the start of project *Theta* and previously used CVS. At the other site, software developers gathered preliminary experiences with Subversion and supported the company wide introduction.

The usage of another piece of software, called Bugzilla, influenced the processes of EC in a different way. Bugzilla is open source software for the purpose of bug tracking. Bugzilla includes procedures to process the reports in a standardised way, and the software developers followed these procedures. These procedures counteract the existing structured processes of EC's ISO 9001 standard. The processes are more finely structured and offer various ways to handle errors in different situations. The developers and the project manager commented on the positive influence of Bugzilla for the management of errors. Finally, it became a standard procedure to report unexpected incidents and errors with the help of Bugzilla. Particularly with the first available samples, many software developers claimed that the direction and list of tasks concerning the development of *Theta* was to a large extent influenced by stakeholders' reported incidents and errors. Similar to the situation with Subversion, with the start of project *Theta* Bugzilla was introduced company wide.

4.3.4.2 Conformity between company standards and practices

More importantly, I discovered some conformity between company standards and practices by developers and the project manager.

4.3.4.2.1 Generic steps of development

The generic steps of development remained, as described in the company's structured processes. The uncovered discrepancy between EC's ISO 9001 standards and their practices cannot hide the fact that embedded system development is a process of merging two complex scopes together: hardware and software. Hardware involves the physical artefacts of electronic components, designed to meet the requirements of project *Theta*. Software includes the programs onboard some electronic components of *Theta* as well as the client program running in a Microsoft Windows environment. Hardware requires software reciprocity; consequently, the development is carried out concurrently as in the structured processes prescribed. When the development merges both scopes, the initiation of the system is critical for the progress of the project. According to the evaluation of the system as such, the development needs to pass through another development loop with the result of improved hardware and software. However, the loop time of hardware may be significantly longer in contrast to software. The reason for this discrepancy is the length of time for constructing new design layouts and assembly of the electronic components. Practical experience with previous development of EC showed that at least two loops of new hardware were necessary to develop a system good enough to meet customer expectations. It might be necessary to alter the software configuration that software runs on new hardware. The outcome of this is the necessity for a concurrent development of software and hardware that merges at a certain time to improve the system as such. In addition, well-developed hardware might serve as a stable basis to further develop the embedded software. The company's prescribed processes as well as the practices during project *Theta* reflected the same picture of these generic steps of embedded system development. Michael stated:

> "Well, a vital part of our philosophy was the embedded personal computer for the automotive environment, with attributes such as temperature resistance, high electromagnetic compatibility, and low power consumption. Project *Theta* was based on that device and through consecutive planning we developed hardware as well as software to fit the purpose of the customer."

In addition, Gabriel stated:

> "Indeed, our philosophy was to take something that we had done before, put some new required features onto it, and try to make it run. The same happened with project *Theta*. We had the previous 32-bit platform and merged it with the MOST part. Then we tried to get it running and improving with some cycles of new hardware and software. Actually, I wanted to take a lot more from the previous 32-bit platform. However, during the initial stage the customer and the project manager wanted to accomplish some features differently."

Raphael commented:

> "The principles of software engineering remain, of course. However, I need to confess that with the increase of project *Theta* I experienced a change in my personal practices. I want to say that my engagement with

a more structured approach is slightly better compared to the time when I started at EC. Information and assessments about ISO 9001 and SPICE keep knowledge and experience about more structured approaches awake. Structured approaches mean thinking about a concept and writing documentations."

4.3.4.2.2 ISO 9001 certification

An important issue for EC was to remain ISO 9001-certified. Therefore, the quality manager conducted internal ISO 9001 assessments regularly. This is a legal requirement to obtain the certificate. In the present case, the checklist for the developers involved 44 points. However, the issues were kept generic, in a way that assessors may interpret to a certain extent. For example, one question was whether the development activities follow a clearly defined specification. Developers who were involved in these issues said to their managers:

> "What is meant by that and that? How should I accomplish this?"

The same developers would have said to assessors:

> "This is clear to me, Sir!"

This unbalanced answer seems odd, but most of them would avoid any hassle with 'bureaucrats like these quality people'. In many cases, the assessors tended to investigate the processes of developers superficially. Official ISO 9001 auditors commented in an interview:

> "Usually, we want to gather knowledge about the system processes over all layers of the entire organisation, for what you need to be assessed. It is not as important to understand the processes, but to identify these processes and if they fit into the environment of the assessed organisation. It is necessary to have assessed processes, because it is helpful to establish ordered management activities in all areas and all layers of an organisation."

A summary of one of the inspection assessments explained some of the assessment criteria: EC enacted an efficient and effective management system that accomplished the requirements of ISO 9001 and it is respected by EC's employees. The top-level management assist the progress of the management system. The following criteria were considered for evaluations of the management system: complaint and market analysis; strategy meetings and regular top-level manager meetings; internal assessments; and customer satisfaction.

4.3.4.2.3 Start of testing

As explained in section 4.3.4.1.2, the start of sophisticated tests preluded an alteration in the thoughts of EC's managers about the value of the tests as part of the development process. New employees kept increasing the development department and the number of different tasks. This difference of hardware development, software development, and test system development set off a new organisational structure with different departments and distinct responsibilities. The introduction of separated departments assisted the compliance with EC's standards to a large extent, because the department managers put more value on prescribed processes. Concurrently, the project size of *Theta* increased and the complexity of project *Theta* became barely manageable (section 4.3.3).

In conclusion, through various processes EC started to adapt to a more comprehensive and structured set of procedures. The leading engineer of the test team commented:

> "The start was difficult, because we demanded some of the responsibilities of other departments, such as the hardware and software departments. Although we are still in discussions about various issues, we established some solutions to a number of problems. Our test equipment is designed and programmed to run some rudimentary performance tests. Other tests validate the soldered hardware connections. Still, some of the pure software tests, such as performance tests between several chips are the responsibility of the software department."

Martin stated:

> "The test team was absolutely necessary for EC's development practices. Thanks to that team, we were able to detect and analyse some of the malfunctions of *Theta*. In addition, one test analyses *Theta* devices before they are delivered to the customers. We had already collected some bad devices and started to investigate the problems, why these devices were faulty. I think the test team brought EC a major step forward, away from the craft corner toward a more sophisticated engineering company."

4.3.4.2.4 Other non-formal processes

Non-formal standards remained in its sphere of action relatively conformist. Non-formal standards are structured processes, which were not part of the written set of EC's standards. A vital non-formal standard was the weekly meeting called 'change-control-board' for the purpose of discussing and deciding changes during the development of project *Theta*. Martin introduced that meeting and he invited representative stakeholders of EC from the marketing, project management, test system development, software development and hardware development departments. Martin commented that the growing complexity of project *Theta* called for more structural efforts in order to manage this project well. After several 'change-control-board' meetings it became a standard procedure at EC. Raphael commented that the 'change-control-board' enabled team members to better understand the development processes of EC and said:

> "The participants of the change-control-board meeting document the discussions of the weekly meeting and send it to a database where the developers have access. I like to read these reports, because it keeps me updated about various decisions and processes inside EC."

In particular, Martin claimed that the 'change-control-board' was essential for the success of project *Theta*. Martin stated:

> "I called in the change-control-board, because I needed to decide several issues and I wanted to make these decisions in consensus. Therefore, I called somebody from the marketing department, Gabriel, one hardware

> developer and the other project managers in order to discuss these issues. For some reason we needed to postpone some decisions to the next and so it became an ordinary occurrence. […] Based on its success, I would have established this meeting even earlier at EC."

In addition, other non-formal processes were important for the success of *Theta* and remained untouched throughout my observation period. As mentioned in section 4.3.4.1.4, Bugzilla and Subversion have their own procedures on how to handle software errors and software versions. Both tools were open source, but remained unchanged by EC in its core functions.

4.3.5 Raising conflicts

The previously explained circumstances of project *Theta* involved the potential for conflicts and tensions between various stakeholders, such as software developers, hardware developers, test equipment developers, the project manager, marketing, and quality manager. The increased complexity of project *Theta* confronted the people from EC with new challenges as described in section 4.3.3. Certainly, lack of conformity (section 4.3.4.1) between company standards and practices created some tension between the quality manager and those who did not comply. On the other side, nobody would have described the context of project *Theta* as chaotic, because the fundamentals of embedded system development and project management remained stable (section 4.3.4.2). However, I observed tensions within EC in several areas. First, the tremendously increasing size of project *Theta* impacted on EC. Second, the installation of a sophisticated test team affected the actions and practices of EC's development and project management. Third, assessments influenced the awareness about that process and product quality. Finally, managing customers with their technical as well as non-technical issues was not coordinated enough.

The increasing size of project *Theta* disclosed gaps in the organisation of EC in addition to the tension between different departments. As section 4.3.4.1.1 explained, the requirement elicitation and management became more difficult with the increasing size of the project. Gabriel stated that there were two reasons for this: first, the decision paths lengthened, because several developers needed to be consulted as to the potential effects of additional or fluctuating requirements; and second, different customers expected a different scope of the project, hence different requirements. Gabriel stated:

> "The different perspectives of the requirements were a major problem during project *Theta*. On one side I did not emphasise enough the advantages of reusing previous development parts. On the other side, the customer was ambiguous about their own requirements, because the knowledge just was not available."

The installation of a sophisticated test team involved changes in EC's development and project management activities. This not only had an influence on the increasing size of project *Theta*, but it also created pressure between the new test team and the developers. Moreover, the lack of sufficient testing (section 4.3.4.1.2) explained most of the first device returns as one of the developers stated. The developer commented further:

> "The quality of the first dozen devices could have been better, if the product had undergone some serious testing. However, we have not had the human resources to do so."

The distribution of responsibilities was a topic of great discussion within EC at the time of the installation of the test team. On the one hand, the developers avoided giving all test tasks to the new test team, and on the other hand, the test team required a clear foundation for their functions.

Furthermore, the accepted behaviour of writing less documentation within the group of developers raised some tension directed towards David (quality manager). Particularly with the increasing importance of project *Theta*, David was worried about the upcoming ISO 9001 assessments by external examiners. David commented that creating the awareness with the developers about the importance of documentation is difficult. David commented:

> "Figures, data and facts – that's what is important for me. Unfortunately, some of the developers have a different understanding and do not help my job at EC. I know that *Theta* might expose itself as a disaster, because of all the quality processes we have skipped."

Another issue that created some tension within EC was an internal ISO 15504 assessment. The standard ISO 15504 is also known as SPICE (Software Process Improvement and Capability Determination); it is a framework for the assessment of processes of software-producing organisations. It created some worries, because this internal assessment focused on the processes of software developers by only applying the SPICE engineering processes as a benchmark. The result of this assessment was staggering for the software developers, as a result of a number of incomplete processes within the software development department (cf. section 9.5). Although the structured processes of EC would have provided the basis for a better result, even their standards lacked some issues vital for the SPICE framework, such as performing reviews at any stage of development. David stated:

> "We hoped to highlight some of the problems of the software development department. Although I expected a poor result like this the difference between ISO 15504 and ISO 9001 is more than I expected. However, we need to understand that ISO 9001 comes from a more generic and holistic perspective to assess an organisation."

Section 9.5 (appendix) provides a detailed comparison of that SPICE assessment (Table 9-4 and Table 9-5).

In addition, several developers reported that with the devices delivered first, some of the users kept asking the developers directly in order to get help with various issues. During that time with the customers, the developers were not able to keep track of their usual tasks. This disclosed a lack of a coordinated position to support the customer. Although the developers said that in previous projects it was part of their job, the queries became too much for the developers. A software developer commented:

> "After a while it was getting annoyed about that circumstance, the constant phone chats about *Theta* with some customers. I was glad that EC engaged somebody for this task, to provide customer support."

4.3.6 Finding solutions

One unexpected development of project *Theta* was the increasing number of customers as well as an increasing demand for that device. Most of these customers had a different scope of tasks; therefore, the new customers valued different features more. These requests and requirements needed to be handled, and the other customers wanted to be satisfied as well. The 'just another project' of EC over time became *the* project of the development department, and therefore the top-level management changed their opinion about the value of the project. The company vision of the full-service supplier altered and EC's idea that they would deliver products directly to the assembly line to the automobile manufacturers so that the end customer may experience an EC product expired. The new company vision was placed around product *Theta*, and it had a strong market success; its market share became almost 70% in that niche in 2006 and 2007. Product *Theta* became a significant tool for manufacturers involved in research and development activities for automobile electronics. Therefore, the idea of the full-service supplier changed to EC offering service suppliers the tools for their tasks within automobile research and development units. The product manager (marketing) commented:

> "The first success was based on mouth-to-mouth advertising. After the management realised that we might have even more success with a coordinated marketing concept, I was assigned to that new task. I convinced Michael and Mariah to perceive *Theta* also as a supportive argument to place our service suppliers at the customer. Indeed, *Theta*

operated as a door opener for many service supplier projects at different automobile manufacturers."

In addition, Michael commented:

> "We used *Theta* as a door opener for a couple of new service supplier projects, although this was not the original purpose of project *Theta*. Over time we coped with the increasing complexity of project *Theta* relatively well."

In a sense, the different scopes of customers influenced the development efforts to design the *Theta* system flexibly. Since the beginning of requirement elicitation, the software needed to reflect a certain amount of modular design. As a result, the embedded software was designed to support any update via the client software. In addition, the client software executed some of the calculations, in order to reduce the efforts and hardware resources of the embedded system. The hardware was designed to reflect a kit of precisely defined components, since its origin started as a simple 32-bit platform. These fundamentals enhanced the required flexibility of developers and the project manager during project *Theta* to meet the demands of the customers. Gabriel stated:

> "We were able to create a relatively flexible system, although we did not always know the consequences of our doings. I guess our trial and error efforts were successful, but I would not recommend it a second time. Maybe a more structured approach would have done well for project *Theta*, because its complexity and size was in the end bigger than expected."

In mid-2005, after the first dozen products were delivered to the initial customer, new customers ordered slightly different *Theta*s. At the beginning, EC wanted to limit the requirements, but the increase in different expectations torpedoed this strategy and EC needed to find a solution to satisfy the new demand for *Theta*s. That solution provided a newly defined baseline to guarantee a minimum of agreed features to the customer. Beyond that baseline, the customer needed to agree to set up a customised solution in a sub-project of *Theta*.

In addition, new issues started to be discussed with the relevant stakeholders of EC in the change-control-board meeting (as section 4.3.4.2.4 explained). That regular meeting, which had first been called on a whim, was very helpful as many developers, as well as Martin, stated. Gabriel commented:

> "I was happy about the change-control-board, because it helped us a lot to move forward some of the issues which needed to be decided. That it became a regular meeting shows us that it was useful and necessary. I almost cannot remember how it was before that meeting!"

Moreover, the start of testing involved the enlargement of the team who were stakeholders of project *Theta*. This new test team partly took over some responsibilities of the development team and established some new regression tests of product *Theta*. EC tried with these concentrated efforts to target blatant quality problems. In addition, EC investigated some of the technical bottlenecks of different hardware versions as well as software versions in order to increase the knowledge about performance. Some people in that new team were previously engaged with finished projects of EC, but several people were hired to cover the various tasks of that new test team. For instance, EC hired a new person for managing customers and their problems. Technical issues were collected and processed in a customer error database, to plot customer issues. Colleagues from finished projects were integrated into other responsibilities. For example, some employees acquired project management tasks of these new sub-projects of *Theta*. This strategy of various sub-projects had a positive impact on the organising of the project management efforts of EC. Martin commented:

> "Divide and conquer! I was very happy about the distribution of project management tasks. The increasing complexity and size of project *Theta*

> required an expansion of the people responsible for the project management. This meets the company targets, to create highly customised devices to meet the expectations of our customers."

One employee investigated the procedural side and looked for improvements, without being influenced by the quality manager.

> "Michael asked to do an anonymous survey about the experiences with project *Theta*. The survey was conducted through all developers. I analysed and subsumed the answers and gave it back to Michael. I think, although I am not sure, that the consequence was the initiation of process improvement efforts."

The anonymous survey of internal *Theta* stakeholders was initiated by Michael and Mariah. The reason for doing this was to evaluate and gather ideas about the problems and their possible solutions within EC. The survey may have led to temporary improvement efforts of the product development process. However, Gabriel, Martin, and many developers commented that this was not beneficial for EC's development process.

The SPICE assessment highlighted some issues at the software developers and it encouraged the efforts of the testing team. In addition, Gabriel demanded more documentation from his employees. Beside the needed improvement in documentation and tests, the SPICE assessment highlighted another issue. It was suggested that reviews were conducted on a regular basis throughout the various development steps. However, I was not able to observe any indication of software reviews during the period of this case study.

Another solution to the increasing tension within EC was to integrate EC's marketing department better with project *Theta*. In particular, non-technical issues are part of discussions within the weekly change-control-board meeting between people from marketing and *Theta*'s project manager.

4.3.7 Summary of project *Theta*

Having experienced the wider context of EC, it may be said that project *Theta* was a far-reaching change for the entire company. At the beginning, stakeholders of EC thought of it as just another project (section 4.3.1). However, that opinion changed during the development phase. Technical challenges involved coping with ambiguous and fluctuating requirements. The necessary flexibility of hardware and software solutions required a careful consideration of decisions (section 4.3.4.2.4). However, the increasing number of customers and developers also demanded other solutions. *Theta* was the first project of EC to involve the satisfaction of miscellaneous customers from research and development departments of automobile manufacturers (section 4.3.6). The proportion of involved developers was more than in previous development projects of EC (section 4.3.6). Some actions and practices of developers changed over time. A more sophisticated procedural structure of internal processes involved the development team and the project manager in their activities (section 4.3.4). A new test team became an influential group within the internal stakeholders of *Theta* (section 4.3.3.5). Recruiting another member of customer support staff released the developers and professionalised the appearance toward customers (section 4.3.6). Hardware and software were rigged up to develop *Theta* (section 4.3.2). Previous experiences in the form of code and design layouts were reused and thrown together with the purpose of fitting in project *Theta* (section 4.3.1). Besides experiences from other projects, trial and error was part of the activities of EC's developers (section 4.3.6). Despite this vital ingredient of unscripted work, it became less during project *Theta*. Gabriel stated:

> "At the beginning we just wanted to generate output. So at many corners we tinkered. With the increasing size of *Theta*, we saw the necessity of tinkering less and increasingly did it the proper way. Although we tried to do so, the software <embedded part> was getting more and more "ver-bastelt" (further tinkered with) over time."

Many developers stated that their approach to small problems became more formal, although it still involved some hacking and patching. One developer commented:

> "Some problems required quick solutions. Although we looked for proper solutions, the time pressure only allowed hacking something together quickly, in order to get it done. Sometimes, when we were not in such a hurry, we investigated some of the problems we had solved with these hacks and found a better solution."

The group of developers and the project manager experienced that time as particularly goal oriented. The developers were ambitious about *Theta* and subordinated other activities under that target. In addition, the developers mentioned deliberateness about their actions in a way to rethink the immediate response of their modifications in code and design layout. One developer commented:

> "Although we hacked, it does not mean doing something and hope for the best. Certainly, time played an important role for many solutions. In most cases we had a rough idea about the things we had altered in the code or layout design. However, some tinkering was also involved in some cases."

In addition, changes or reinterpretations of the requirements were discussed in the change-control-board. Developers were involved with that regular meeting not only as passive participants (reading the meeting report), but their expertise was valuable for the decisions made. Martin stated:

> "For some issues we consulted the developers in order to get a better grasp. For some details I think it is helpful to ask them about their opinion and estimation, although I got a pretty good understanding over time."

Although some of the activities of project *Theta* might seem chaotic, the wider picture disclosed similar generic steps of embedded development. The distributed development incorporated two branches. Hardware development included designing several layouts of boards for project *Theta*. In cycles, the development strived for an improved version in every iteration loop. Software design was sub-divided into embedded software and client software. Embedded software took into account the various pieces of different software for different processes onboard *Theta*. Client software comprehended the user interface with a calibration tool and a download tool to transfer the recorded data onto the user's computer. Each software development occurred in loops with some merging points for a new release of the software package. In addition, the different hardware versions required distinctive software releases to work in an intended way. In conclusion, embedded system development of EC was developed in different loops with merging points for hardware and software. Despite the development becoming more structured, this issue persisted during the period of observation. In addition, EC continued to be ISO 9001-certified, although a number of the company's structured processes were barely followed. However, the interview with external ISO 9001 assessors may indicate that the ISO 9001 certification does not much reflect the actions and practices of the developers. For an ISO 9001 assessment it is more important that the fundamentals are established and, nevertheless, the ISO 9001 covers the entire organisation. Furthermore, non-formal processes dissemble another source of continuity within EC at the time of observation. The change-control-board's, extensive use of development tools such as Bugzilla and Subversion offered a worthwhile standard during project *Theta*.

The prototype and final mainboards of product *Theta* are depicted in Figure 9-5 and Figure 9-6.

4.4 Customer opinions about project Theta

Theta facilitated the research and development of automobile manufacturers to a large extent. I gathered generally positive acknowledgements in interviews with some of *Theta*'s users. Previous test configurations with various tools have not provided time-coordinated records of data or were not compliant with automobile standards. *Theta* provided a solution to these lacks and, in addition, it provided its client a user interface with high usability. Many customers mentioned it seemed like a really sophisticated product and they were curious about the professionals involved with that project at EC. Many customers I have interviewed have not experienced the change when the customer service moved from the developers of *Theta* toward an employee with the task of supporting customers. Here are some quotes from *Theta* users:

> "*Theta* is a small device, which helps a lot in my work. [...] I like the easy interface <client> most, because that's the way tools should be. You are always just a few mouse clicks away from almost all necessary configurations. Other tools do not provide that sort of genius."

On the other hand, some users also had some criticisms. In general, the researchers and developers in the automobile industries were worried about the possible negative influences of any test equipment during their tests. *Theta* was not a certified MOST product and raised some concerns about possible side effects during test configurations. In addition, one user stated that there was no feedback provided about the ideas he mentioned for improving *Theta*. However, it is interesting to note that the interviewed customers noticed nothing about the complications and worries during the development of *Theta*.

4.5 Follow-up to Theta

An appealing feature of this case study was the possibility to observe the follow-up to project *Theta*. Previous sections reported a lively influence on the actions and practices of EC and their customers. As noted earlier, the success of *Theta* provoked EC to alternate their full service supplier philosophy. With *Theta*, EC began to concentrate on the marketing of *Theta* as a necessary tool for their service suppliers to be involved with research and development departments of automobile manufacturers. The appearance of EC should provide the image of a company able to provide service suppliers and customised devices for their success. During the observed follow-up period of *Theta*, the predevelopment phase of successor projects built on that philosophy. However, the successor project was connected with a more advanced version of *Theta*.

This advancement integrated newer hardware, such as the 32-bit processor and the digital signal processor. In addition, the technical standards of the connected bus systems were changed and EC needed to adapt to that new situation. The success of *Theta* encouraged EC's competitors to improve their efforts in that market niche. Because of these efforts, competition increased and the customers were able to choose from several products that covered a similar scope of services. Follow-up projects needed to address these circumstances with contemporary processors and bus interfaces. In addition, market forces demanded a deliberate approach to meet the expectations of potential customers. The number of potential customers (different departments of various automobile manufacturers in the world) increased with the spread of the MOST standard and this situation influenced technical and marketing decisions about successor projects to *Theta*.

The far-reaching changes with *Theta* involved an increase of EC's pool of experiences. Technical capabilities and procedural requirements were lifted to a stage beyond a simple shop. Regular assessments of processes within the entire organisation according to the ISO 9001 standard continued EC's policy of continuous improvement (Kaizen). In addition, non-formal company standards, such as the use of Bugzilla, Subversion, change-control-board, round-off the picture of organisational process. Furthermore, the structural setting involves extensive testing and experienced customer support. Follow-up projects would benefit from the experiences gathered from project *Theta*.

5 Analysis

5.1 Introduction

The previous chapter presents the case study of an embedded system development process. This comprises descriptions of the wider organisational context, as well as a detailed narrative story on one project called *Theta*. The data provides in-depth insights into EC's embedded system development process, its challenges, complexity, standards, conflicts and realised solutions. In this chapter, I analyse the data, structured around three analytical topics.

5.2 Analytical topics

The three analytical topics that emerged from the case were:

a) structured processes (such as the process management framework of ISO 9001);
b) practice of improvisation and bricolage activities; and
c) reflexivity and human agency.

The analysis is suffused with efforts to maintain connections between the analytical topics and project details, such as social context, individual differences and technical circumstances. These project details had the potential to influence the practice of prescribed processes, stimulate the emergence of new standards and provoke coping actions. The ISO 9001-certified organisation was examined for processes that were practiced by EC's employees (sections

5.3.1 and 5.3.2). The investigation revealed processes that emerged over time and became valuable standards at EC (section 5.3.3). Although the first topic provides a useful angle to analyse the whole bandwidth of structured processes, it involved several shortcomings. Activities that do not fall within that realm cannot be analysed from the angle of structured processes. As a result, another topic was used to illuminate the case study from an opposite angle. The second topic of improvisation and bricolage involved the capacity to illuminate the field data from that particular angle. For example, procedures that were difficult to describe, control and manage, or were situated in emergent contexts were exposed better by using the second topic. Consequently, the strengths of the second topic overcame the weaknesses of the first topic. The application of the second topic highlighted different patterns of improvisation and bricolage activities. Those patterns involved various stages during the development process, such as the prototype development (breakthrough – section 5.4.1), initial phase of 'series' development (innovative practices – section 5.4.2) and advanced phase of series development (adaptive practices – section 5.4.3). In addition, the social context of an embedded system development was analysed from an improvisational and bricolage perspective (section 5.4.4). In conclusion, the first and second topics provided help to analyse actions and practices of embedded system developers and managers. With those two topics, the analysis offered a rich description of the processes at EC and how these processes were shaped. However, the analysis was lacking in descriptions of the human actions and the motives of individuals, regarding why those processes occurred during project *Theta*. The concepts of reflexivity and human agency helped to explain the human actions and the motives of individuals and thus make those vital aspects of social life understandable. Consequently, this third topic was applied to structured approaches (section 5.5.1) and improvisational and bricolage activities (section 5.5.2).

Briefly summarised, the analysis is structured as follows: Firstly, I analyse the structured processes of the case data; secondly, I disclose how improvisation and bricolage were involved in the activities of developers and managers; thirdly, I examine the actions and practices of research participants with regard to the reflexivity and human agency; and, finally, I reveal interacting influences of structured processes and improvisation and bricolage. I present a short summary of the findings at the end of the analysis chapter.

5.3 Structured processes

The case illustrated that structured processes had significant influence on the embedded system development process. Those structured processes involved formal and non-formal procedures as valuable practices of EC. The detailed analysis of the structured processes begins in section 5.3.1 with a focus on the influences of ISO 9001, i.e., those pertaining to development practices. Section 5.3.2 examines the influence of non-formal standards, which were not documented as organisational processes, but were regularly practiced and endorsed by participants. Section 5.3.3 analyses emergent patterns in the identified structured processes from the case study site. Finally, section 5.3.4 investigates the consequences of those emergent patterns of structured processes.

5.3.1 Influence of ISO 9001

The ISO 9001 certification for EC had different effects on the organisation. First, I describe the motivation of EC to acquire ISO 9001 certification. Second, the positive effects of the established structured process framework of ISO 9001 are exposed, and, third, I explain the constraints of ISO and the corresponding response inside EC. Finally, I compare the ISO 9001 with the SPICE framework and assess the engineering activities of EC according to SPICE.

The case data revealed a complex picture of the motivation for EC to receive the ISO certification. External factors were dominating, because this certificate provided a competitive advantage for offering the products and services of EC on the market. The ISO certificate gave potential customers the knowledge about a certain stable quality of the products and services offered by EC. Furthermore, the potential customer expected that EC is an organisation that gives serious consideration to quality issues. In addition, the data disclosed factors regarding how EC wanted to benefit in its internal procedures from an ISO 9001 certification. For example, the certification provided help for establishing and maintaining structured procedures; ISO offered guidelines for EC employees concerning what, when and how to do their work; and ISO 9001 certification provided the managers of EC a certain level of control over the given processes within EC. Table 5-1 distinguishes the different motivating factors for EC to acquire ISO 9001 certification.

Table 5-1 Motivation for EC to receive ISO 9001 certification

External motivations for ISO 9001 certification	**Internal motivations for ISO 9001 certification**
Competitive advantage in marketing	Provided help for establishing and maintaining structured processes
Customer assumption of stable quality of products and services	Offered guidelines for EC employees concerning their work
Customer assumption of serious considerations concerning quality issues	Possibility for EC's managers to obtain better control over the processes within EC

It is important to note that the attitudes concerning ISO 9001 were diverse within EC. These could be divided into two groups. Higher in the company hierarchy, the opinion was strong that internal motivation was the major reason why EC introduced ISO 9001. Nevertheless, those reasons to obtain ISO 9001 resulted in positive effects for EC. Lower in the company hierarchy, the prevalent opinion was that external motives resulted in useful outcomes in the relationship between EC and its customers. As data from the appearance of EC (e.g., internet homepage, flyers) showed, EC advertised with the ISO 9001 certificate. Therefore, customers were aware of this circumstance, because EC put emphasis on certified processes. Consequently, customers assumed that EC was able to maintain a consistent quality in their products and services, and that EC gave serious thought to quality issues. Moreover, the internal motivations caused positive effects inside the organisation of EC. Previous experiences with development projects and supplying service established several processes. With the efforts to receive ISO 9001 certification, those processes were honed and better structured, so that EC's managers were able to manage better. Further positive aspects were new guidelines for EC's employees and a universal framework that helped to establish and maintain processes within EC.

Apart from those positive consequences for EC, the ISO 9001 certification also gave rise to some constraints and corresponding responses inside EC. In analysing those responses, a pattern of intended and ingenuous (unplanned) activities became identifiable. Intended responses were, for example, the organisational ISO 9001 standard prescribes the use of a software revision control database called "Concurrent Versions Systems" (CVS), but a successor software of CVS was better, and therefore the software developers used the successor software called "Subversion"; project managers of development projects were previous developers who had gathered sophisticated technological knowledge, except for one (Martin), who, as a previous service supplier, became project manager in order to have an intensive interaction with the customer; preventive measures and corrective actions were described vaguely, but then another catchphrase was introduced called "Kaizen". Those constraints and their responses by EC's employees were certainly conscious and purposeful at the time of the appearance of the ISO 9001 certification. Another pattern of ingenuous responses to constraints of ISO 9001 certification became evident. These sorts of responses were hardly planned and had more of an emergent character. Ingenuous responses were, for instance: ISO 9001 specified development processes, but managers did not always follow those prescriptions; project task procedures were constrained through ISO 9001, however, developers hacked; the ISO 9001 gave support for specified task assignments for project members, whereas Raphael behaved more like a ’fire fighter’. Table 5-2 lists the constraints of ISO 9001, its corresponding responses inside EC, and the identified patterns of activities.

Table 5-2 Constraints and the response inside EC (Molnar & Nandhakumar, 2007)

Type of constraints	**Example of response to constraints**	**Type of identified pattern**
Use of Concurrent Versions System (CVS)	Successor of CVS is better and therefore they use Subversion	**Intended activities**
Project manager of the project *Theta* was a developer, with deep technological knowledge	For this project the project manager was a service supplier to gain best possible interaction with customer	
Weak description of preventive measures and corrective actions	Introduction of the buzzword Kaizen to increase awareness	
ISO 9001 processes of product development management	Take shortcuts, iterations and zigzag routes to stay on schedule	**Ingenuous (unplanned) activities**
Work processes of product development	“Quick and dirty” solutions, hacking and patching involves the product development	
Specified responsibilities during project	Fire fighter Raphael	

Although the main share of success had its basis upon the influence of structured processes, developers and managers could not always cope with the constraints of structured processes. Some constraints were built into weak descriptions of processes and other processes were stipulated to be rigid. The analysis identified two different patterns of responses to the various constraints, i.e. intended and ingenuous responses inside EC which led participants to a higher level of performance, because, through those responses, EC's employees were able to overcome the constraints of ISO 9001.

As previously stated, the software developers' actions and practices do not mirror the organisational ISO 9001 processes. Only some fundamental procedures have been part of the daily life of the software developers. To analyse the features of the observed actions and practices of the software developers, the engineering processes of the SPICE framework were a great help. The case study stresses the point that the system architectural design and software design processes were underdeveloped. Moreover, several base practices of the engineering processes were not described in the processes of organisational ISO 9001. This analysis of the pure documents indicated that the EC would achieve the SPICE maturity level 0 (incomplete process) (cf. Table 5-3 and Table 9-4).

Table 5-3 Comparison of organisational ISO 9001 standard with SPICE specification		
SPICE specification		**Compliance of EC's standards to SPICE Level 1**
ENG.1	Requirements elicitation	Fully achieved
ENG.2	System requirements analysis	Partly achieved
ENG.3	System architectural design	Not achieved
ENG.4	Software requirements analysis	Partly achieved
ENG.5	Software design	Not achieved
ENG.6	Software construction	Largely achieved
ENG.7	Software integration	Largely achieved
ENG.8	Software testing	Partly achieved
ENG.9	System integration	Largely achieved
ENG.10	System testing	Largely achieved
MAN.3	Project management	Largely achieved

In addition to this comparison of ISO 9001 with the engineering processes of the SPICE framework, I assessed the engineering activities of managers and developers according to SPICE (Table 5-4 and Table 9-5). Besides the lack of conformity between organisational ISO 9001 processes and the SPICE framework, EC's employees did not follow the specifications of ISO 9001 either. Developers and managers showed a lack of documenting and conducting reviews in every stage of the development process. Furthermore, testing efforts were insufficient and hardly overseen. One developer expressed these actions and practices as unorthodox, but in general it was difficult for all of EC's employees to explain their actions and practices discursively. For instance, the word "tinker" is afflicted with negative attributes in the professional work of developers. Therefore, none of the developers claimed to be a tinkerer. On the other hand, the developers had difficulties with the compliance with the organisational ISO 9001 processes. Nobody claimed to follow the structured processes either. Many developers and the development manager described their actions and practices in the middle of this continuum, where following structured processes is on one end and tinkering is on the other end.

Table 5-4 Comparison of EC's practices with SPICE level 1

SPICE specification		**Compliance with EC's practices**	
		Comparison Project 1	**Comparison Project 2**
ENG.1	Requirements elicitation	Fully achieved	Fully achieved
ENG.2	System requirements analysis	Partly achieved	Partly achieved
ENG.3	System architectural design	Partly achieved	Partly achieved
ENG.4	Software requirements analysis	Not achieved	Not achieved
ENG.5	Software design	Not achieved	Not achieved
ENG.6	Software construction	Not achieved	Not achieved
ENG.7	Software integration	Not achieved	Not achieved
ENG.8	Software testing	Not achieved	Not achieved
ENG.9	System integration	Not achieved	Not achieved
ENG.10	System testing	Not achieved	Not achieved

It should be pointed out that ISO 9001 was an important factor in the activities of EC. Some development efforts were sustainably influenced by ISO 9001 standards. That influence was largely desired by the management and customers of EC. A detailed comparison of ISO 9001 with the SPICE framework highlighted several gaps in the engineering procedures of the organisational ISO 9001 standard. Moreover, EC's employees did not always follow the specifications of ISO 9001. After all, obtaining that certification implied a number of positive effects, but also gave rise to constraints and in turn corresponding responses inside EC. Adhering to ISO 9001 involved a form of dependence on its prescribed procedures and, unquestionably, flexibility was constrained to a degree that affected responses.

5.3.2 Influence of non-formal standards

Besides ISO 9001, non-formal standards were influential in the day-to-day activities of EC's employees. Although it is natural that managers and developers follow their individual routines, this analysis concentrated on the non-formal structures or standards that were identifiable as a pattern on an organisational level. Namely, I determined five non-formal standards that were practiced or recognised by many research participants. First, I describe the idea of Kaizen and its influence on the organisation; second, the vision of EC and some of its implications; third, practiced organisational values by EC; fourth, core ideas of how to write software; fifth, use of specific software tools to organise larger projects better; and, sixth and finally, development upon different technical platforms.

The case study disclosed that EC highly regarded the idea of Kaizen. As in the already briefly explained case study, Kaizen is a Japanese philosophy that focuses on continuous improvement. Again, particular people in the higher hierarchy levels were attuned to recounting their experiences with Kaizen. Research participants in the lower hierarchy levels stated mixed feeling concerning Kaizen. It is interesting to note that David (quality manager), concerning the improvement efforts of an organisation, mentioned that Kaizen was in strong contrast to ISO 9001. David maintained that a fulfilment of ISO 9001 processes would not need other ideas such as Kaizen. However, the management of EC were very positive about the idea of Kaizen. Therefore, new employees were familiarised with that aspect of the organisation culture; the idea was introduced on flyers, and, in larger organisational meetings with the board of managers, Kaizen became a frequent part of presentations. Consequently, EC established an interesting aspect of the organisation culture, and Kaizen was very important for EC, particularly for the higher hierarchy levels. Interestingly, though, there was a consensus in the lower hierarchy levels that Kaizen was not as important in their daily activities at EC.

Developers gave preference to more solid tools and techniques that helped in their day-to-day activities.

The leaders of EC described another non-formal principle as the vision of the organisation. In contrast to the acceptance of ISO 9001 and Kaizen, the vision was explained by means of comprehensible wall charts, using the analogy of a cathedral and its stages when it is being built. Just like a building, where people can observe almost daily progress, EC grew at the time of this study tremendously. The organisation underwent enormous growth and the employees noticed that circumstance. Therefore, the case revealed a coherent pattern, and the employees as well as the managers were able to describe this concept and many stated their experiences, regarding when they had started to work for EC. Insofar as this is a simple pattern, EC's employees identified their work with working on something larger. For that principle the board of managers chose a clear wall chart to explain to every employee that they are part of something bigger. Therefore, employees were encouraged not only to be aware of their current sphere of action, but that of their colleagues as well.

The employees with personnel responsibility at EC practiced several values, such as self-organisation, continual learning and involvement. As the case study disclosed, all employees were trained during their probation in those values that were important for the organisational leaders. Although no one of the research participants referred to those seminars, they were part of the training and the people with personnel responsibility wanted every employee to participate. Because new employees were sent routinely to those seminars, they became part of the organisational standards. Also, in organisation-wide presentations of the board of managers, the importance of those seminars was emphasised. Moreover, they encouraged employees to facilitate the gathered knowledge in their day-to-day activities. Therefore, EC was able to fall back on human resources, with a similar understanding regarding certain values. A common understanding in important values for business life, such as self-organisation, continual learning and involvement, seem to make it easier for managers to shift human resources and to remain flexible.

In addition to those issues that were standard at the case study site, solid software development techniques were also made consistent. In order to ensure compatibility within code that was produced by EC's developers, EC's software developers used modular design and regular style code. Not surprisingly, that kind of conduct is common in software-producing organisations and more or less state-of-the-art design of software. What is interesting, though, is that the development manager and project manager emphasised that issue. For example, Gabriel (software development manager) mentioned that although EC has put some effort in ensuring compatibility with code fragments for embedded software, he had hoped that the developers would have been able to reuse more code from previous products. However, different hardware configurations of embedded systems required different software. Consequently, this inability to reuse large amounts of code that had been written previously was a fundamental challenge in embedded system development. That was also the reason for the underlying strategy of EC to develop their embedded systems upon different platforms. These platforms would provide room to specify the later products and reuse a significant amount of code. Therefore, the non-formal standard of using state of the art design had some practical implications to streamline development efforts.

EC chose to use distinctive software tools to organise their projects better. Several software tools were used, such as Subversion and Bugzilla. Both tools were designed to support software development efforts and were widely recognised in this industry. Subversion was used to facilitate collaborative software development by controlling the different software versions throughout a software development project. Bugzilla was used to track bugs of development projects, such as *Theta*. Both specialised software tools were vital for the development of *Theta*, as well as other projects at EC. Working with those tools was compulsory and, therefore, it could be said that EC used Subversion and Bugzilla as standard tools. Using those tools was certainly not revolutionary, but it helped the developers to cope with problems during the development of *Theta* by having sophisticated tools to control software versions and to track bugs.

Finally, the managers of EC emphasised the strategic decision, to develop upon different platforms. The steady growth of EC involved an increasing capability to master the technical challenges of developing 16-bit processor and 32-bit processor platforms. The managers of EC decided to use their available platforms in order to develop new products around them. Therefore, the platform architecture involved many possibilities to connect new components easily. Consequently, developers and managers knew the technical possibilities as well as difficulties, when new components were attached to an existing platform. Project *Theta* was developed upon the available 32-bit processor platform of EC.

In summary, non-formal standards were very important for EC. Non-formal standards involved not only neat ideas like Kaizen, platform development, the vision of EC and practiced values, but solid software development techniques and tools were also used at the case study site. Whereas some non-formal standards are so common for EC's employees that they find them hard to describe, software specialists were willing to describe their development techniques and tools. However, this mirrors the down-to-earth attitude of many employees of EC. In addition, non-formal standards helped managers to adhere to some main ideas, and helped developers to work according to some solid principles at EC. The analysis disclosed that employees at lower hierarchy levels as well as top-level managers initiated non-formal processes at EC. Specific tools and practices that were used as part of the developers day-to-day activities were approved by the top-level managers, but the developers initiated those helpful tools and practices. Top-level managers established non-formal processes that were more strategically oriented (Table 5-5).

Table 5-5 Overview of the non-formal processes initiated at EC

Employees at lower hierarchy	**Top-level managers**
Development tools: Subversion, Bugzilla	Kaizen
State of the art design of software	Metaphor of building a cathedral
	Appreciation of values for business life
	Develop upon different platforms

5.3.3 Emergent patterns in structured processes

The analysis of this case study revealed emergent patterns of structured processes. Although some formal and non-formal managed processes remained stable over the period of the project, the detailed study of obtained data disclosed change of the structured processes. ISO 9001 processes were better valued over a period of project *Theta*; hence, over time, those processes became more valuable. Likewise, new non-formal standards emerged and became advantageous means to cope with a variety of challenges that occurred at that time.

Data that were collected at the initial stage of project *Theta* indicated that developers and middle managers were critical concerning ISO 9001. For instance, the project manager stated that practical solutions might lead to better results than a sophisticated framework of process management; the development manager mentioned similar considerations; particularly condemning were developers, who said “common sense and dedication” would be better, details of EC’s ISO 9001 framework were wrong, and that the initial phase of project *Theta* was not conducted in a structured manner at all. Those attitudes might be connected with the circumstances of the project work at that time. At the initial phase of project *Theta*, it was necessary to develop a proof of concept, gather requirements, and plan the course of the project. Those innovative activities were availed less often than processes that would frequently reoccur. Although each task would take place at every project, those activities are highly project oriented and the organisational ISO 9001 processes provided only help to generically describe those tasks. Detailed explanations in what, when and how to operate were not involved in ISO 9001 process descriptions and therefore they did not appear to be of much help.

Indications of developers and middle managers concerning ISO 9001 at the final stage of project *Theta* were in contrast to those at the initial stage. Although they remained critical about ISO 9001, many developers, project managers and the development manager realised the positive aspects in more closely following ISO 9001 standards. That pattern may be related with the collective experiences with project *Theta*. Project *Theta* started as just another project of EC, but it became the most important development project of EC at that time. The challenges seemed to be straightforward, but with the increasing scope of the project, the challenges expanded too. As project *Theta* became more demanding, the elicitation and management of requirements was not simple. To follow the generic described requirement engineering processes would have avoided some mistakes by EC during the task. In particular, the tasks around the testing of product *Theta* became a symbol of the appropriateness of ISO 9001 standards, because at the progressed phase of the project they regretted that they have avoided following the prescribed test guidelines of EC. The organisational ISO 9001 standard describes test efforts as follows. They have to commence development concurrently and need to continue until the project is finished. However, as developers and middle managers stated, sophisticated test efforts had not begun until they realised that customer complaints had increased a lot. Those protests forced EC to think about the tests of product *Theta* as a whole. Previously, developers had sole responsibility about their sphere of activity, but nobody was assigned to test *Theta* in its entirety. Perceiving this immediate need of action, which should have been conducted at the beginning of the project, revealed the appropriateness of ISO 9001 standards for many people, who were very critical earlier.

The analysis identified a trend of valuing the ISO 9001 standard better among the employees of EC. In particular, at the end of project *Theta*, many employees realised the value of the process framework, because working on project *Theta* involved many new experiences. In addition to that trend, new non-formal standards emerged and became important aspects to manage a range of difficulties that occurred at that time. For example, the requirement for elicitation and management was not influenced by the ISO 9001 processes, but soon induced Martin to initiate the change-control-board meeting to discuss and decide changes during the development of project *Theta*. Although the meeting was started on a whim, it became a more regular convocation of the various internal stakeholders in order to cope with the challenges as they were occurring. Therefore, the meeting became a standard meeting, and it contributed to the success of project *Theta*. In addition, new software tools were introduced by one developer on a project-wide scale at the beginning of project *Theta*. All developers deployed Subversion and Bugzilla quickly, because the development manager and project manager supported the implementation of those tools. Subversion fulfilled its purpose of managing the various software versions throughout the development process. Bugzilla was used not only for its original purpose to track and manage errors, but the tool also acquired the role of recording unexpected incidents with product *Theta*. Particularly at the beginning, the analysis of the data revealed that Bugzilla was the central tool to collect unclear phenomena of the developed device. Further driving the habit to collect unexpected incidents was a prevalent uncertainty concerning the newly developed hardware as well as software, in order to clarify those phenomena at a later stage of development. Therefore, Bugzilla was a little diverted from its intended use of tracking errors, although it fulfilled both tasks for the development of *Theta*. So, through the introduction of new tools, the emergent use of Subversion and Bugzilla indicated that non-formal standards altered over time, and the usage of those tools was adjusted in such a way that it would fit the demands of the project circumstances.

Beyond those tools, new human resources were hired, to accommodate the increasing challenges. For example, when it became apparent that the growing number of customers would require a different approach to manage their enquiries, a new employee was assigned to that task. Previously, the project team would handle enquiries and other questions concerning the product as such. Accordingly, the project manager was the first contact point for many customers. In questions concerning technical details, the project manager referred to the responsible developer. This procedure was good enough for smaller projects, with fewer customers. However, this approach was frustrating not only for the customers, but also for the project manager, who was bombarded with enquiries, and for the developers, who were more and more communicating with customers instead of developing for EC. This experience led EC to take a different approach by hiring an individual who was assigned only to manage enquires of EC's customers. Although it seemed to be a small alteration, this change had a positive impact on the developers and middle managers, because a growing factor of distraction was eliminated and customer satisfaction grew too.

In summary, the analysis of these emergent patterns revealed that structured processes and its involved perceptions changed during project *Theta*. The perception of developers and middle managers changed from being critical about ISO 9001 towards being more positive about ISO 9001. It seemed that the previous arrogance of developers using "common sense and dedication" which was going hand in hand with autonomy and flexibleness yielded some humility toward the rightness of the ISO 9001 approach. In addition, new tools were introduced, and new styles to use those tools emerged. Developers appeared to be very positive about those tools, and the abilities to use them. Finally, new employees helped to take new approaches in managing customers, when the previous approach failed. Consequently, people inside EC and their customers were pleased with this emergent arrangement.

5.3.4 Consequences of structured processes and emergent patterns

Revealing those emergent patterns highlights also the non-uniformity of structured processes during a development project. The perception of formal process management frameworks altered, non-formal standards became obsolete and new structures arose. What still remains unclear were the methods as to how EC developed its products. Although developers admitted to perceiving ISO 9001 as helpful for projects such as *Theta*, the approach to analyse the actions and practices from the focus of structured processes did not explain how EC developed product *Theta*.

According to the internal and external ISO 9001 auditors, the transparency of development processes improved over time. Since developers and middle managers started to document more extensively and work according to the processes prescribed by ISO 9001, auditors were less concerned. In addition, some processes were followed more rigidly on the end of the development process (e.g. testing) by developers and middle managers. However, the analysis of structured processes revealed also that non-formal processes were very important during the development of *Theta*. They helped EC's managers to positively influence their employees, and solid principles helped the developers in their work. In addition, the analysis also disclosed patterns of emergent structured processes. That means structured processes were not fixed, but an influencing factor in itself was liable to change.

These early results gave rise to the thought about the potential value of a proposed methodology at the case study site. Although the organisation was

ISO 9001 certified, this involved no specific methodological approach for the development of embedded systems, but the empirical data evinced attributes of various ISD methodologies. For example, the requirement elicitation approach of the organisational ISO 9001 documents was proposed as a preliminary step in the development, similar to specification-based methodologies. However, Martin's (project manager) practices involved an ongoing dialogue with the customer, so that ambiguous and unclear requirements could be clarified during the development of *Theta*, similar to an agile development methodology. The depicted development process of EC (Figure 4-2) describes an evolutionary development approach, which is based on the evolution of various prototypes. The developers and managers maintained this proposed development approach of the organisational ISO 9001 guidelines for the development of *Theta*. In addition, the more rigid following of some processes and the emergence of practices that were beneficial during the development of *Theta* indicated that an established and maintained methodology would have been helpful for the developers and managers at EC.

Those intermediate results required further analysis of the empirical data, because several weaknesses were associated with the approach to focus on structured processes for data analysis. Those weaknesses involved the inadequacy to analyse processes that were first, difficult to document and describe; second, difficult to control and manage; and, third and final, processes and properties that were emergent. Therefore, another focus realm was used to sufficiently target the lack of structured processes.

5.4 Practice of improvisation and bricolage

As the previous section on the structured processes at EC revealed, structured processes were an important part of the day-to-day activities of managers and developers at EC. However, activities that cannot be classified as structured processes need to be focused with a different realm. The theoretical lens of improvisation and bricolage was utilised to get a better understanding of the social details of the research setting. In addition, the strengths of that sensitising device reduced the weaknesses of focusing on the structured processes. Therefore, improvisation and bricolage seemed to complete this analysis of the case study (cf. Molnar & Nandhakumar, 2008b). The comprehensive analysis of the improvisation and bricolage activities begins in section 5.4.1 with a focus on the activities of developers and managers, when the prototype of project *Theta* was developed. Section 5.4.2 examines the early stage of the series development at EC, when many new practices occurred. Developers and managers tailored the present practices at an advanced stage during the development of *Theta*, and section 5.4.3 analyses those activities. Finally, section 5.4.4 describes the social context of developers and managers at EC, to get a better understanding of how those activities were situated.

5.4.1 Pattern of breakthrough

The initial prototype development of project *Theta* was challenging for the employees at EC. One feature technology (record of the MOST signal) of *Theta* was reverse engineered, which means that software was written by developers who were not involved with the original development of that technology. That feature was invented by a different organisation, and the usual way to embed this technology would have been to pay licence fees to that organisation. Proprietary technology was commonplace in business and benefits for the inventor were, for example: a) securing technological leadership; b) obtaining return of investment; and, c) supplying service for a limited customer base. Most parts of that technology were linked technically and legally with the inventor; however, a gap was identified by EC in the licence details. Therefore, EC's goal was to capitalise this gap and to develop the required technology. By retrieving the technology's details from available hardware, EC's developers were able to analyse the proprietary technology. Based on that idea, the developers were able to reverse engineer this feature, and this was considered a breakthrough. The analysis of the case study data revealed this technology was the key to uniquely position *Theta* in the automotive market for similar products.

The detailed analysis of the case study data revealed patterns of improvisation and bricolage, when EC employed the idea of reverse engineering to the feature technology. Developers and middle managers stated that they hacked, because they thought it would be worth a try. In addition, the whole idea emerged spontaneously in a peripheral discussion between Michael and Raphael. Consequently, the idea was not elaborated with sophisticated analysis, plans, tests, and so on. What they did was to design an adaptor board to connect a MOST network with EC's existing platform and attempt to reverse engineer the recording of the MOST signal. With this reverse engineering, EC constituted a disclosure of a proprietary technology. The developers relied on tinkering (hacking) and in using tools and devices at hand. Therefore, EC's developers used the material and tools available to them, without buying the licences of the inventor of this feature technology. The two main actors during the breakthrough were Michael and Raphael, who were experienced in embedded system development. Their experiences were based not only on previous training and practical knowledge, but also on theoretical know-how to master various challenges during development activities. That involved the capacity to compete in an undisclosed environment, such as the development of new products. Michael and Raphael were confident enough to put effort into the attempt, although they were not certain about the outcome. Raphael's role was critical in this breakthrough phase, because his individual efforts cleared the way for other developers of EC. In addition, the analysis exhibited that the duration of this phase was about six months for four people within EC. Therefore, the idea to pursue the effort until the breakthrough showed some planned serendipity.

In addition to that pattern of improvisation and bricolage, the data yielded the attribute of being a breakthrough of this kind of practice. The development of this feature technology of project *Theta* was a breakthrough of known boundaries, because this feature technology was new for everyone except the inventor of the technology. However, since this technology was reverse engineered, it is an instance of improvisation and bricolage, because the development's objective was clear but the way to achieve that goal was through hacking and using the tools and devices at hand. The analysis of data made it clear that the development manager, project managers and the developers perceived this as a breakthrough during the development of *Theta*. In retrospect, many developers at that time mentioned that hacking was just a way to cope with the circumstances. However, the analysis of the case study revealed no similar event at EC after the breakthrough occurred. Therefore, that breakthrough kind of improvisation and bricolage was unique among the various activities at the case study site. Furthermore, the breakthrough was the essential step of project *Theta* in developing that feature technology.

5.4.2 Patterns of innovative practices

After that supreme effort to develop the prototype of project *Theta*, series development started to proceed. Naturally, the development of an embedded system involved highly diversified tasks, such as project management, hardware development and software development. An overarching process was the eliciting and managing of requirements, because it involved the agreement of customers and the specialist departments of EC. For example, the power supply of the new embedded device inside an automobile was critical, because power is limited when the engine is not running. Therefore, discussions between the customers, project manager, hardware specialists and software specialists were necessary to solve that issue. Moreover, some requirements were formulated ambiguously, such as the way to operate product *Theta*. Clearly, product *Theta* needed client software that enabled customers to retrieve recorded data from the storage. However, Gabriel and Martin mentioned that the negotiated requirements were not sufficiently explicit to solve that issue in a straightforward manner. Gabriel's idea was to keep the client software simple for the developers and to develop a text-based interface. In contrast, Martin's conception involved a user-friendly graphical interface, which needed to be started from scratch. Concerning the concept of the client software, Martin's conception became accepted. In addition, the early phase of project *Theta* involved also fluctuating requirements, prompted by changing demands by customers or altering market conditions for hardware devices (chips, and so on). This is a common challenge in embedded system development projects, and requires managers and developers to be able to cope by developing new solutions. The analysis of this case study disclosed also that unusual answers were sought to cope with the management and technical issues inside EC. For example, the change-control-board meeting became a non-formal standard of the work activities of middle managers and developers, but Martin first introduced it when the growing complexity of project *Theta* called for a different way to manage this project well. After several change-control-board meetings were held and participants realised the necessity of that meeting, it became a non-formal standard procedure at EC, according to Raphael and Martin.

The analysis showed that the developers and managers of *Theta* strived for many new solutions, in order to cope with various challenges. For example, the change-control-board meeting was a vital procedure during the development and it started almost accidently, when Martin was looking for ways to cope with the growing complexity. He looked for a new way to manage project *Theta* and improvised that new meeting, which later became a non-formal standard. In addition, although the requirements for the client software were ambiguous, developers needed to quickly create something out of nothing. However, after the breakthrough, less and less was left to chance, because middle managers and developers mentioned that more planning and less serendipity occurred during this period. The project manager and assigned software developer engaged with the task to program the client software by improvising with the tools at hand (available editor and compiler software at EC; free open source libraries downloaded from the internet). The assigned software developer had previously gathered some experiences with programming graphical user interfaces and Martin was capable of doing the software architecture. Moreover, fluctuating requirements demanded innovative solutions that were partly improvised. Developers and the middle managers mentioned some releases needed to be configured according to recent changes of requirements, and a common way to fix things for the moment were hacks. Research participants defended that way of coping with the lack of time to find more sophisticated solutions, plus those solutions were at the time useful and necessary to immediately accommodate the expectations. Notwithstanding to those improvisational and bricolage solutions, developers referred to the sophisticated solutions, which were accomplished at a later date. Therefore, the analysis revealed the employees of EC completed many innovative achievements through improvisation and bricolage activities. Yet, this phase took another six months with seven people within EC.

Besides the pattern of improvisation and bricolage, the case study offers the element of being an innovative form of improvisation and bricolage. Those acquired solutions during project *Theta* were important for EC and were unconnected to ideas that had been established inside EC earlier. The discovered innovative solutions exhibited a great degree of flexibility, novelty and creativity. Those improvisational and bricolage activities were part of the accustomed activities from developers and managers at EC. Innovative solutions accomplished through improvisational and bricolage activities were sustained throughout the project and remained vital procedures at EC (e.g., change-control-board). Furthermore, the project development benefited from the solutions, because expectations from customers were met in time, which nurtured additional orders. In contrast to the previous breakthrough period, the accomplished solutions at this phase did not show the uniqueness of the reversed engineered feature technology. However, that stage of project *Theta* was a period with many innovative solutions, accomplished through improvisation and bricolage.

5.4.3 Patterns of adaptive practices

When the different expectations of customers concerning project *Theta* started to go beyond the initial scope, Martin defined with the initiator a baseline of requirements, in order to guarantee a minimum of agreed-upon features. With that baseline, project *Theta*'s requirements remained relatively stable, apart from the excessive desires of customers that needed to be clarified in sub-projects of *Theta*. However, the analysis of the case study disclosed that the development team needed to cope with challenges, such as ambiguously defined requirements and hardware proclamations. Therefore, marginal adjustments of the development specifications were necessary to accommodate the well-being of the development activities. In addition, their possibilities to act were limited by the path dependence of the previous development phases of this project. Besides those technical issues, which became relatively stable at that time, EC's employees rehearsed organisational processes. In addition, the prevalent processes were supplemented with necessary new processes, such as the change-control-board meeting, in order to assist the development of project *Theta*. The available set of organisational processes was only slightly adapted to cope with emergent circumstances, such as fluctuating requirements, necessary prompt actions and the increasing number of human resources. Moreover, the economic circumstances of EC remained stable during the development of project *Theta*. Therefore, EC did not suffer any impacts from outside concerning project *Theta*.

Those small adjustments of development specifications and organisational processes were important to cope with various difficulties during project *Theta*.

Although the circumstances remained relatively stable at the time, unclear requirements and sudden changes (e.g., hardware proclamations) occurred. In addition, slight adaptations of organisational processes were prevalent, as the analysis of the case study revealed. Those adjustments involved no planning or other forms of systematic attempts by managers or developers at EC. Rather, it was the consistent advance of efforts undertaken by EC's employees, in order to work on project *Theta*. That work involved improvisation and bricolage activities, referring to actions and practices without sophisticated coordination, using available tools and human resources, and based on knowledge and experiences from individuals at EC. Those actions and practices were more associated with serendipity, based on the understanding of developers and managers. At last, this phase took more than 30 months and when time moved on the organisation hired new developers and the *Theta* developer team grew to 19 people.

In comparison with the previously described patterns of improvisation and bricolage (breakthrough and innovative practices), the analysis revealed that the flavour of those improvisational and bricolage activities was different. Many actions and practices resulted mainly in doing things better and in coping with the unavoidable. Those activities showed a high amount of rigid, conventional and uniform unfolding. In other words, those adaptive practices were important for project *Theta*, particularly when the development started becoming steadier. The requirement baseline and a sophisticated set of organisational processes helped to enter a phase of project *Theta* that was almost stable. Moreover, the analysis exhibited the major development phase of project *Theta* was finished and the project team re-adjusted its focus to improve the maintainability. Consequently, adaptive skills became more important, as in previous development phases. However, improvisation and bricolage activities were still useful and necessary to cope with unexpected circumstances and to retain some vividness inside EC in order to improve processes.

5.4.4 Situatedness in the social context

5.4.4.1 Wider social context

The analysis of the case study data showed that the wider social context implicated an interesting field for researchers concerned with process management frameworks such as ISO 9001. Many Western European organisations wanted to share the benefits of such certifications that were described in section 2.2.3. Particularly in the German business context, many organisations needed to qualify with valid standards. For example, it would have been required that organisations obtain ISO 9001 certification if they wanted to market their products in the automobile sector in Germany. By prescribing business processes, the certificate promised a sustainable foundation for professional quality management systems. Many automobile manufacturers demanded that their suppliers achieve the certificate. Although ISO 9001 is known as an indicator for professionalism, it was not the only certificate that automobile manufacturers ask for. For example, suppliers concerned with software development were required to obtain SPICE certification. In order to get new orders, suppliers would have needed to accommodate their processes according to a certain SPICE level. With such constraints, obtaining certifications was not only a question of improving the organisational processes but remaining on the market. Therefore, external reasons were eminently important in order to get a better understanding of the context of organisations in that market field.

Another interesting aspect of the wider social context was the context of the engineers, because it involved particular perspectives. During the analysis of the field data, unsurprisingly, German engineers tended to strive for technical competence, by valuing gathered knowledge and experience. However, it was interesting that they seemed to avoid emphasising issues such as creativity, precisely because German high-tech products had a reputation of being innovative and novel. Developers seemed to be bewildered in talking about their work related to creativity. In addition, when speaking about improvisation and bricolage, most interviewees understood those activities as unprofessional, and therefore they had a bad reputation. Improvisation and tinkering were acknowledged as activities of amateurs and hobbyists. Consequently, improvisation and bricolage were exactly the opposite of the gathered knowledge in which engineers were educated.

5.4.4.2 Social context of people at EC

Another interesting aspect of this case study was the intricacies of the developers' social context. Inside EC, individuals with different responsibilities (such as hardware development, software development, project management, and so on) had various perceptions concerning how to approach development activities. The various technical specialists facilitated the development. However, the project management in general and the interaction between developers and the project manager was more of a balancing act. The continuous bricolage and improvisational activities from the developers hindered the efforts of the project manager. However, a common form of understanding of the technology evolved between the developers and the project manager, particularly during the critical initial phase of *Theta*, when the developers and project manager worked overtime in order to overcome the technological challenges as described in the case study. The project manager was respected as a core member of the team, because he provided the domain knowledge in that development and he placed emphasis on the needs of the customer in order to avoid any over-engineering of the device and its software. When the project was in the maintenance phase, the project manager was located apart from developers, hence their day-to-day interaction with the developers decreased. Although from time to time there were some events for socialising, the closeness between the project managers and developers was lost.

Besides that relationship of developers and project manager, the individual context of the various developers was also interesting. Particularly the role of Raphael, the fire fighter, was very interesting, because he had tremendous influence in the development of *Theta*. In addition, I found that the fire fighter in an ISO certified organisation seemed to counter the restrictions of ISO certification at EC. A fire fighter compensated for the hindered creative and critical thinking potential of the developers within the organisation. In contrast to Curtis's (1988) exceptional designer, the fire fighter role portrays someone who does not function according to ISO, but rather uses more unorthodox methods for achieving goals. In addition to that role, the case study analysis revealed that the other developers participated in the given procedures and culture. Those characters were also very important for EC and its development efforts, because those individuals shaped the capabilities of EC's development department as such. Although they were capable in their individual domain, no single one of them appeared to be exceptional. However, all developers seemed to share a common team spirit for the project they were involved in. Using a metaphor, all developers played a part in the orchestra of EC's development department and were therefore important for the musical performance. Therefore, it was important that they harmonised in the group play in order not to sound off-key. In the realm of embedded system development, the final product was more valued than the individual performances, even though they could be important at the time they occurred.

Along with those individual differences, developers shared a common understanding of the procedural setting at EC. Process management frameworks such as ISO 9001 were mainly perceived as constraining, although they realised the necessity of it. What were thought-provoking were the data that indicated hacking or muddling through were understood as positive, whereas tinkering and improvising were associated as negative. According to the detailed analysis, developers were usually educated and trained to act in a sophisticated manner that involved planning before doing. Some data gathered in interviews showed that developers tried to describe their activities as organised and structured. However, on enquiring what those organising and structuring activities involved, developers found it difficult to explain those practices to the interviewer. This does not mean that developers acted without organising and structuring activities, but those practices might be problematic to describe retrospectively. In addition, those activities might not have involved the absolute distinctive steps of planning and executing, and therefore it was problematic for the developers to describe. Nevertheless, just because some research participants refrained from being tinkerers and improvisers, the collected field data fit the descriptions of improvisers and bricoleurs found in the academic literature. Moreover, the development manager described the people at EC (developers) as “tinkering enthusiasts who like to fix problems.”

The development and project managers realised that improvisational and bricolage activities were involved in developing *Theta*. However, both got along with that circumstance although they knew that the process management framework of ISO 9001 required annual assessment regarding the activities at EC. The managers expressed concern regarding ISO 9001 and its practical help. For example, Martin stated that at the beginning no one gave any thought about structural guidelines to run the project. Their main concern was to obey the ISO 9001 regulation at the organisational interface for procurement and documented meetings with other organisations. Aside from these concerns, they ignored the standards for their day-to-day work. The analysis revealed for example that their approach for the requirement of elicitation and management was particularly risky. The reason for that was that they relied on the same way to cope with that task as with much smaller projects that EC had previously realised. The old way meant that the project manager was responsible for the requirements by using the project manager’s mature understanding of the wants and needs of the customer. However, Martin was capable of rising to the challenges of this task. His independence of software tools to support the requirement elicitation and management tasks showed an adequate understanding of the necessary actions and practices. The advantage of having this mature understanding of the wants and needs of developers at the first customer site was a critical success factor for *Theta*. In addition, Martin needed to improvise and collect all available information to complete the requirements. However, the fact that a complete list of requirements was never formulated propagated itself in the development lifecycle. Consequently, improvisation and bricolage was not restricted to the requirement phase, but was observable throughout the entire development.

Departments that had only marginal influence on the development activities at EC obtained a different view about improvisational and bricolage activities. For example, the employees from the marketing department mentioned that they perceived some indications that developers may not have worked entirely according to the standard processes of the organisational ISO 9001. The people from the marketing department felt that probably at the start of project *Theta,* many unconventional activities took place. However, in order to accommodate customer desires, they would not have reported those work styles and they emphasised the high flexibility of EC. In addition, the quality manager retained the opinion that phrases like tinkering and improvisation have an unprofessional connotation and associations of such with the work practices of EC had to be avoided. In examining those data from the case study, it appeared to be clear that some of the representatives of EC swept the notion of improvisation and bricolage under the carpet. At fairs the marketing experts focused on the flexibility of EC and the quality manager refused to admit some of the actions and practices of EC's developers and managers. Consequently, they acclimatised themselves to the expectations outside EC, in positioning the organisation in a bright light.

Although the detailed analysis of the case study revealed many differences about the social context of people at EC, there were some continually identifiable common issues. The informal work atmosphere and work team culture within EC influenced the development activities. For example, much of the information collected from contact persons within EC remained unverified and was taken for granted. The analysis disclosed that this trust and confidence in the abilities of EC's employees was very high particularly during the initial phase of the development, when the prototype was developed and the series development began. The increasing demands of the complex *Theta* development caused some setbacks and, as a result, some new processes were introduced. As described in the case study, the change-control-board meeting was introduced to allow internal stakeholders to discuss the various changes and developments. However, the change-control-board meeting was used not only to discuss the recent fluctuations of requirements, but also to keep up with recent developments for all stakeholders. Moreover, new team members were educated about EC's internal processes and the current issues of the complex *Theta* development. Therefore, new employees were able to familiarise quickly and to get involved in the work team culture. In addition, new employees were hired to fulfil tasks that had previously been handled inadequately. Accordingly, they felt indispensable in their new environment and were integrated quickly into EC. Those simple tasks did not take many efforts, but were valuable to nurture the informal work atmosphere and work team culture. Obviously, the work atmosphere and cultural setting positively influenced the managers and developers of project *Theta* to improvise and work with the available items.

Finally, the case study yielded also data concerning the top level's management and its efforts to create an ambience for the managers and developers at EC. Field data from EC's internet homepage and data from the transcribed interviews disclosed no indication that the top-level managers were aware of the enhanced improvisational and bricolage activities. The homepage advertises their management efforts according to ISO 9001 standards, their application of Kaizen, and that SPICE also belongs to their process management framework. In addition, the interviews supported their advertisements statements and pointed out the value of ISO 9001 for EC. Beyond those issues, the top-level managers enforced informal events for all employees at EC. For example, during *Theta*'s development EC held its five-year celebration on an island in the Mediterranean Sea. For this celebration, all employees were invited to enjoy the get-together. Moreover, EC celebrated annual summer events and Christmas staff parties. In analysing those issues, it was beyond doubt that the top-level management sought to generate a good work atmosphere and cultural setting within EC, and the previous paragraph supported this argument. Furthermore, the obtained data revealed the top-level managers wanted to present EC generally as an organisation in good shape and with ISO 9001 certification. Those issues exhibited only that the top-level managers tried to do their best to support business opportunities and to foster the work atmosphere at EC.

In conclusion, the detailed analysis of the richly flavoured social context of the case study exhibited a complex setting. The wider context revealed that the economic circumstances set the requirement for certificates to show professionalism and engineers are taught to comply with the economic setting in aiming at technical competence. People at EC made known an interesting social setting of the circumstances when *Theta* was developed. For example, the social context of developers and project managers enhanced their ability to cooperate during project *Theta*; differences between the developers put emphasis on the individual aspects, but developers also shared a common understanding of the procedural setting at EC; development and project managers exemplified improvisational and bricolage activities to the developers through their behaviour; other departments at EC neglected the occurrences of the developers; and all employees at EC enjoyed an informal work atmosphere and work team culture, which was nurtured by the activities of the top-level managers. In sum, the research participants were not able to describe their activities as improvisation and bricolage as such, because their social context was in many ways influenced such that those actions and practices needed to be avoided. Nevertheless, the field data analysis revealed a colourful image of the various effects on how the social contexts were shaped at EC.

5.4.5 Magnitude of improvisation and bricolage

Using the theoretical lens of improvisation and bricolage was valuable to get a better understanding of the social details of the research setting. Processes that were difficult to describe, control, and manage were better acknowledged. Emergent processes were identified and described. The analysis of the case study revealed different patterns of improvisational and bricolage activities: breakthrough, innovative practices and adaptive practices. Those determined patterns complete the analysis of the case study in answering what processes occurred during the embedded system development of project *Theta*. In addition, the social context of improvisational and bricolage activities was elaborated, and the analysis disclosed a richly flavoured social context of the case study. Therefore, the realms of structured processes and improvisation and bricolage complemented the analysis by acknowledging the full flavour of processes in the social context. Although I know what processes occurred and how these processes were shaped, the analysis was lacking in explanation of the human action and the motives of individuals. As a result, another focus was required to adequately explain the individual within the social system of the case study.

5.5 Concepts of reflexivity and human agency

Vital aspects of social life, such as reflexivity and human agency rounded off the analysis of the case study, because those concepts provided a better understanding of the individuals, their actions and motives. Those concepts were used in analysing the following: section 5.5.1.1 begins with analysing the structured approaches that were influenced by top-level managers; section 5.5.1.2 examines the structured approaches that were influenced from members of project *Theta*; emergent structured approaches are investigated in section 5.5.1.3; and finally, section 5.5.2 analyses the improvisational and bricolage activities of EC's employees. In combining the concepts of reflexivity and human agency for analysing the case study, this section continues to interpret the field data.

5.5.1 Structured approaches

5.5.1.1 Structured approaches influenced by top-level managers

The analysis of the field data revealed several structured approaches at EC that were influenced by top-level managers. For example, EC's top-level managers triggered the establishment of ISO 9001 certification within that organisation. Therefore, they accommodated requirements of ISO 9001 assessments, such as establishing a position that is concerned with quality management within EC, creating organisational processes according to ISO 9001 and educating EC's employees about the quality management standard. Although EC's top-level managers acknowledged possible side effects, their desire to establish the management framework was not necessarily only an intrinsic motivation. EC's business market demanded the framework as a legal requirement. Therefore, they could have acted otherwise and refused to become an ISO 9001-certified organisation. However, that would have meant that EC remained a small organisation with the ability to cover only a niche market. But just because the top-level managers discursively stated the internal reasons why they wanted to become ISO 9001-certified, it was clear that this step was on purpose.

A detailed study of the ISO 9001 standards disclosed an interesting parallel. ISO 9001 standards obtained an approach, which had similarities to reflexivity in social sciences. ISO 9001 wanted with different defined tasks, such as measuring, analysing and improving, to demonstrate conformity to product requirements, ensure conformity of the quality management system and to continually improve the effectiveness of the quality management system (ISO, 2008b). In other words, ISO 9001 organisations sought to gather data concerning their business process, and quantify those data in order to analyse it. Afterwards, those analysed data were taken as reference, in order to take actions to improve processes. On the other hand, reflexivity in social sciences involves a monitored character of the continuous flow of social life (Giddens, 1984). This is a form of measuring, in the sense that its participants are constantly assessing the activities of development. That reflexive monitoring was dependent on the competence of social agents, in terms of their capacity to rationalise (analysing) ongoing social life. And purposive actions (improving) were embedded in the ongoing flow. That detailed analysis linked a framework that relies on quantitative measurements in order to rationalise actions with reflexivity, an important theme of Structuration Theory. Nevertheless, the developers and managers routinely acted upon the processes in which they participated without further thinking. Consequently, activities were often unconsciously reflexive or taken-for-granted and did not pass through "consciousness." Although the developers were not normally aware of reflexivity, they became aware of it through conscious reflection (strong potency of reflexivity) or when some form of disruption to routine occurred.

Kaizen (Japanese philosophy that focuses on continuous improvement) was introduced at EC because the founder had experienced that idea previously. Although ISO 9001 already involved the aspects of continuous improvement, employees at EC were educated about Kaizen. However, the analysis of the case study revealed no indication that Kaizen was heavily used during project *Theta*. For many developers, Kaizen seemed to remain too hollow without any solid tasks to perform it. For example, developers stated that Kaizen lacks of practical examples that aid to utilise the principle of Kaizen. Furthermore, managers of EC did not demand any form of Kaizen accomplishments by the employees. The authority to assess those Kaizen achievements was lacking at

EC and consequently EC employees were concerned more with other issues. Moreover, that lack at EC provided no motivation to utilise Kaizen better. However, that familiarisation may have contributed to the unconscious insight that EC valued efforts to improve the work circumstances somehow. In addition, the buzzword may have helped to evoke thoughts about rethinking work circumstances, or in other words, to reflexively monitor actions and practices of the daily life. So, the buzzword Kaizen may have not had any direct impact on the circumstances at EC, but the circulation of that idea influenced many employees in an unconsciousness way.

The case study described a questionnaire that was administered at the end of *Theta*'s development. The top-level managers required the project team members to answer those questions and bring forward any ideas how to improve procedures at EC for future developments. The questionnaire was received positively from the developers and middle managers, because it gave them the impression that the top-level management knew about the various difficulties during the development of *Theta*. Moreover, the management was willing to reflect on those issues together with the people who had experienced the challenges. Therefore, the motivation, acceptance and expectation of the developers and middle managers toward any results of that questionnaire were high. Interestingly though, the case study data revealed no direct consequences by the top-level managers on the basis of the responses to the questionnaire. The analysis of field data disclosed that the questionnaire involved a significant potency of reflexivity. For example, developers and the project manager came up with a detailed assessment of what worked and what didn't work when the founders of EC asked them to answer the questionnaire about the observed actions and practices during the development of project *Theta*. This shows that some reflexive practices may be enhanced by particular managed actions.

5.5.1.2 Structured approaches influenced from members of project *Theta*

In addition to the structured approaches that originated from EC's top-level managers, some standards were influenced from members of project *Theta*. Important tools were established and helped the developers in their day-to-day work. The case study data revealed developers and middle managers appreciated tools like Subversion and Bugzilla, because they supplemented EC's processes. Subversion was used to coordinate the development of the various software versions better. Bugzilla was implemented to track errors of product *Theta* during the development process. Although the organisational ISO 9001 standard prescribed different tools, EC's developers welcomed Subversion and Bugzilla, and the managers at EC supported this. The analysis disclosed the issue that Subversion and Bugzilla were implemented organisation-wide recently and the developers acknowledged the need for improving the situation with tools that were well known in the community of software developers. Therefore, after talking to the development manager, some developers put some effort into the setup of those databases at EC. Those developers mentioned that the undertaking to establish Subversion and Bugzilla happened after they realised the need for this in order to accommodate development efforts of EC.

Furthermore, the analysis revealed a recurring behaviour in the particular reflexive practices of Raphael and Martin along with other developers and managers at EC. Raphael claimed in interviews that he was an autodidactic (self-taught) person, willing to contribute to the well being of project development, whereas Martin stated that he assimilated various views from stakeholders in his work. Raphael and Martin paraphrased their broader understanding of the situation around them (project *Theta* and organisational circumstances), but did not refer to it as a reflexive practice because they stated that this broader understanding came along with their ability to optimise available resources to fix problems. As human actors, developers and managers reflexively monitor their actions and those of others in contexts of activity (Giddens, 1984). This reflexive behaviour might take place in the discursive (known) or practical (undisclosed) consciousness. The potency of conscious reflexivity was considerably higher when questions about their actions were asked. In addition, the example of the questionnaire indicated that some reflexive practices may have been stimulated by particular managed actions.

5.5.1.3 Emergent structured approaches

The analysis of the case study revealed emergent structured approaches during project *Theta*. For example, the usage of Bugzilla altered over time in a way that the developers tracked unknown incidents and events of products of *Theta* behaviour in addition to obvious errors. That was interesting, because it showed how ambiguous the requirement specification of project *Theta* was. Furthermore, the development of new hardware and embedded software was challenging and involved errors. Those difficulties triggered the idea of reporting obscurities during development, in order to carry on with development activities. Furthermore, the change-control-board meeting emerged as an important weekly event during the development of project *Theta*. Before that meeting, the project manager realised the need to continue the project development, after important internal stakeholders were involved in discussing and deciding how to proceed with development activities. On a whim, the project manager asked those stakeholders to meet to do what was necessary at that time. Without that meeting, decisions would have been delayed and valuable time would have been lost. Therefore, the stakeholders became aware of the fact that it was urgent to continue with the development. In coming regularly to that point, when necessary decisions had to be made, over time that meeting became a weekly meeting. The change-control-board meeting was very valuable for EC during project *Theta*.

5.5.2 Improvisation and bricolage

The case study described varies moments of improvisation and bricolage during project *Theta*. The previous sections analysed this and distinguished different patterns of improvisation and bricolage, such as breakthrough, innovative practices and adaptive practices. Beside that insight, improvisation and bricolage share some important similarities. The detailed analysis of the field data revealed that people who perform improvisation and bricolage had to maintain a certain reflexive behaviour, in order to act purposively. For example, the development managers described the developers as "tinkering enthusiasts," and that was of great importance, because enthusiasts were not only experts in their field but brought an intense and eager interest to what he or she was doing. Therefore, those tinkering enthusiasts were determined in what they were doing; they have reasons for their activities, and they were able to explain those reasons. The developers and managers at EC were eager to develop project *Theta* and every pattern of improvisation and bricolage involved that dedication.

During the initial phase and breakthrough period, the project team members were excited about the opportunity to develop something new. Naturally, the newness involved ambiguities and uncertainties. In order to remain goal-oriented, the project team needed to handle their actions and practices. Accordingly, handling involved a monitored character of ongoing activities fitting into the setting of embedded system development. Derivations of the original target to develop the feature technology would have been disastrous for project *Theta;* developers and managers were aware of the possible consequences in the case of failure. Although the analysis did not uncover any form of extrinsic motivations for the project team members, such as bonuses or other forms of awards, it can be assumed that developers and managers were intrinsically motivated to accomplish that goal. The reason for that lay in the form of excitement to develop something new and important.

When improvisational and bricolage activities obtained more innovative practices, developers and managers were still excited about the development project. Inherently, although the share of ambiguity and uncertainty decreased significantly, series development involved a different spectrum of challenges. Developers and managers coped with those arising difficulties and improvised and practiced bricolage activities. For example, the initial start-up of the change-control-board meeting was improvised, and Martin mentioned that this seemed to be an appropriate way to cope with the difficulties at that time. In analysing the field data, the same pattern of reflexive behaviour was prevalent during the innovative practice of improvisation and bricolage. The project managers' interpretation of the situation at that time argued for a different approach, in order to speed up development and to accommodate decisions along with stakeholders of project *Theta*. Acting otherwise, for example solo attempts or maintaining the traditional way at EC, would have had bitter outcomes. Solo attempts may have had negative consequences for Martin, because passing developers over colleagues would have critically disturbed the work atmosphere. Moreover, maintaining the traditional way at EC would have lengthened the process to come to a final decision. Therefore, Martin acted not only purposively but acknowledged the social practices at EC, such as teamwork. The power to initiate that meeting was certainly part of the responsibility as project manager. Nevertheless, Martin established that meeting because he rationalised the sheer necessity of it and that finding was grounded in Martin's ability to reflexively monitor the continuous flow of project development work.

As the previous section indicated, after a challenging time of technical endeavours and organisational alterations, the adaptive practices of improvisation and bricolage became apparent. Obvious changes on the outcomes of those practices were difficult to identify, because it involved many little actions and practices of managers and developers to make things better.

However, improvisational and bricolage activities were necessary to cope with ongoing difficulties, such as the quick integration of new employees. Although new employees at EC quickly obtained responsibilities, an ongoing problem was to integrate them appropriately so that they were able to deliver results quickly. The people at EC realised that problem and one solution among others was the participation of new employees in the change-control-board, which had already become at that time a weekly meeting. Although it seemed to be a small step to realise that new employees might need help to get integrated quickly, first it needed the actors to do so and second it required adequate adaptations of the prevalent social practices. Some actors were able to become conscious about that necessity, because they reflexively monitored how and when new employees were sufficiently integrated in project development work. Those actors wanted not only to help the new employees to become better integrated, but also project *Theta* and EC benefited from better-involved employees. Therefore, a reasonable way to embed new employees better was to invite them to important meetings, even when they were only passive observers. As observers, they disturbed the ongoing meeting only slightly, gathered information about the ongoing activities of EC and acquired a feeling for the various people at EC.

Many improvisational and bricolage actions and practices were achieved because of the ambiguous and fluctuating requirements. Those changes required technical and organisational flexibility in which they could occur. A too tight corset of technical limitations and organisational regulations would have limited the success of improvisational and bricolage activities found in the field data. In addition to those circumstances, the developers and managers knew what they were doing, because improvisation and bricolage took place as a way to cope with ambiguous and fluctuating requirements. That presumes actors were purposively and well-informed in what setting they needed to respond. That ability of reflexivity is bonded to the actors' will and capability to do so. However, being the driving force of improvisational and bricolage activities might have led to either intended or unintended consequences. Nevertheless, reflexivity of the actors straightened out some of the unintended side effects.

5.6 Interacting influences

The analysis of the case study revealed a linkage between the utilisation of process management frameworks and improvisation and bricolage at EC. As previously indicated, the use of certain software tools was combined with the associated processes for developers and managers. In particular, two software tools induced positive results in managing the processes of project *Theta* systematically. Subversion was a version control system that helped the developers keep track of the multiple modifications made over time. The second tool was called Bugzilla, which was used to track bugs during development projects. Both software tools provided various processes and those processes were inherited by the developers and managers in using those tools. Interacting influences were exposed as corresponding factors between ISO 9001 processes and improvisation and bricolage.

First, the influential factors of ISO 9001 on improvisation and bricolage were analysed and the data disclosed that the motive for seeking certification was weak in general throughout the developers and middle managers. However, higher up in the hierarchy the awareness grew of the positive contributions of a certified process management framework. Studies by Dick (2000) show that the motive for seeking certification tends to be an important factor for later performance of the implemented practices and techniques. Because of the weak motivation of the developers, the direct influence of ISO 9001 onto developers' work on the embedded system was fragile. Software tools like Subversion and Bugzilla influenced the developers' actions significantly more than ISO 9001 because of their daily interactions with these software tools. The use of software tools and processes and regular meetings were central to the developers' day-to-day activities, especially during the later stages of development.

Second, the effects of improvisation and bricolage on the structured processes were examined during embedded system development. The analysis disclosed that innovative improvisation and bricolage activities were more dominant over structured processes. However, at a later stage of *Theta*'s development, innovative practices of improvisation and bricolage were less important than structured processes. The case study analysis highlighted the early phase of project *Theta*'s development, when Bugzilla was not really used, in contrast to the maintenance phase when Bugzilla was used intensively. In many cases, developers reported various incidents (not errors) on their bug-tracking tool. As a result, this tool became an incident reporter alongside its original purpose. Through continuous use, software tools and processes were transformed and established in their given context because of improvisation and bricolage actions. In this way, improvisational and bricolage actions and the process management tools in use were mutually shaping each other. This usage of tools for improvisational and bricolage actions helped to instantiate them and sometimes helped to modify or transform them in their use. In doing so, the use of tools became more acceptable and they became established in the organisational context as acceptable practices, and they then became part of acceptable routine practices with each subsequent use.

5.7 Summary

This chapter presented an analysis of the embedded system development process at EC featuring three emergent analytical topics: structured processes, practice of improvisation and bricolage, and reflexivity and human agency. In sum, organisational process management framework had only marginal influence on the activities of developers and middle managers. Actions and practices of developers and managers involved improvisation and bricolage, which became modified in regards to the various stages of a system development process. Improvisational and bricolage activities reciprocally influenced the utilised standard procedures at EC. Moreover, the reflexive behaviour of managers and developers was important for the progress of the development project.

The following chapter features a discussion of the understanding gained, which culminates in a bottom-up theory concerning the development processes in an embedded system development context.

"Discovery is seeing what everybody else has seen, and thinking what nobody else has thought."

Albert Von Szent-Gyorgyi (1893-1986) Hungarian scientist

6 Discussion

6.1 Introduction

The previous chapter presented an analysis of the embedded system development process at EC featuring three emergent analytical topics: structured processes, improvisation and bricolage, and reflexivity and human agency. This chapter features a discussion of understanding gained that corresponds the analysis findings with the literature of the theoretical lenses. First, in section 6.2 are the examined improvisational and bricolage activities related to the theoretical constructs to provide a better understanding how the developers and managers practiced improvisation and bricolage. Accordingly, section 6.2 aims to answer research question 1. Section 6.3 investigates the role of reflexivity and human agency in the practice of embedded system development. Finally, section 6.4 draws on the findings of this research to present a conceptualisation. That conceptualisation aims to describe what measures help to balance the tension between process management and improvisation and bricolage, hence to answer research question 2.

6.2 Improvisational and bricolage activities

The analysis disclosed a variety of improvisational and bricolage activities that were part of the day-to-day activities of managers and developers in the case study setting. This discussion shed some light of improvisational and bricolage activities of developers and managers in the context of embedded system development.

6.2.1 Intended and ingenuous activities

The analysis explored how an organisation with established process management develops an embedded system in practice. The process management framework of ISO 9001 reasoned various constraints on the development activities of the people at EC. Particularly the activities that involved a lot of creativity were hampered (Aaen & Pries-Heje, 2004), but creativity is seen as important in all aspects of ISD (e.g. Gallivan, 2003). Kondo (1996) describes both elements of creativity and PM as important for organisational management, although they are thought to be mutually exclusive. The developers' constraints identified in the analysis confirm this finding. Moreover, the literature review indicates, that the tensional arrangement between attempts to manage and to be creative has been analysed and discussed by many researchers (e.g. Adler, Goldoftas, & Levine, 1999; Benner & Tushman, 2002; Ciborra, 2000; Cunha & Cunha, 2006; Eaglestone, et al., 2007; Edmonds & Candy, 2002; Gallivan, 2003; O. Hoffmann, et al., 2005; Plsek, 1997; Sawyer & Guinan, 1998; Teece, 1996).

Along with those findings in the research literature, the findings of this research can be integrated in the ongoing discussion. This research focuses on the process management framework of ISO 9001, which involves the three main activities to introduce and remain ISO 9001-certified: document routines, process improvement, and continuous comparison of process effectiveness and efficiency (Benner & Tushman, 2002; Harrington & Mathers, 1997). Definitely, those activities favour to determine routines and eliminate vagueness and therefore they have influence on improvisation and bricolage. For example, a popular belief of practical advantage of an ISO 9001 certification is process standardisation (e.g. R. Jones, et al., 1997; Jovanovic & Shoemaker, 1997; Rada, 1996; Singels, et al., 2001). However, exactly those efforts that try to channel the activities of developers and managers are quite the opposite of those that constitute them. This analysis revealed that the developers ignored the ISO 9001 standards for their day-to-day work. Particularly at the initial stage of the development, developers and managers conducted their work the same way as in previous projects at EC. That meant they improvised and performed bricolage and followed no prescribed guideline. This finding confirms earlier research on the limitations of using prescribed standards in ISD, which state that practitioners do not fully accept proposed standards (e.g. Fitzgerald, 1998; Nandhakumar & Avison, 1999). In addition, the analysis revealed a difficult situation concerning the ambiguous and fluctuating requirements during the development of *Theta*. Developers and managers tried to cope with that situation through improvisation and bricolage, and that involved activities, which were not prescribed and suggested by the organisational ISO 9001 guideline. This finding confirms du Plooy's (2002) research, that the application of structured methods in ISD tends to fail in turbulent environments. Furthermore, this research supports the findings of Fitzgerald (1998) that standards in ISD are too general without any solid foundations and contextual considerations.

Moreover, the analysis revealed various examples of response to constraints at EC and identified two patterns, namely intended and ingenuous (unplanned) activities (cf. Table 5-2). Intended and ingenuous activities correspond to the examined improvisation paradoxes of pragmatic creativity, oriented drifting, and managed serendipity. Intended activities share respective characteristics that involve high pragmatism, strong orientation, and management. For example, the successor of CVS was just better and therefore developers and managers decided to employ that development aid, instead of using the proposed CVS. Ingenuous activities share exactly the opposing strengths: high creativity, likely possibility to drift, and expected serendipity. For example, the work process of product development involved hacking and patching activities.

Certainly, those actions were highly creative, because the hack solution finding process involves task uncertainty (Dahlbom & Mathiassen, 1993) with no fixed routine or other suggested patterns to follow. Consequently, the product development encompasses the likely possibility to drift to unintended solutions. Therefore, those who practice ingenuous activities necessitate a certain serendipity, because the activities themselves are difficult to manage. However, the experimental culture (Cunha, et al., 1999) of the observed developers nurtured the expectations that the ingenuous activities help to overcome the difficulties during the development of *Theta*. A juxtaposition of intended and ingenuous activities highlights the different strengths of the corresponding improvisation paradoxes (Table 6-1). As described above for example, intended activities involved highly pragmatic actions, whereas ingenuous activities included very creative actions that are the exact opposing strength of that improvisation paradox.

Table 6-1 Intended and ingenuous activities and corresponding improvisation paradoxes

Improvisation-paradox	**Intended activities**	**Ingenuous activities**
Pragmatic creativity	Highly pragmatic	Highly creative
Oriented drifting	Strong oriented	Likely drifting
Managed serendipity	Managed	Expected serendipity

6.2.2 Hierarchical distribution of improvisation and bricolage

The analysis described non-formal standards that were important for EC (cf. Table 5-5). Those non-formal processes involved to some extent some strengths of particular improvisation paradoxes, namely: pragmatic creativity, oriented drifting, and anxious confidence. For example, the vision of EC (metaphor of building a cathedral) expresses a gap between the real world and possibility that can encourage developers and managers to act positively outside the given processes, hence to improvise and practice bricolage

(Crossan, 1998; Mintzberg & McHugh, 1985). Improvisation and bricolage may involve increased pragmatism or creativity, and both include leaving behind prescribed processes. In addition, top-level managers decision to develop upon different platforms includes an articulated goal that Weick (1993b) describes as “magnetic field”. That magnet provides orientation for developers and managers to remain focused and to avoid too much drifting. To develop upon different platforms compromises another improvisation paradox: anxious confidence (Mirvis, 1998). Weick (1998) described shared experiences as requirement of improvisational success, because shared experiences increase confidence in the abilities of an organisation. Therefore, improvisational and bricolage activities involve the potency of success, if the organisation shares common knowledge.

Although the overview of non-formal processes initiated at EC (Table 5-5) involves a differentiation between people of lower and higher hierarchy levels, this discussion reveals less indication that this differentiation can be linked to specific improvisation paradoxes. The various strengths of improvisation paradoxes seem to be distributed throughout the organisation, without specific strengths at one hierarchy level. However, this does not imply that the creative potential is evenly distributed. Instead, the analysis put emphasis on the creative actions and practices of individuals like Raphael (fire fighter) (associate) and Michael (senior manager), who were responsible for the development of the key feature of project *Theta*. This finding supports Kirton’s (1976) findings, that people involve different styles of creativity and that some people (innovators) possess higher originality than others. Moreover, the success of *Theta* supports also the findings of Gallivan (2003), he states that organisations do not necessarily require that all employees are innovators.

In addition to the innovators at EC, other developers and managers possessed higher rule-conformity and coped with challenges by following formal and non-formal guidelines. Situated in the development context, those guidelines reasoned not always difficulties and problems and thus, developers and managers followed the way with least tension. For example, the comparison of EC's practices with SPICE level 1 depicts, that the process concerning the requirement elicitation was fully achieved (cf. Table 5-4). And that achievement shows, established PM may have some practical advantages for an organisation (e.g. Rada, 1996). However, the previous discussion highlights also that the practices of developers and managers involved improvisation and bricolage in order to overcome various challenges throughout the development process and difficulties to follow the guidelines of the organisational process management framework (ISO 9001). Moreover, the poor results of the SPICE assessment (cf. Table 5-3 and Table 5-4) do not support the findings of Stienen (1999) that an ISO 9001-certified organisation usually fulfil SPICE level 2.

I want to emphasise that structured processes and activities such as improvisation and bricolage, are contrasting, and that is an important discovery for the conceptualisation of the research findings. Figure 6-1 depicts the tensional arrangement between structured processes and improvisation and bricolage. Despite that tension, the analysis above revealed, developers and managers utilised both means to progress with embedded system development activities.

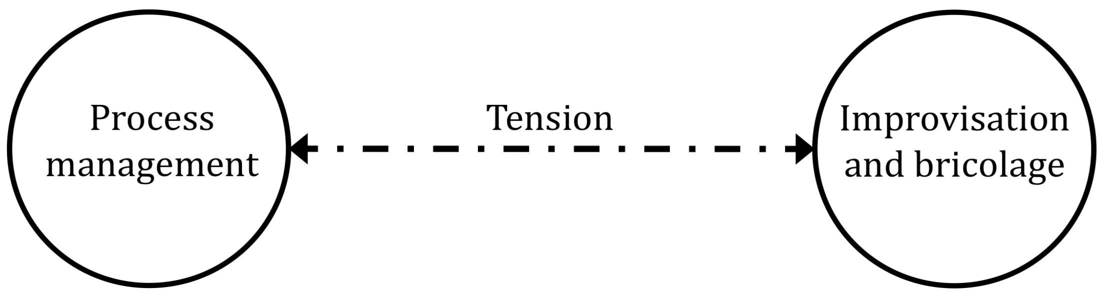

Figure 6-1 Tension between structured processes and improvisation and bricolage

6.2.3 Situatedness of improvisation and bricolage

The developers and managers at the case study site were experienced, so that many were proficient how embedded systems are developed in practice. Those experiences involved particularly the ambiguous and fluctuating requirements during development projects. The practices of the developers and managers involved improvisation and bricolage activities, because the prescribed process management framework (ISO 9001) provided less adequate help, how to cope with those difficulties concerning the requirements. As expected by the developers and managers at EC, ISO 9001 did not address all the challenges of ambiguous and fluctuating settings that may occur during complex embedded system development projects. That finding supports that process management frameworks such as ISO 9001 are too general without considering the challenges of requirement elicitation and management adequately (Fitzgerald, 1998). Indeed, the case study revealed high complexity in the task to elicit and manage requirements, and any prescriptive approaches are likely to handle the challenge unsatisfactorily.

This section differentiates between different situatednesses of requirements and how insular the response of developers and managers is. The analysis revealed that the experienced developers and managers at EC expected the situation to be that of ambiguous requirements. Because of weak specified requirements in the turbulent environments, the improvisational and bricolage activities were almost planned to occur. This oriented drifting supports Miner et al. (2001) findings, that organisations can plan to improvise without the actual content of the improvisation being planned in advance. Because this improvisational and bricolage activities occurred then relatively remote from influences by customers, the response to those ambiguous requirements were self-contained solutions. When intermediate results were presented to the customer, the drifting improvisation and bricolage received a possibly new orientation. However, the analysis revealed that the project manager was previously involved at the customer side, and this was certainly an advantage, because he was experienced enough to give the development the right orientation. The ambiguous requirements resulted in the development of a functioning prototype of project *Theta*.

In addition to ambiguous requirements, which were expected to a certain extent, fluctuating requirements provided a different challenge for the developers and managers of EC. Fluctuating requirements were experienced suddenly during the development of Theta, and therefore, the whole situatedness was contrasting to ambiguous requirements. Accordingly, the response of developers and managers to fluctuating requirements was unlike the response to expected challenges with ambiguous requirements. The sudden changes involved a sense of urgency, which required a re-interpretation of the environment, seeing it in its full richness and complexity (Crossan, 1998). Therefore, the responses to fluctuating requirements were conditioned upon the given changes of requirements and involved improvisation and bricolage. The results of those improvisational and bricolage activities were adjusted product specifications of project *Theta*.

Table 6-2 summarises the situatedness and insularity of improvisation and bricolage when requirements were ambiguous or fluctuating. Figure 6-2 depicts those findings and how ambiguous and fluctuating requirements influence improvisation and bricolage differently. Situation "A" symbolises the improvisational and bricolage activities of developers and managers with ambiguous requirements during the embedded system development. These activities were self-contained, and they motivated further improvisational and bricolage actions to create a functioning prototype of *Theta*. Situation "B" represents the different situated improvisational and bricolage activities influenced by fluctuating requirements. They necessitated a prompt response and resulted in an adjusted product specification of project *Theta*.

Table 6-2 Situatedness and insularity of improvisation and bricolage (Molnar & Nandhakumar, 2008a)

Example	Situatedness	Insularity	Quotes of research site members
Ambiguous requirements	Expected	Self-contained	Martin stated: "Although many features were not specified in detail, I think the development benefited from the experiences I collected before I started to become project leader of *Theta*."
Fluctuating requirements	Sudden	Conditioned	Gabriel stated: "A number of features seemed to be ready developed, but because requirements were changing we needed to redo it."

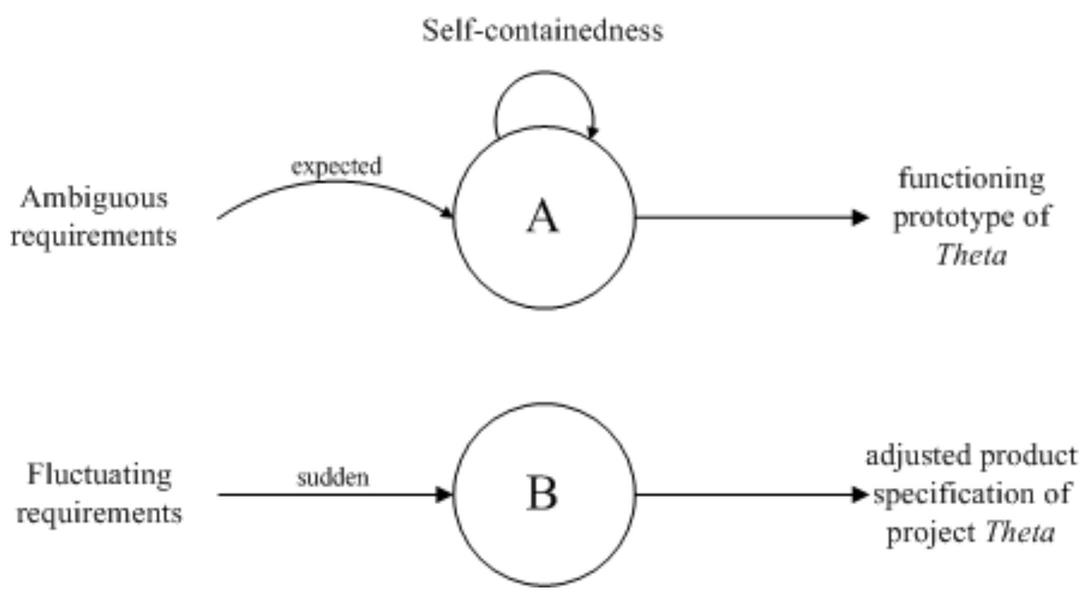

Figure 6-2 Ambiguous and fluctuating requirements influence improvisation and bricolage (Molnar & Nandhakumar, 2008a)

6.2.4 Alteration process of improvisation and bricolage

The analysis revealed different patterns of improvisation and bricolage activities throughout the development process at the case study site. The differentiation highlights the distinct phases in development projects, when after a highly innovative initial phase the development activities get more focused. Particularly the initial phase of the development involved the development of innovative high-tech solutions. In addition, many difficulties (e.g.: new platform, re-manufactured technology, refinement of requirements) were overcome through improvisation and bricolage. This discussion reveals different strengths of the improvisational paradoxes at that phase of development. For example, the initial phase of project *Theta* included many creative solutions, such as the development of the feature technology. That finding is supported by findings from Orlikowski (1995) and Weick (1998) which describe situations when radical improvisation or purer instances of improvisation occur and change happens. The ambiguous requirements at the initial stage and the hardly-followed but prescribed standards supported some drifting of the development activities. For example, the development of the client software as a graphical interface was not specified as such; however, Martin was committed to develop it so that it would be user friendly. Those ambiguous requirements could be also described as minimal structure, which express the desire to achieve improvisations that progress (Barrett, 1998; Cunha, et al., 1999). Furthermore, unplanned activities occurred particularly during the initial phase of the development and, as discussed above, those ingenuous activities necessitated serendipity. In addition, the individuality of some *Theta* developers and managers (namely: Michael, Martin, and Raphael) was very important at the initial phase of the development, because they were the striving force behind the prosperous project (Mirvis, 1998). Finally, although the managers at EC were confident to initiate project *Theta*, the findings support Mirvis' (1998) findings that anxiousness was also involved in the improvisational and bricolage activities. At the initial phase, developers and managers were unable to foresee if project *Theta* was a success at all. Therefore, anxiousness was predominant, and *Theta* was seen as just another project with limited influence for the organisations' wellbeing.

At a progressed phase of the development, many obstacles were overcome and the foundations of the architecture were specified, and the developers wrote proper code by finding solutions in the available circumstances. The different strengths of the improvisational paradoxes altered and shifted to the contradicting side. For example, although improvisational and bricolage activities still involved creative attributes, the development was far more pragmatically conducted. Weick (1998) describes those activities that shift, switch or add as weaker instances of improvisation and Orlikowski (1995) relate this as pragmatic everyday improvisations. Moreover, just because the foundations of the architecture were specified, many activities were far more oriented. Nevertheless, improvisation and bricolage were still vital, because the requirements fluctuated. In addition, the entire development approach became more managed, because the previously established understanding determined some contours of the development (Ryle, 1979; Weick, 1998). Therefore, obtained comprehensions influenced the improvisation and bricolage in important ways (Miner, et al., 2001; Vera & Crossan, 2005). Furthermore, *Theta* was always a project that was based on the capabilities of few people. The analysis revealed that the development of *Theta* involved a large team, and many people shouldered the responsibilities. However, the people that were at the initial phase very important had still vital influence on the development. Moreover, the managers and developers were far more confident in the success of *Theta*, because the troubled initial phase of the development had passed. In summary, one finding of this research is the alteration between the different strengths of corresponding improvisation paradoxes. Although other research findings (e.g. Orlikowski, 1995; Weick, 1998) indicate different instances of improvisational and bricolage activities, this research highlights the different nuances of improvisation and bricolage during a development process of an complex information system.

A juxtaposition of the different strengths of the improvisation paradoxes depicts the shift from the initial to the progressed phase of project *Theta* (Table 6-3). As described above, the initial phase involved for example to greater extent creativity than pragmatism. However, at the progressed phase of system development, that strength shifted to the reverse and involved more pragmatic than creative activities.

Table 6-3 Alteration process of improvisation and bricolage and corresponding improvisation paradoxes

Improvisation-paradox	Initial phase	Progressed phase
Pragmatic creativity	Creativity	Pragmatism
Oriented drifting	Drift	Orientation
Managed serendipity	Serendipity	Management
Collective individuality	Individuality	Collective
Anxious confidence	Anxiousness	Confidence

6.2.5 Funnelling process of improvisation and bricolage

The discussion above concentrated on the different strengths of improvisation and bricolage, and how they altered throughout the development process. This is interesting and gives rise to a discussion in perceiving improvisation and bricolage as a whole and how it was shaped over time during that development project. The analysis investigated the emergent patterns in structured processes and the various patterns of improvisation and bricolage. In merging those analysis findings, it seems that the development process of *Theta* involved a funnelling process of the actions and practices that occurred during the period that is considered a breakthrough, and the almost routinised activities that happened at the progressed phase of the project development. This funnelling process involved not only alterations of the different strengths of improvisation paradoxes, but the analysis indicates also emergent patterns of structured processes. For example, the change-control-board meeting helped to steer everyday actions and became over time part of the non-formal processes. Therefore, some improvisational and bricolage activities were covered by that routinised meeting and transformed. Another social circumstance concerned the increased number of developers, which reasoned the shouldering of responsibilities on many people. Consequently, the development activities needed to become more managed and the striving force of some individuals substituted by the capabilities of a team of developers. In addition, the analysis revealed emergent structured processes, and those processes were legitimised in the already established organisational process management framework of ISO 9001. That standard legitimised some practices, as well as prevented the developers and managers from doing anything that could not be legitimised. The analysis indicated that the organisational members changed their perception concerning the prescribed structured processes. Further to those altered social circumstances, the technical necessities changed over time too. For example, at the initial phase of project *Theta,* the requirements were ambiguous and not complete. However, over time, the requirements became more stable and the progressed development of the embedded system limited the possibilities to make fundamental changes to the project. Similarly, Ronkainen and Abrahamsson (2003) found that the rigidity of embedded software development practices has to steadily increase during a development project. Based on these findings, Figure 6-3 depicts the influence of technical limitations and social circumstances that influenced a funnelling process of improvisation and bricolage, during initial and progressed phase of the development project.

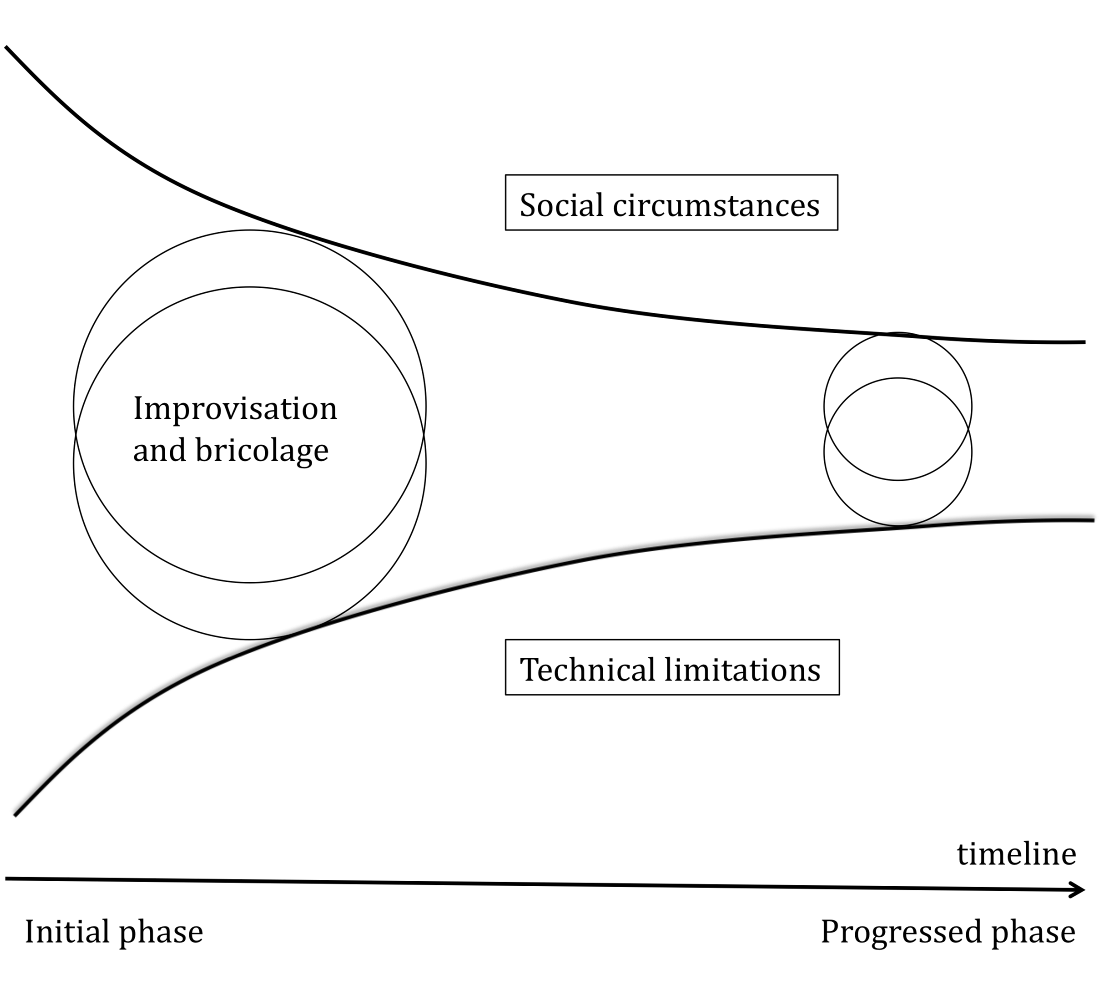

Figure 6-3 Influence of social circumstances and technical limitations on the funnelling process (Molnar & Nandhakumar, 2008b)

6.2.6 Unfolding activities

Despite EC being an ISO 9001-certified organisation, its prescribed process only partly helped during the development of *Theta*. The analysis indicates that non-formal processes, improvisation and bricolage, technical limitations, and social circumstances influenced the development activities. During the initial phase of the development, the organisational ISO 9001 guidelines were not sufficient enough to cope with the complex undertaking of embedded system development. In order to cope with the challenges of that undertaking, developers and managers improvised and performed bricolage. Indeed, the developers and managers enjoyed the emergent development and management style, because it had been part of the organisational culture. However, over time, the organisation needed to adapt to different social circumstances (e.g.: change-control-board meeting, hiring of new employees, different perspective concerning prescribed processes, organic growth of the entire organisation) and technical limitations (e.g.: ambiguous and incomplete requirements, fluctuating requirements, increased unit volume). Certainly, the developers and managers collected rich experiences with various practices throughout the development of *Theta*. Over time, best practices became practiced more regularly to finally become non-formal guidelines. That unfolding of non-formal guidelines formed a set of practices, which was sufficiently helpful that the managers and developers were able to cope with difficulties adequately. In other words, the non-formal guidelines enabled the managers to organise better. However, the compliance with the official process management framework (ISO 9001) remained limited, and only necessary issues were established, such as communication with other organisations and procurement. Many organisational ISO 9001 standards, such as reviews, early tests and documented working were not followed.

Figure 6-4 draws on the analysis and depicts the interplay of emergent and organised activities in the embedded system development context. The cloud around the emergent activities symbolises the crudeness of those activities during the initial phase of project *Theta*. The square-edged box depicts the mature application of more organised activities as observed in the progressed phase of *Theta*. The two arrows point out a continuous process between the contrasting approach types. As the more organised activities became established, the ability to cope with the complex embedded system development was lost. Therefore, the cycle proceeded with the continued emergence of more improvisational acts and subsequently these improvisational and structured processes became more formalised with their recursive practices over time.

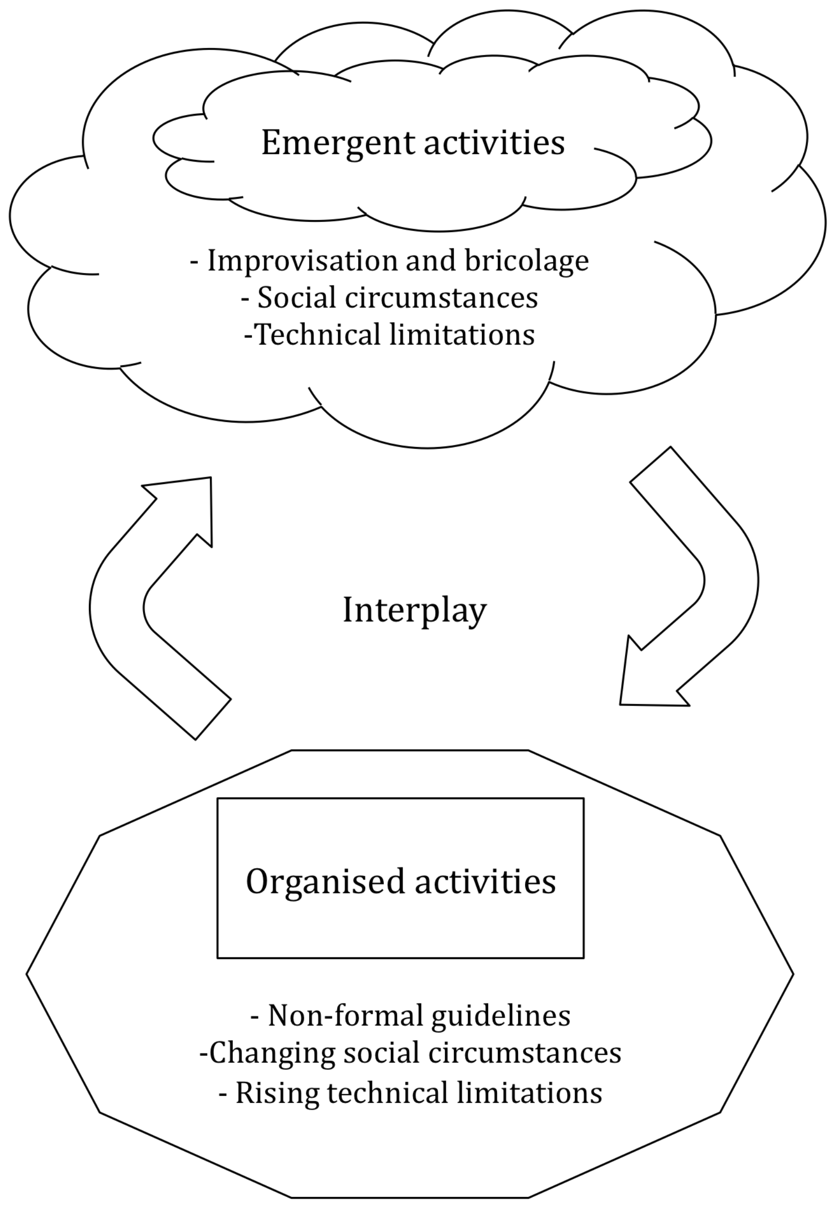

Figure 6-4 Interplay of emergent and organised activities (Molnar & Nandhakumar, 2009)

This finding is supported by other research (e.g.: Prigogine & Stengers, 1984) that complex systems are 'dissipative structures' (self-generating, selfrenewing organising processes) (Akgün, Byrne, Lynn, & Keskin, 2007). At this point, I may elaborate on the phenomenon of this transformation of activities. In this case, the developers and the managers were engaged in the particular context of embedded system development. Although the social context of the developers influenced the activities of developers and managers significantly, their individual capabilities nurtured the reliance upon non-formal standards. This relates to the power vested in the individual developer or manager - Giddens's (1984) 'agency'. The specific behaviour or activities in which humans engage was guided by the rules and resources in which social interactions take place (Giddens, 1984). This capability and power of transforming the social context of the developers might also have been influenced by the social circumstances of EC. However, the individual understanding of developers and managers inspired them too. By doing so, the developers and managers enriched their interactions with more organised activities, although improvisation and bricolage activities were involved as well.

6.2.7 Summary

This section discussed how developers and managers practiced improvisation and bricolage in the context of process management during embedded system development. First, the analysed two patterns of activities (intended and ingenuous activities) were examined and related to the corresponding improvisation paradox. Second, various strengths of the improvisation paradoxes were investigated concerning the distribution on different hierarchy levels at EC. However, the discussion revealed that specific hierarchy levels did not obtain specific strengths of the improvisation paradoxes. In addition, the research findings confirm the tensional arrangement between process management activities and improvisation and bricolage. Third, different situatedness of improvisational and bricolage activities were examined, and the distinct responses of managers and developers were discussed. Fourth, an alteration process of improvisation and bricolage during the development of

Theta was examined. The findings detail the different strengths of the improvisation paradoxes during the initial and progressed phase of the system development. Fifth, the actions and practices during the development of *Theta* involved a funnelling process between structured processes and improvisation and bricolage. The funnelling was influenced by the social circumstances and technical limitations during the development of project *Theta*. Sixth and finally, the interplay of emergent and organised activities was discussed. Emergent activities became organised activities, which helped the developers and managers to cope with changing social circumstances and rising technical limitations. However, when the organised activities lost their ability to cope with the complex embedded system development, organised activities were eliminated and new emergent activities prevailed.

The next section describes the role of reflexivity and human agency in embedded system development practices. First, process management guidelines are examined, how they may support reflexive practices of individuals. Second, reflexive practices of improvisers and bricoleurs are discussed. Third, different consciousness levels of reflexivity are analysed and how they may have influence on structured processes, improvisation and bricolage. Fourth and finally, the dynamics of emergent organisational processes are discussed.

6.3 Capacity to be reflexive in embedded system development

6.3.1 Involved in process management

The literature review concerning process management revealed that many PM frameworks (e.g. ISO 9001, SPICE, and so on) highlight activities, which involve the monitoring of prescribed processes. That monitoring is accomplished through frequent reviews, which are conducted by internal and external examiners, and the case study supports this, because external examiners audited EC according to ISO 9001. Indeed, many PM frameworks require external examination, for the processes to be established according well-recognised management frameworks, such as ISO 9001. The information obtained about those examinations shall be integrated into the efforts to improve the effectiveness of the monitored processes (e.g.: ISO, 2008b; ISO/IEC, 2003). For example, ISO 9001 (ISO, 2008b) suggests that it is the commitment of the management to conduct reviews to continually improve the effectiveness of the quality management system. Decisions that are reached in reviews may be based on information of audit results, customer feedback, process performance and product conformity, status of preventive and corrective actions, and so on. Accordingly, committed decisions may follow corrective or preventive actions. Similar to ISO 9001, other PM frameworks, such as SPICE, suggest also that when planned results are not achieved, correction and corrective actions shall be taken as appropriate (ISO/IEC, 2003). However, the analysis of this case study revealed that corrective actions did not influence the practices of developers. Indeed, developers expressed relief, when the annual assessment was over and the observation notes indicated that they were not at the centre of interest by external ISO 9001 auditors. In addition, it is interesting to note that PM frameworks, such as ISO 9001 and SPICE, tend to commit corrective actions from higher towards lower hierarchy levels. That is reasoned in its approach in which it is the top management's commitment to continually improve the effectiveness of the PM framework and the organisational documents of this case study support this finding.

ISD methodologies prescribe also processes, which involve the review of previous results. For example, the waterfall model, which is a specification-based methodology, suggest the control and monitor testing phase at a very late stage of the development process (Boehm, 1986; Royce, 1970). Evolutionary development methodologies, such as the spiral model, analyse the feasibility of requirements early to overcome potential risks early in the development process (Boehm, 1986; Sommerville, 1996). Furthermore, agile development methodologies emphasise work output as the primary measure of progress (Baskerville & Pries-Heje, 2004; Cockburn & Highsmith, 2001; Manhart & Schneider, 2004). The agile development methodology of Extreme Programming (XP) put pair programming in use, which is a form of continuous review, where two programmers develop software side by side at one computer (Cockburn & Williams, 2001). Thereby, many mistakes get caught as they are being typed in rather than in later development procedures (Cockburn & Williams, 2001). In other words, ISD methodologies want to make sure that that what they advocate for – a structuralised development process of information systems – is conducted properly, by monitoring of organisational activities, similar to many PM frameworks. And that is not surprising, because the whole idea of PM is so fundamentally simple and basic, and it involves nothing more than using common sense (Smith, 2002). However, some ISD methodologies tend to support the capabilities of people at lower hierarchy levels, by advocating practices (e.g. pair programming) that do not involve a decision making process with other hierarchy levels.

Looking through the theoretical lens it appears to be that the recognised phenomena are nothing else than the capacity to be reflexive of those who follow process management or ISD methodology guidelines. The literature and the findings of this case study suggest that monitoring activities, questionnaires, reviews, audit results, testing phases, risk analysis, and perceiving work output as the primary measure of progress support reflexive monitoring of action, by helping through prescribed guidelines. Although the flow of practices presumes reflexivity that is not necessarily prescribed (Giddens, 1984), advocates of PM frameworks and ISD methodologies claim that an accomplished process standardisation avoids the vagueness in daily work and therefore it is an advantage for managing an organisation (e.g. Rada, 1996). The analysis of the case study supports this, because it was revealed that provided help for establishing and maintaining structured processes was one motivation for EC to receive ISO 9001 certification. Besides this motivation to establish and maintain processes that prescribe reflexive behaviour, there are also motives that come from outside the organisation. For example, better proximity to customers is supported through customer feedback, hence the organisation may base corrective actions upon those information. In addition, continuous record keeping of attributes during the development process may help to minimise liability risks. And the findings of this research support this, because the analysis disclosed that an external motivation for EC to receive ISO 9001 certification was the customers' assumption of stable quality of products and services. So, it is comprehensible that many organisations strive to establish and maintain practices, which should be intrinsic in all organisational actors.

However, the difficulties and limitations with prescribed processes remain, because research literature indicates about the practice of prescribed processes (e.g.: PM frameworks or ISD methodologies) that practitioners do not fully accept proposed processes (Fitzgerald, 1998; Nandhakumar & Avison, 1999). The findings of this research support this, because the developers and managers at EC stated that non-formal standards were more valuable as methods to follow routinely than prescribed processes by their organisational ISO 9001 standards. The analysis of this research supports also that developers and managers thought the ISO 9001 was not appropriate enough to cope with unpredictable events and other conditional factors that occur in turbulent environments, such as embedded system development (du Plooy, 2002). So, either the environment is too complex and turbulent, which cannot be changed, or the organisational framework is too limited.

Nevertheless, the capacity to be reflexive that is supported through prescribed processes, may be enhanced by different approaches. Some PM frameworks tend to focus on top managements' commitment to enforce reflexive behaviour throughout the organisation, such as ISO 9001 and SPICE. Hence, it remains the responsibility of managers to impose processes that imply reflexive behaviour (arrows 'prescribe reflexive behaviour' in Figure 6-5). The ability to act (probably differently than expected) remains in every individual. However, higher hierarchy levels in non-agile settings involve higher potencies to deploy on others than lower hierarchy levels (thicker arrows in Figure 6-5). Lower hierarchy levels may have only limited influence on others in non-agile settings (thinner arrow in Figure 6-5). In addition, some agile ISD methodologies emphasise the developers' responsibility to continually reflect on their actions and practices. However, that does not mean that managers at higher hierarchy levels are not committed to the actions and practices of their staff, because somebody must have decided to work according to agile methods. It appears to be that the practice of agile methodologies implies a shift in the capacity to involve reflexive behaviour in day-to-day work and to act appropriately without further consultations. Therefore, in contrast to the capacity to act and to deploy on others in non-agile settings that involve different potencies, the capacity to act in agile settings is comparable (double arrow in Figure 6-5). Figure 6-5 summarises this finding and depicts the different approaches, how prescribed processes may support the capacity to be reflexive and how this involves the capacity to act appropriately.

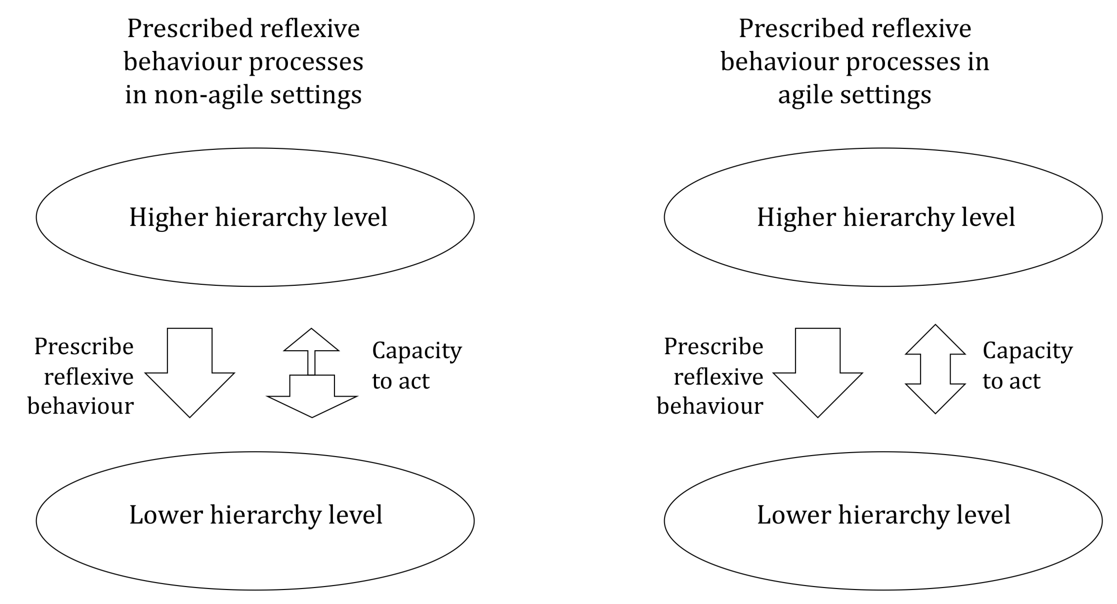

Figure 6-5 Prescribed reflexive behaviour and the capacity to act in non-agile and agile settings

6.3.2 Utilised in improvisational and bricolage activities

The comprehension of improvisation and bricolage is complex and the various paradox pairs indicate the different nuances. Moreover, the literature describes factors that influence improvisation and bricolage, such as uncertainty (Ciborra, 1996) and environmental turbulences (Moorman & Miner, 1998b). Certainly, people have different choices, how they might cope with those challenges. For example, they can ignore changes and continue as planned, they may try to cope with changes with iterative planning cycles, or they can improvise and perform bricolage by merging the planning and execution process (Moorman & Miner, 1998b). The analysis of the case study revealed that uncertainty (e.g.: ambiguous requirements) and environmental turbulences (e.g.: fluctuating requirements) influenced the activities of developers and managers. Different improvisational and bricolage practices were analysed in this case study and some of them involved different characteristics of creativity and pragmatism. Another influencing factor of improvisational and bricolage activities may involve task complexity (Hutchins, 1995; Weick & Roberts, 1993), because some tasks are difficult to pre-plan in a rational way. For example, the analysis disclosed that the accomplished reverse engineering of the feature technology was considered as a breakthrough. Hence, the task complexity to reverse engineer this feature was very high and it was successfully completed through improvisation and bricolage.

Besides the different nuances of creativity and pragmatism, oriented drift is another paradox pair of improvisational and bricolage activities. For example, Dehlins' (2008) work emphasises positive improvisation as chosen voluntariness and this involves a certain orientation, whereas negative improvisation involves vagueness and unpredictability (Dehlin, 2008). The findings of this research described improvisational and bricolage activities that involve self-containedness and therefore a certain voluntary characteristic of developers and managers. Although preconditions involved ambiguous requirements, the result of those activities was a functioning prototype of *Theta*. In addition, Dybå (2000) describes how improvisers and bricoleurs may utilise knowledge to have success. But knowledge also involves experiences about the behaviour of the item at hand in various circumstances. Improvisers and bricoleurs tend also to search for new knowledge and explore new solutions (Dybå, 2000). The analysis revealed that the developers and managers utilised many known software aids, such as Bugzilla and Subversion, and explored new solutions through reverse engineering.

Looking through the theoretical lens it appears to be that improvisational and bricolage activities involve the capacity to be reflexive. For example, uncertainty and environmental turbulences require actors that reflexively monitor their surroundings, in order to cope with difficulties adequately. If actors improvise and perform bricolage, that means they merge the planning and execution process (Moorman & Miner, 1998b), actors must reflexively monitor continuously and intensively, because errors would result immediately. Proceeding without errors result in encouragement to continue with improvisation and bricolage activities, whereas defective results tend to encourage improvisers and bricoleurs to re-think their current actions and practices (Weick, 1995). Particular complex tasks require high potencies of reflexive capabilities, because the development of compound systems such as embedded systems, require knowledge and experience from different realms (hardware and software). Changes at one piece of the hardware, may have consequences on hardware and software. Therefore, continuous reflexivity of actions concerning the development is needed in order to be successful. If actors perceive indications that the success may be threatened, purposive actors want to change this and induce means to cope with the difficulties. In addition, the functioning prototype of *Theta* showed that reflexivity was utilised in improvisational and bricolage activities of developers and managers at EC.

Although improvisation and bricolage were practiced throughout the organisation, there was less indication found that higher hierarchy levels were able to influence reflexive behaviours of the improvisational and bricolage activities of lower hierarchy levels (cross in Figure 6-6). If the time gap between planning and execution becomes void, it is rather difficult to influence reflexive behaviours. And the analysis of this case study support this, because there was less indication found that managers were directly involved in how developers should monitor the ongoing activities at project *Theta*. However, that does not mean that managers at higher hierarchy levels were not able to act appropriately during the development, because at critical issues, it remained the responsibility of Michael (senior manager) to guide the development. This guidance during the improvisational and bricolage activities relates to the paradox pair of oriented drift, by offering some direction whilst tolerating a degree of flexibility. Therefore, hierarchical levels do not become obsolete if improvisation and bricolage occurs, rather they may provide the potential to influence the direction and to undertake necessary actions appropriately (double arrow in Figure 6-6). Figure 6-6 gives a summary of this finding and depicts that higher hierarchy levels have limited influence to prescribe reflexive behaviours of improvisers and bricoleurs, whereas managers and developers may have the capacity to act as such.

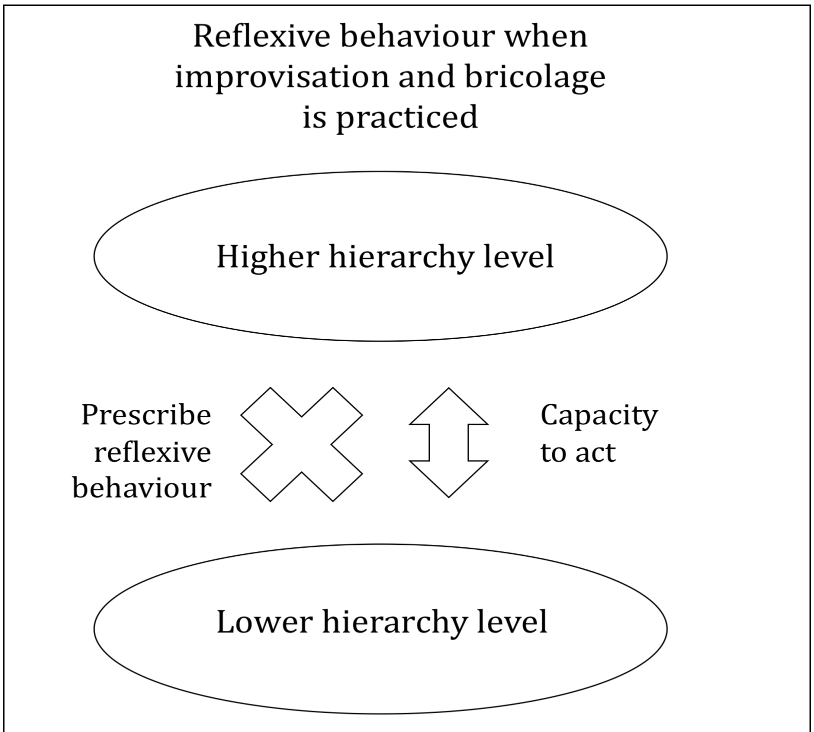

Figure 6-6 Agency and reflexive behaviour when improvisation and bricolage is practiced

6.3.3 Different consciousness levels of reflexivity

The evidence seems to suggest that the influence of reflexive practices was significant for developers and managers to decide what actions to do next. Previous experiences by developers and managers enhanced the introduction and facilitation of non-formal structures such as the change-control-board meeting and software tools like Subversion and Bugzilla. Although these aids provided sufficient comfort to the developers and managers, reflexive monitoring of activities is a continuing feature of everyday action and involves the doings not just of the individual but also of others (Giddens, 1984). Furthermore, Giddens (1984) does not see actions as isolated phenomena but sees them as a flow of events, with agency taking place with the different levels of consciousness. The conventional character and necessity of these aids (change-control-board, Subversion, Bugzilla) did not call for an alteration of these non-formal standards. Examination of processes by developers and managers seemed to happen at a level of discursive and practical consciousness, because using these aids was not prescribed and emerged over time in the organisation.

The case also revealed that many actions were based on improvisation and bricolage. The previous section suggests that improvisation and bricolage entails reflexive behaviours, because in the moment of situated actions, purposive agents monitor their activities. This ability to continually reflect upon work results was an underlying behaviour of developers and managers. The interviewees had difficulties to describe their working style explicitly; however, the observation indicated a continuous mode of reflexive self-estimation regarding the potential impacts and side effects of one's own work. Giddens (1984) states that "human agents always know what they are doing on the level of discursive consciousness under some description" (p. 26). Therefore, improvisational and bricolage activities are practiced mainly in the discursive consciousness level. For example, Raphael described his work style as "autodidact with some experience and passion" and "somewhere between creative chaos and structure." However, when he was challenged to describe his opinion of certain actual situations he described explicitly the technical determining factors and he suggested approaches to master challenges. In addition, his success in working for project *Theta* provides evidence that he knew what was he doing, hence his improvisational and bricolage activities were practiced mainly on a discursive consciousness level. In addition to findings of Berliner (1994) and Weick (1998) who allude the role of practical consciousness to improvisation, this research highlights that improvisation and bricolage in the context of embedded software design occurs mainly on a discursive consciousness level. Moreover, this ability to reflect upon one's own behaviour as well as that of group members was an important practice for the success of the embedded system development. For example, part of the developers' everyday actions and practices was to spend time on evaluating potential side effects; previous experiences of developers led them to introduce the software tools Subversion and Bugzilla. Furthermore, non-formal structures (e.g., change-control-board meeting) were the result of improvisation and bricolage. Giddens (1984) claims that reflexivity should be understood not just as self-consciousness, but as the monitoring character of ongoing everyday actions. To be a human being is to be a purposive agent who both has reasons for his or her activities and is able, if asked, to elaborate discursively upon those reasons, including lying about them (Giddens, 1984).

In summary, formal structured process are practiced mainly on the practical consciousness level, meaning that processes are characteristically simply done and can be followed blindly. Non-formal structured processes involve reflexivity to a great degree that is practised in discursive and practical consciousness whereas improvisational and bricolage activities involve reflexive behaviour that is practiced mainly in the discursive consciousness level (Figure 6-7).

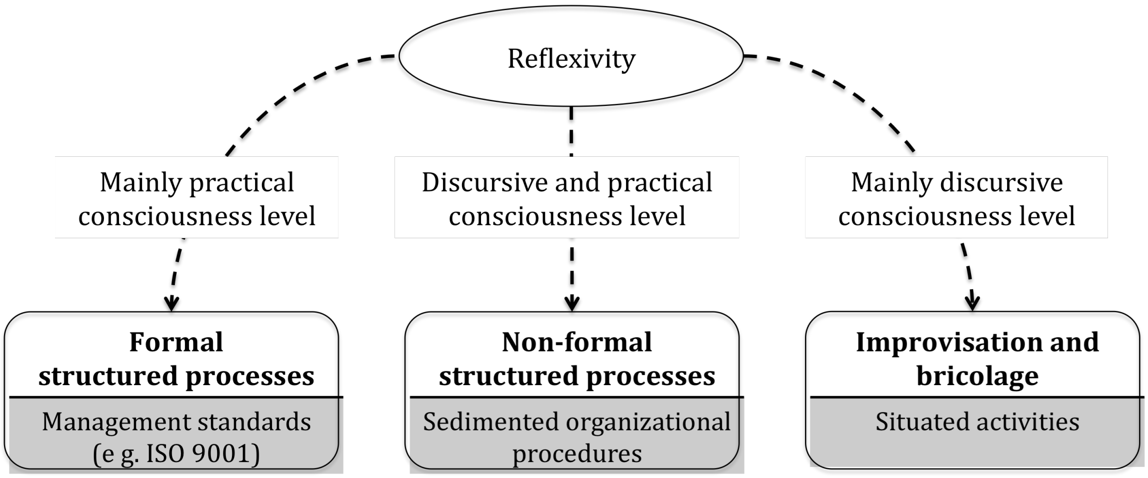

Figure 6-7 Different consciousness levels of reflexivity when performing formal structured processes, non-formal structured processes, or improvisation and bricolage

6.3.4 Dynamics of emergent organisational processes

It was also evident from the analysis that organisational processes emerged over time. During the initial phase of *Theta*, improvisation and bricolage were strongly practised to cope with technological uncertainties, but the project manager tried to set limits to the seemingly chaotic approach. The involvement of improvisational and bricolage actions was a common behaviour during development, although these situated activities appeared as lacking in concision to outsiders. Although these processes were improvisational and bricolage actions, project members agreed in their usage, because other developers and managers followed them over time too. This finding supports previous findings by Orlikowski (1995) concerning situated change when organisations change their practices. However, contrary to Orlikowski's (1995) findings, this research revealed that organisational processes may emerge, which involve more sophisticated planning than previous processes, because later in the development process, some of the previous improvised processes became structured processes as well as organisational standards. As previously analysed, those non-formal processes were important for the project management, and helped the developers and manager to cope with difficulties during the development.

The emergence of structured processes and organisational standards from improvisational and bricolage activities confirms that the influence of ISO 9001 was significantly less important than non-formal procedures such as various processes based on software tools or weekly meetings. However, some practices of research participants were part of their daily activities and their interaction in a group. During continuous repetition of similar actions, they became part of the cultural sediment of that organisation. These actions, which became routine practices of organisational members, reflected the organisational portfolio of best practices. As a consequence, this organisational portfolio of best practices may influence future modifications of the organisational ISO 9001. It is useful to know that the structural framework of ISO 9001 and other process management frameworks were the result of a collection of previous best practices by many experienced people (Fowler & Highsmith, 2001). In addition, common sense influenced the sedimentation process of reproduced improvisational and bricolage actions and practices and over time became structured processes. The analysis revealed that some research participants drew on their common sense (without knowing it explicitly) and acted upon this in their everyday work life; in other words, they draw on their reflexive capabilities. This reflexivity needed no structural setting to be followed, just an individual with an open attitude and the capabilities to do what was necessary. Some procedures were established through individual insight and the sheer need for it (e. g. change-control-board meeting).

Consequently, sedimented activities become structured processes and may influence and shape new additional dynamics of improvisational and bricolage actions and practices. New additional dynamics means that actors move away from previous processes, because ongoing monitoring activities as part of the everyday actions reveal an odd situation. That is, agents review organisational structured processes constantly, as part of the ongoing flow of organisational life. However, actors not only continually monitor the flow of their activities and expect others to do the same for their own, they also routinely monitor aspects, social and physical, of the context in which they operate (Giddens, 1984). In addition, the routines supported the confidence within the developers and managers at EC that what they are doing is right for the development of *Theta*. Giddens (1984) describes this confidence as ontological security. Moreover, reflexive monitoring is ongoing infinitely by the agent and may cause unintended consequences. If routines are breached, change occurs and may excuse improvisation and bricolage. Breaking routines may decompose organisational structured processes. The decomposition of structured processes may therefore introduce improvisation and bricolage as a response. For example, a previous code of practice was to wait for Michael to make decisions about the embedded system development, but the introduction of the change-control-board meeting (founded on a whim) breached the previous practice. Therefore, the developers and managers' ability to be reflexive has a profound influence on maintaining or avoiding structured processes in an embedded system development setting.

Figure 6-8 depicts the insights gained of how improvisational and bricolage activities mature and establish in a pool of reproduced, similar improvisational and bricolage actions and practices. Over time, those activities become part of the organisational sediment of structured processes. Additional dynamics may cause to breach structured processes and give rise to other improvisational and bricolage activities.

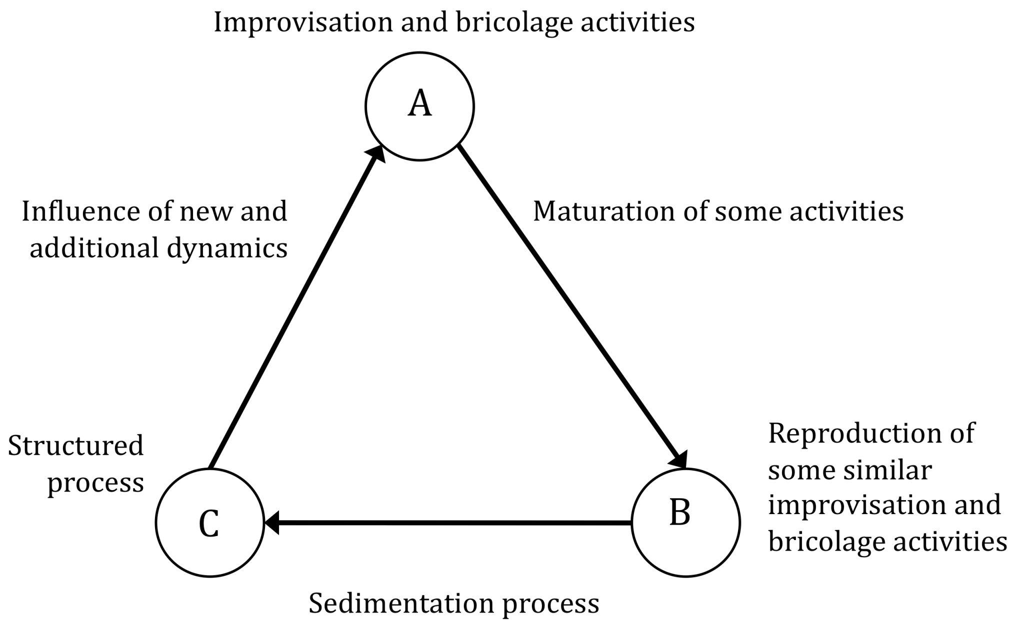

Figure 6-8 Dynamics of the emergent organisational structured processes

6.3.5 Summary

This section explored the role of reflexivity and human agency in the practice of embedded system development. First, PM frameworks and ISD methodologies were investigated concerning reflexive practices. The investigation discovered different approaches, how reflexive behaviours are supported through PM frameworks and ISD methodologies. Second, the capacity to be reflexive in improvisational and bricolage activities was examined, and it was discovered that higher hierarchy levels have limited capacity to influence reflexive behaviours of improvisers and bricoleurs. Third, different consciousness levels of reflexivity were revealed and in sum, formal structured processes involve reflexivity largely embedded in the practical consciousness of the actors. Non-formal structured process involves reflexivity that is embedded in both discursive and practical levels of consciousness. Whereas improvisational and bricolage activities mainly involve reflexivity at the discursive consciousness level. Fourth and finally, the dynamics of the emergent organisational structured processes were examined. Mature improvisational and bricolage activities were reproduced by organisational actors. Over time, they become part of the organisational structured processes. However, additional dynamics may influence those structured processes and give raise to new improvisational and bricolage activities.

The next section presents a conceptualisation of the findings that help to answer the second research question.

6.4 Conceptual model

The previous section revealed how improvisers and bricoleurs reflexively monitor embedded system development activities without explicitly knowing. In addition, the role of reflexive practices within process management frameworks was analysed. The discussion revealed that the two contradicting approaches, following prescribed processes and improvisation and bricolage, have vital concepts of social life in common. It appears that to overcome the tensional arrangement between those contradicting approaches, it is worthwhile considering strengthening what they have in common: the capacity to be reflexive. That means if the people in the context of embedded system development strengthen their ability to perform reflexive practices within this domain, this could balance the tension between process management and improvisation and bricolage.

The capacity to be reflexive is involved in process management (cf. section 6.3.1) and improvisational and bricolage activities (cf. section 6.3.2). Certainly, the two different approaches include a different understanding, how reflexive behaviours shape the ongoing activities. Nevertheless, the capacity to be reflexive is for both approaches fundamental, because it is either prescribed (e.g.: process management frameworks, ISD methodologies) or intrinsically required (improvisation and bricolage) in order to be successful. Giddens (1984) states that reflexivity is "grounded in the continuous monitoring of action which human beings display and expect others to display" (p. 3). The capability of doing things (e.g. being reflexive) refers to agency, so it relates to the power of individuals to act differently at any phase in a given sequence of conduct (Giddens, 1984). Therefore, the capacity to perform reflexive practices is part of the ongoing activities of people. However, the conceptual model put emphasis on the reflexive practices that are related to the domain of embedded system development and states that a strengthening of the capacity to be reflexive could balance the tensional relationship between process management and improvisation and bricolage (Figure 6-9). The double arrows between the capacity to be reflexive, and the two contradicting approaches indicate that both approaches involve reflexive practices (cf. sections 6.3.1 and 6.3.2). In addition, the capacity to be reflexive may influence process management or improvisational and bricolage activities, if the respective situation is required to breach the ongoing practices (cf. sections 6.2.6 and 6.3.4).

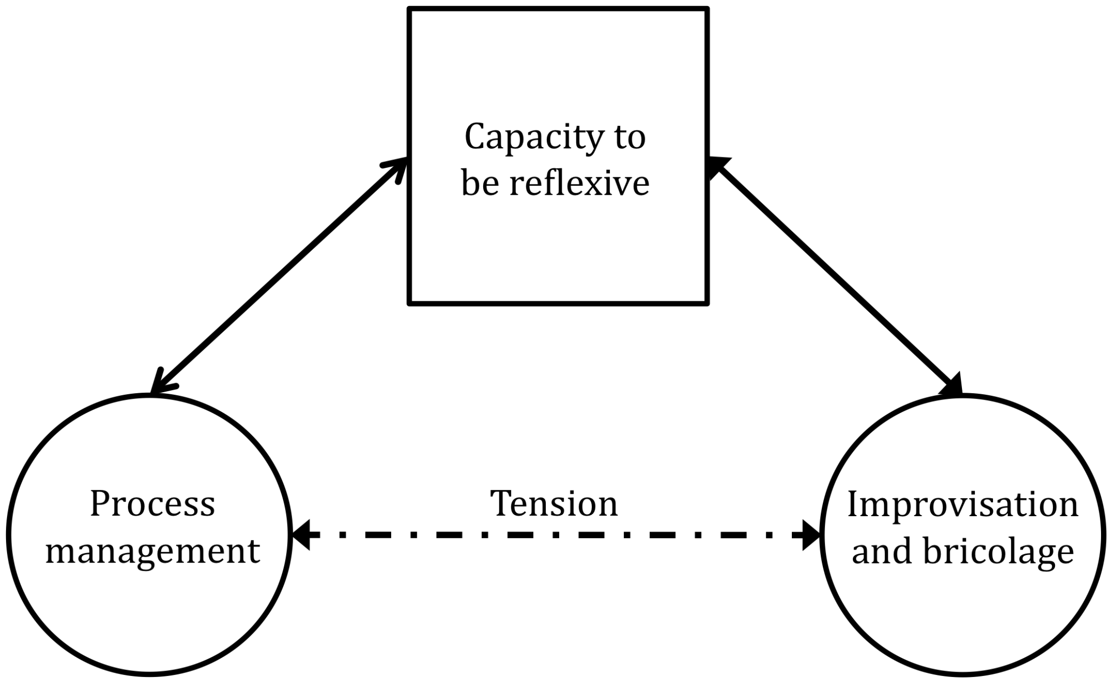

Figure 6-9 Capacity to be reflexive and the tensional arrangement between process management and improvisation and bricolage activities

The previous findings in section 6.3.3 indicate that formal structured processes, which might be prescribed by process management frameworks such as ISO 9001, involve reflexive behaviours that are practiced mainly on the practical consciousness levels, because they follow these routinely. The practical consciousness level is what actors know about conditions, including especially the conditions of their own action, however cannot express discursively (Giddens, 1984). Therefore, the prescribed processes are in the main simply done, without an awareness that has a discursive form. Contrary, improvisational and bricolage activities are practiced mainly in the discursive consciousness level, that means the actors are able to give verbal expression to the conditions of their own action (cf. Giddens, 1984). One procedure to balance the tension between the different approaches necessitates a change of the reflexive practices that predominantly involve the ability of actors to give discursive expressions of the conditions of their own action. Therefore, reflexive practices should be practiced more on a practical consciousness level, so that actors know tacitly about how to continue in the context of development activities. Giddens (1984) state that the division between practical and discursive consciousness can be altered by many aspects of the agent's socialisation and learning experiences. Moreover, the findings of this research support this explanation, because section 6.3.3 revealed that non-formal structured processes were practiced on the discursive and practical consciousness level. Indeed, those non-formal structured processes originated from improvisational and bricolage activities, when developers and managers were able to describe explicitly the conditions of their actions (discursive consciousness). Some of these activities matured and became continuous reproduced over time by developers and managers (cf. section 6.3.4). The reproduced activities became part of the organisational routines that involved a taken-for-granted character of the specific activities and the developers and managers had difficulties to express discursively the conditions of their own actions (practical consciousness). Although those routinely practiced activities maintained its non-formal character, they became relatively structured. For example, the change-control-board meeting became a weekly appointment with agenda; software tools like Subversion and Bugzilla became common aids for software development. In sum, the ability of non-formal structured processes to balance the tension between process management frameworks and improvisational and bricolage activities is important and advantageous.

In addition, reflexive practices that become more routinised support the ontological security of organisational actors, because routines provide confidence within the developers and managers. This was evident for example, where some practices of research participants became routine practices of organisational members. Research participants expressed positive attributes and they felt safe concerning those non-formal structured processes, such as the change-control-board meeting. Interruptions occurred when odd situations aroused through new dynamics and actors breached routines and change occurred. However, this moving away from previous processes may be useful, and the findings of this case study support that argument, because when the lag to wait for Michael (senior manager) to make decisions about the development activities was too long, the change-control-board meeting was founded almost by accident and reasoned to become a new routine.

Likewise, a common understanding of all hierarchy levels concerning the need for reflexive behaviours helps to strengthen the capacity to be reflexive. Reflexive practices that become part of the organisational standards obtain the potency to change into "knowledge of 'how to go on' in forms of life" (Giddens, 1984). Giddens (1984) describes these stocks of knowledge as mutual knowledge that is mostly incorporated in actors' practical consciousness level. That common knowledge is inherently in the capability to go on within the routines of social life. Certainly, fundamental behaviours of reflexivity are already part of the common knowledge, but this argument focuses on reflexive practices in the context of embedded system development activities. If they become part of the common knowledge in an organisation, it will make obsolete the need to prescribe reflexive behaviours through formal structured processes (e.g. ISO 9001). However, formal structured processes help to lift the common knowledge in an organisation to a certain level. Hence, they may be utilised to jump-start the fundamental behaviours of reflexivity in order to make known what 'common sense' activities involve (e.g. Smith (2002) describes the idea of process management as only using common sense). Moreover, the findings of this research support this argument, because managers and developers practiced improvisation and bricolage, which involved contextual reflexive behaviours, although the organisation maintained a process management framework. For example, Martin (project-manager) initiated the change-control-board meeting on a whim, when the previous procedure was not sufficient to cope with the difficulties and some developers introduced development aids (Subversion and Bugzilla) to benefit from the capabilities of those tools. Therefore, the various hierarchy levels shared common knowledge concerning the situated needs within the context of embedded system development activities.

In sum, the conceptualisation of this research is threefold and describes what measures may strengthen the capacity to be reflexive:

a) Reflexive practices may be practiced mainly on the practical consciousness level such that actors continue to be reflexive in the context of development activities, without really thinking about it.
b) Reflexive practices may be routinised to support the ontological security of developers and managers.
c) Finally, reflexive practices may become part of the common knowledge within the organisational actors that all hierarchy levels share a similar understanding of contextual reflexive behaviours.

It appears that these measures can strengthen the capacity to be reflexive and the empirical evidence of this research supports this conclusion.

6.5 Summary

This chapter has provided a discussion of the improvisational and bricolage activities and the role of reflexivity and human agency in the practice of embedded system development. The analysis findings were examined and corresponded to the literature of the theoretical lens. In sum, section 6.2 describes how developers and managers practice improvisation and bricolage in the context of process management during embedded software design (cf. research question 1). Section 6.3 provides details about the role of being reflexive in embedded system development. Drawing on those findings, a conceptualisation aims to describe what measures help to balance the tensional arrangement between process management and improvisation and bricolage activities (section 6.4) (cf. research question 2).

How those findings are relevant to theory as well as practice is described in the next chapter.

7 Implications and conclusions

7.1 Introduction

The previous chapter explored improvisation and bricolage during development practices and the role of reflexive capabilities therein. In addition, it presented a conceptualisation that describes what measures help to address the difficulties between prescribed and emergent processes. These findings have implications for theories and models, which can benefit from the contributions of this research work (section 7.2). Furthermore, the obtained theoretical insights generated have implications for the practice described in section 7.3. Section 7.4 describes opportunities for future research that can be linked to the results of this thesis. Finally, section 7.5 ends with concluding remarks concerning this research work.

7.2 Implications for theories and models

7.2.1 New paradox pair of improvisation and bricolage

The previous chapter presented findings concerning how developers and managers practice improvisation and bricolage. The setting of those findings involved the development of an embedded system in an organisation, which had a certified process management framework established and maintained (ISO 9001). The findings describe how embedded software is developed in practice; hence, the findings try to address the knowledge gap. Moreover, how developers and managers practice improvisation and bricolage seems to reveal a new paradox pair of improvisation and bricolage, what I call '*progressive saturation*'. That means, improvisation and bricolage involves a process that is varyingly positioned throughout the activities that include system development. The proposed paradoxes of improvisation (Table 3-4) that were used as a theoretical lens in order to illuminate the field data that involved procedures that were difficult to describe, control and manage, or were situated in emergent contexts, comprehended this aspect not in its full flavour. Therefore, a new paradox pair of improvisation and bricolage may be useful, to describe improvisation and bricolage better in environments that involve development activities. So, the implied new paradox pair of progressive saturation intends to complete the prevalent paradoxes and highlight the phenomena of improvisation and bricolage from a slightly new angle. This new angle involves a temporal perspective of improvisational and bricolage activities. Previously proposed paradoxes of improvisation and bricolage withhold the attribute of temporal alteration. For that reason, the new paradox pair of progressive saturation contributes to the knowledge about improvisational and bricolage activities, so that future research may describe those activities more comprehensively.

The new paradox pair of progressive saturation relates to the theoretical constructs described in the discussion. For example, ingenuous activities involve certainly a very innovative surrounding that provides space for progressive actions. Contrary, intended improvisational and bricolage activities include an environment that is about to be saturated with unplanned improvisation and bricolage (cf. section 6.2.1). Moreover, improvisational and bricolage activities that are expected to occur involve many progressive activities, because the people had enough opportunities to develop self-contained. Whereas, improvisation and bricolage activities that needs to occur suddenly involve less progressiveness but a setting that is saturated and conditioned. Therefore, improvisation and bricolage activities are shaped differently to activities that are 'situated expectable' (cf. section 6.2.3). In addition, during the initial phase of a development project, improvisational and bricolage activities involve high progressiveness, contrary to later phases of a development project, when social circumstances and technical limitations influence the development (cf. sections 6.2.4 and 6.2.5). Finally, emergent activities involve also higher progressiveness as activities that are organised. The analysis revealed that previous emergent activities matured to organised activities that were involved by changing social circumstances and rising technical limitations. Therefore, the development project involved a certain saturation of improvisational and bricolage activities (cf. section 6.2.6). Table 7-1 describes the implicated new paradox pair of improvisation and bricolage and relates it to its theoretical constructs that are empirically based upon the findings of this research.

Table 7-1 New paradox of improvisation and bricolage

Improvisation and bricolage paradox	**Related theoretical constructs**	**Key reference**
Progressive saturation	Intended and ingenuous activities	Section 6.2.1
	Situatedness	Section 6.2.3
	Alteration process	Section 6.2.4
	Funnelling process	Section 6.2.5
	Unfolding activities	Section 6.2.6

7.2.2 Capability to improvise and perform bricolage

The findings of this research indicate that the capability to improve and perform bricolage is not evenly spread within the organisation. Moreover, the findings of this research supports Kirton's (1976) findings; that people involve different styles of creativity and that some individuals possess higher originality than others. In addition, the analysis demonstrated that improvisation and bricolage activities involve creativity in various forms. Therefore, this research implies that the capability to improvise and perform bricolage is differently developed among individuals. Moreover, this is not necessarily surprising, because the capability to improvise and perform bricolage is shaped by the different knowledge and experiences that individuals collect over time. In addition, the research findings confirm this implication, because an actor like Raphael (perceived role as fire fighter in the context of embedded system development during project *Theta*) possessed with his abilities significant influence in the development of *Theta*, whereas other developers had less influence in the development activities.

7.2.3 Capability to be reflexive in development activities

The conceptualisation of this thesis highlights the importance of reflexive capabilities in development activities. It describes three measures that may strengthen the capacity to be reflexive. First, reflexive practices should be practiced mainly on the practical consciousness level; second, reflexive practices should be routinised; third and finally, reflexive practices should become part of the common knowledge within the organisational setting. Therefore, this insight may help to address the knowledge gap and aids as theoretical conceptualisation that can find relevance in academic courses, professional forums, and consulting practice. For example, it can find relevance in academic courses that provide a comprehension of ISD as social activities (Truex, et al., 1999). In addition, those findings may find relevance in professional forums[1], in order to make it more public how ISD is practiced and because there are only few studies that focus on the process of ISD in its real organisational context (Fitzgerald, 1997; Nandhakumar & Avison, 1999). Furthermore, consultants may benefit from the findings of this research,

---

1 Part of this research project was the publication of one paper at the proceedings of the international SPICE conference in 2008, and professionals mainly frequented this conference.

because the conceptualisation may influence what they advice their customers in development settings.

7.2.4 Contributions to ISD knowledge

In addition to the conceptualisation that may find relevance in various parties, the findings of this research help to contribute to the knowledge gaps identified. This thesis exhibits the undivided comprehension of the findings concerning improvisational and bricolage activities of developers and managers in the context of embedded system development. However, those findings may also find relevance in the ISD context and not only in the sub-domain of embedded system development. In particular, chapter six discusses the research findings, answers the research questions, and relates findings to previous research discussed in the literature review. The findings of this case study describe tendencies in the domain of ISD with the help of the rich insights that were discovered in the sub-domain of embedded system development, For example, the analysis revealed intended and ingenuous improvisational and bricolage activities and how they correspond to improvisation paradoxes found in the literature. The research disclosed different situatedness of improvisation and bricolage and related it to improvisation paradox pairs, such as oriented drifting (Miner, et al., 2001). Findings concerning the alteration and funnelling process of improvisation and bricolage highlight the different nuances of improvisation and bricolage during a development process and support the new paradox pair what I call '*progressive saturation*'. In addition, the interplay of emergent and organised activities describes unfolding activities, which become continually reshaped within a complex setting. Those findings may help to describe improvisation and bricolage in similar environments better and provide a valuable contribution to the knowledge of ISD in practice. Furthermore, the findings are based upon the empirical data of this research.

In addition, findings concerning the role of reflexivity and human agency in the practice of embedded system development provide new insights into the ISD in practice. The results demonstrated how those two vital aspects of social life are capable of influencing development activities of a complex system. In addition to research that studied the development of complex systems (e.g. Curtis, et al., 1988), this study implies that the capability to be reflexive is influenced by the processes that are practiced by managers and developers. Process management frameworks and ISD methodologies may prescribe processes that support reflexive practices. However, the findings of this research support the insight that practitioners do not fully accept proposed processes (Fitzgerald, 1998; Nandhakumar & Avison, 1999). Moreover, improvisers and bricoleurs also involve reflexive capabilities that are not necessarily prescribed. These people need to act reflexively, because they are confronted with an uncertain and turbulent environment; hence they merge the planning and execution process (Ciborra, 1996; Moorman & Miner, 1998b). Finally, it depends on whether individuals follow prescribed processes or improvise and perform bricolage, to what degree they make use of their capacity to be reflexive in development activities.

Moreover, the analysis results about the different consciousness levels of reflexive practices indicate the different strengths of formal structured processes and improvisational and bricolage activities. Formal structured processes are practiced mainly on the practical consciousness level and implicate that practitioners need to rehearse those managed processes well enough that they are characteristically simply done. On the contrary, improvisational and bricolage activities involve reflexive behaviour that is practiced mainly in the discursive consciousness level and imply individuals, who continually monitor their environment that they can react quickly enough to cope with turbulences. In addition, non-formal structured processes are practiced on the discursive and practical consciousness level and implicate that those processes are situated enough to become sediment of organisational procedures.

Furthermore, findings about the dynamics of the emergent organisational structured processes implicate that organisations are in the state of being in continual process, never arriving but always in change (Truex, et al., 1999). So, structured processes that are part of the organisational sediment of processes in practice remain in a state in which new and additional dynamics can influence them. Those new dynamics give rise to improvisational and bricolage activities and means that actors try to cope with the dynamics. If those coping activities are reproduced, they involve the likely possibility to become sediment of the organisational processes. Hence, a new cycle starts, because new dynamics may influence the established processes.

7.2.5 Contributions to means of studying complex ISD processes

The data collection and presentation approach of this research involved the collection of empirical data through interviews, observation, and analysis of published data and organisational documents. For a comprehensive presentation of the case study, I have formulated a narrative that describes significant events, by emphasising the meanings that participants gave to various incidents. Consequently, the case study contained enough description of context to understand the nature of the incidents. The various findings that were analysed in chapters 5 and 6 implicate that this approach to collect and present data were helpful, and they call for a focus on the complex and emergent presence of everyday life. Therefore, future studies that investigate actions and practices of individuals in similar environments could benefit, by using the same data collection and presentation approach as this research utilised. Particularly the study of improvisational and bricolage activities that occur in real situations require a data collection approach, which involves the timely collection of rich data. Cunha (2005) points out that "it is meaningless to study bricolage ex post facto, due to retrospective justification and to the actors' reconstitution of previous organisational circumstances" (p. 22).

In addition, the application of theoretical frameworks as "theoretical lenses" appeared to be helpful to analyse the case study. Indeed, this thesis applied three analytical topics that emerged from the case, in order to analyse the empirical data sufficiently. The first analytical topic of structured processes emerged from the environmental setting of this research in an ISO 9001-certified organisation. However, this analytical topic was unable to illuminate field data of procedures that were difficult to describe, control and manage, or were situated in emergent contexts. Therefore, the second topic of improvisation and bricolage was used to overcome these shortcomings. With those two topics, the analysis provided rich descriptions, but was lacking in descriptions of human actions and the motives of individuals. Consequently, applying concepts of reflexivity and human agency helped to explain human actions and motives of individuals and thus make those essential aspects of social life understandable. This approach to illuminate the field data with different analytical topics implies that it may be required to analyse complex settings by applying multiple theoretical lenses. Therefore, this approach may help to overcome shortcomings of previous analytical topics and support a comprehensive analysis of the case study.

7.3 Implications for practice

The findings of this research have also some implications for practitioners. For example, the findings of this case study revealed that improvisational and bricolage activities can happen in an environment that is frequently assessed to fulfil tight process management frameworks, such as ISO 9001. The prescribed processes of the PM framework were practised leniently, so that developers were not hassled too much in their daily tasks. The organisational vision, development strategy, and overall aids, such as software tools, seemed to support the developers and managers to improvise and perform bricolage. Only necessary prescribed guidelines were followed shallowly in the development area, in order to endure annual PM framework assessments. This means for practitioners, that they may use process management frameworks as a scaffolding (section 7.3.1), and improvisation and bricolage in the daily work of system development. Indeed, the process management framework was used to serve the necessities for marketing the products and services of the organisation. The developers and managers followed the PM framework just enough to sustain the certification process, but let enough room to practice improvisation and bricolage.

7.3.1 Process management frameworks as scaffolding

Just because the PM framework was used only as scaffolding indicates that there are severe limits in the practical application of it. The analysis revealed that some prescribed processes involved a strong emphasis on various requirements, methods, and results, so that on the end nobody followed the process at all. This indication supports du Plooys' (2002) findings that process management framework and ISD methodologies generate an overemphasis on tools, techniques, and methodologies. And this overemphasis set limits to contextual factors such as social circumstances and technical limitations. Therefore, prescribed PM frameworks can seldom resolve complex difficulties. Moreover, practical intelligence is not about straight, rigid, and univocal knowledge embedded in prescribed processes, but seeing beyond them even if this seems oblique at the time (Ciborra, 2002). Conversely, following the PM frameworks successfully implicates that the organisation is very narrowed, without contextual factors such as turbulences, unpredictable events, and other conditional factors (du Plooy, 2002; Turk, et al., 2002). However, those factors likely occur and difficult to prevent and therefore, the PM framework was used only as a scaffolding.

7.3.2 Process management in practice

The argument above implies another point that process management frameworks are fertilising improvisational and bricolage activities. They can provide valuable points of departure, but should not be allowed to influence every aspect of organisational activities. Improvisation and bricolage involve oriented drifting, so that actors maintain a certain control to act purposeful in order to make progress (Ciborra, 2002; Cunha, et al., 1999; Zheng, et al., 2007). The requirement to have control may be differently developed, but in this case study, the organisational ISO 9001 guidelines served as scaffolding for developers and managers. Nevertheless, improvisers and bricoleurs may utilise various resources and knowledge as scaffoldings, such as ISD methodologies, software aids, and previous experiences. Indeed, Ronkainen and Abrahamsson (2003) state that pervasive use of software aids may be helpful in turbulent environments, hence they may serve as scaffolding for improvisational and bricolage activities.

7.3.3 Improvisation and bricolage as a coping strategy

The analysis of the case study disclosed that improvisation and bricolage activities served to cope with difficulties at the given moment. Therefore, it can be implied that the organisation strategically utilised the improvisational and bricolage capabilities of managers and developers to overcome the difficulties. Difficulties are natural during development projects of information systems and as the discussion highlighted, ambiguous requirements were expected. Therefore, managers and developers tended to improvise and perform bricolage as part of a coping strategy, in order to solve the difficulties. It appears that managers and developers certainly involved trust and confidence in the capabilities of the people that were responsible for the development, because it seems to be relatively risky to rely on improvisation and bricolage as a coping strategy. Consequently, organisations that utilise this coping strategy have people engaged, who are capable enough to cope with unforeseen, without further guidance. Indeed, the case study highlighted the technical capacity of *Theta*'s developers, for example Raphael.

7.3.4 Atmosphere of improvisation and bricolage

Strategically utilising improvisational and bricolage capabilities of developers and managers to cope with difficulties also implies that this organisation involved an atmosphere, where improvisation and bricolage was welcomed. Moreover, the findings of this research suggest that an atmosphere, which facilitates improvisation and bricolage, can be beneficial for development activities. The successful project *Theta* demonstrated that a development atmosphere in which improvisational and bricolage activities were practiced was valuable. In addition, the findings of Weick (2003) support this argument, because there is a relationship between organisational reliability and improvisational and bricolage activities.

Certainly, an atmosphere that facilitates improvisation and bricolage may not be established overnight in practice and it necessitates numerous factors, individuals, and interactions that influence and shape an atmosphere. Such a change needs adequate and continuous adjustment of the organisation, its managers, employees, and the context. As such, procedures and resources allocated to ongoing support of the adjustment are crucial. Consequently, such a culture of improvisation and bricolage needs to support open communication, to ensure that continuous and adequate adjustments take place, to improve the respective circumstances. Open communication needs to take place between decision-makers of the organisation, other managers, employees and relevant surrounding organisations. However, such a change not only involves a top-down approach of communication channels, but criticism and judgment need to be expressed also from bottom-up, so that appropriate and sustainable adjustments may occur. Those approaches facilitate an atmosphere so that improvisation and bricolage is possible.

7.3.5 Nurturing reflexive capabilities

The findings indicate that organisational processes are not fixed but may be malleable by additional dynamics of the ongoing flow of organisational life. This is not necessarily a disadvantage because purposive agents may act to obtain better outcomes. It appears to be that it is worthwhile considering to improve reflexive capabilities of managers and developers because it may provide them advantages in coping with additional dynamics in system development. Giddens (1984) states that reflexive monitoring of action depends upon rationalisation, understood here as a process rather than a state, and as inherently involved in the competence of agents. Furthermore, competence means maintaining a continuing 'theoretical understanding' of the grounds of an agent's activity (Giddens, 1984). Therefore, nurturing reflexivity requires a higher proficiency in the theoretical understanding of the grounds of their activities, such as the context of embedded system development. For practitioners, these findings imply that awareness about insights into the complexity of development processes is important. For example, participation of software developers in other issues of development such as project management or hardware development, questionnaires and assessments in open-minded environments, and the feasibility to put into practice the purposive thoughts of employees may encourage them to think about their actions and practices, improving confidence in their abilities. The case study evinced that questionnaires at EC encouraged developers and managers to reflexively monitor their development activities and to state what worked and what did not work.

7.3.6 Provoking reflexive activities

In addition to the continuous improving of the theoretical understanding, reflexivity could be provoked though non-formal structures. Reflexivity operates mostly in practical consciousness (Giddens, 1984) and, therefore, non-formal structures that support reflexive behaviours would increase the capability to reflexively monitor day-to-day activities. These non-formal activities occur more on an interpersonal or social level and may stimulate the reflexivity of developers and managers. Orlikowski and Hofman (1997) state that enabling conditions for improvisation in an organisation are dedicated resources for ongoing support of improvisers and alignment of processes, according to the organisation and technology. However, the findings of this research imply that reflexive capabilities should become more practiced as part of non-formal structures. For practitioners, this means that interpersonal as well as organisational routines need to be adjusted if developers and managers should utilise their reflexive capabilities better. Those routines should involve certain procedures that involve reflexive behaviours, such as meetings that questions why certain procedures are there and followed. Moreover, routines would improve the ontological security, hence routines support a feeling of security that is convenient and comforting for organisational actors (Giddens, 1984).

7.3.7 Sharing routinised reflexive capabilities

Likewise, the findings imply that reflexive practices that become part of organisational routines involve the potency to change into "knowledge of 'how to go on' in forms of life" (Giddens, 1984). Surely, some knowledge as to how to proceed is inherently part of individual capability, but this argument focuses on the contextual knowledge about ISD activities in an organisation. Mutually sharing that knowledge of how to go on with development would shift the organisational ability to cope with difficulties. Those difficulties may occur suddenly in turbulent environments and require managers and developers, who share reflexive capabilities in order to detect difficulties quick enough. For practitioners, this means that employees and their managers should be treated as one entity, in order to support teambuilding and last but not least 'Teamgeist'. And the empirical findings of this research support this, because Martin (project-manager) stated that the development team shared a common

Teamgeist, particularly at the initial phase of the development. The Teamgeist enabled their capabilities to work together and share knowledge how to go on in order to cope with difficulties during the development of *Theta*.

7.4 Future research

This research focused on the embedded software design in practice in a process management environment. The in-depth case study involved the collection and analysis of rich data from a single case and therefore it would be interesting, how people develop software in similar environments. More in-depth case studies from comparable development settings would provide an even better understanding for researchers and practitioners. However, placing these findings in a well-established theoretical framework, such as concepts of the Structuration Theory, enables other researchers to relate the findings of this research to other cases and findings (Klein & Myers, 1999).

Therefore, future research in other development settings could be useful in order to verify the findings and conceptualisation of this research. The results of this research were based upon the improvisational and bricolage activities of developers and managers, which create an embedded system in an organisation that is ISO 9001-certified and works in the automobile business sector. Therefore, future research that may verify the results of this research could involve differences in:

a) Creating other information systems;
b) Possessing other process management frameworks; and
c) Investigating other business sectors.

The theoretical conceptualisation of this research focused on the capacity to be reflexive, in order to overcome the tensions between process management and improvisation and bricolage. Future research may intend to widen the conceptualisation, in order to involve other aspects that may help to overcome that tension. Other development settings may enable researchers to determine new aspects for that purpose. In addition, other analytical topics may highlight additional issues that may help to distinguish those aspects. However, this research was unable to identify possible other aspects for overcoming the tensional arrangement.

In addition, the new paradox pair of improvisation and bricolage may be helpful for future research, which investigate field data that involve the description of improvisational and bricolage activities. By highlighting the process that is variably positioned throughout development, the new paradox pair intends to complete the prevalent paradoxes. Therefore, future research may describe the phenomena of improvisation and bricolage better.

Moreover, the analysis of empirical data of this case study revealed that some improvisation and bricolage paradoxes were less apparent. For example, the improvisational and bricolage activities of research participants involved less collective individuality and retrospective order. Accordingly, future research may involve an investigation concerning the occurrence of the various paradox pairs. Those findings may determine the explanatory power of the paradoxes.

7.5 Concluding remarks

The work presented in this thesis has focused on the analysis of improvisational and bricolage activities of managers and developers in a process management environment. The aim of this thesis was to make a contribution to knowledge regarding the nature of improvisation and bricolage activities in the practice of embedded software design and how the tensional relationship between process management and improvisation and bricolage can be balanced. There is a lack of understanding as to the full flavour of improvisation and bricolage in embedded system development contexts and how the difficulties correspond to prescribed and emergent processes. In order to address this knowledge gap, I conducted an in-depth case study of an embedded system development project in the German automobile context between December 2004 and November 2008. Collecting and analysing empirical data revealed new insights as to how embedded systems are developed in practice. I adopt the position that emergent processes occur not randomly, but as purposeful agents that navigate through a turbulent environment of ongoing need to improvise with the items at hand. The finding indicates that the success to achieve the aims is bound to the capabilities to be continuously reflexive and induce corrective actions as appropriate. A theoretical conceptualisation disclosed measures that may enhance the capacity to be reflexive. The findings implied that process management frameworks help as scaffolding in order to practice improvisation and bricolage as a coping strategy. Moreover, improving the capabilities to cope with difficulties means to improve reflexive capabilities. The original contribution of this research is founded on rich descriptions and interpretations as to how embedded systems are developed in practice, and the theoretical conceptualisation that can aid to balance the tension between process management and improvisation and bricolage.

8 References

Aaen, I., & Pries-Heje, J. (2004). *Standardising software processes – an obstacle for innovation?* Paper presented at the Sixth IFIP 8.6 Working Conference, Leixlip, Ireland.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods. *Espoo 2002* Retrieved 28th November, 2007, from http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf

Adler, P., Goldoftas, B., & Levine, D. (1999). Flexibility versus Efficiency? A Case Study of Model Changeovers in the Toyota Production System. *Organization Science, 10*(1), 43-68.

Akgün, A., Byrne, J., Lynn, G., & Keskin, H. (2007). New product development in turbulent environments: Impact of improvisation and unlearning on new product performance. *Journal of Engineering and Technology Management, 24*, 203-230.

Akgün, A., & Lynn, G. (2002). New product development team improvisation and speed-to-market: an extended model. *European Journal of Innovation Management, 5*(3), 117-129.

Altheide, D., & Johnson, J. (1994). Criteria for assessing interpretive validity in qualitative research. In N. Denzin & Y. Lincoln (Eds.), *Handbook of Qualitative Research* (pp. 485-499). London, UK: Sage.

Amabile, T. (1988). A model of creativity and innovation in organizations. *Research in organizational behavior, 10*, 123-167.

Automotive-SIG (2005). Automotive SPICE Process Assessment Model. *Automotive SPICE©* Retrieved 5th October, 2007, from http://www.automotivespice.com/web/download.html

Avison, D., & Fitzgerald, G. (2003). Where Now for Development Methodologies. *Communications of the ACM, 46*(1), 79-82.

Avison, D., & Fitzgerald, G. (2006). *Information Systems Development*. Berkshire, UK: McGraw-Hill Education.

Avison, D., & Nandhakumar, J. (1995). The Discipline of Information Systems: Let Many Flowers Bloom! In E. Falkenberg, W. Hesse & A. Olive (Eds.), *Information systems concepts. Towards a consolidation of views* (pp. 1-17). London: Chapman and Hall.

Avison, D., Powell, P., & Adams, C. (1994). Identifying and incorporating change in information systems. *Systems Practice, 7*(2), 143-159.

Baker, T. (2007). Resources in play: Bricolage in the Toy Store(y). *Journal of Business Venturing, 22*, 694-711.

Baker, T., Miner, A., & Eesley, D. (2006). Improvising firms: bricolage, account giving and improvisational competencies in the founding process. *Research Policy, 32*, 255-276.

Baker, T., & Nelson, R. (2005). Creating Something from Nothing: Resource Construction through Entrepreneurial Bricolage. *Administrative Science Quarterly, 50*(3), 329-366.

Bansler, J., & Havn, E. (2003, March 12-13). *Improvisation in Action: Making Sense of IS Development in Organizations.* Paper presented at the Action in Language, Organisations and Information Systems, Linköping, Sweden.

Bansler, J., & Havn, E. (2005). *Improvisation in information systems development.* Technical University of Denmark, Kongens Lyngby, Denmark.

Barrett, F. (1998). Managing and improvising: lessons from jazz. *Career Development International, 3*(7), 283-286.

Baskerville, R. (2006). Artful Planning. *European Journal of Information Systems, 15*, 113-115.

Baskerville, R., & Pries-Heje, J. (2004). Short cycle time systems development. *Infomation Systems Journal, 14*(3), 237-264.

Beck, K. (1999). Embracing Change with Extreme Programming. *Computer, 32*(10), 70-77.

Beck, K., Fowler, M., & Kohnke, J. (2001). *Planning Extreme Programming (XP)*. Amsterdam, Netherlands: Addison-Wesley.

Ben-Ari, M. (1999). *Bricolage Forever!* Paper presented at the Eleventh workshop of the psychology of programming interest group, Leeds, UK.

Benner, M., & Tushman, M. (2002). Process Management and Technological Innovation: A Longitudinal Study of the Photography and Paint Industries. *Administrative Science Quarterly, 47*, 676-706.

Berliner, P. (1994). *Thinking in Jazz: The Infinite Art of Improvisation*. Chicago, IL, USA.

Boccardelli, P., & Magnusson, M. (2006). Dynamic Capabilities in Early-Phase Entrepreneurship. *Knowledge and Process Management, 13*(3), 162-174.

Boehm, B. (1986). A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes, 11*(4), 14-24.

Boehm, B. (2002). Get ready for agile methods, with care. *IEEE Computer, 35*(1), 64-69.

Bogost, I. (2006). Comparative Video Game Criticism. *Games and Culture, 1*(1), 41-46.

Boje, D., & Winsor, R. (1993). The Resurrection of Taylorism: Total Quality Management's Hidden Agenda. *Journal of Organizational Change Management, 6*(4), 57-70.

Brooks, F. (1987). No Silver Bullet: Essence and Accidents of Software Engineering. *Computer, 20*(4), 10-19.

Brown, R., Nerur, S., & Slinkman, C. (2004). *Philosophical Shifts in Software Development.* Paper presented at the Tenth Americas Conference on Information Systems, New York, NY, USA.

Brown, S., & Eisenhardt, K. (1995). Product Development: Past Research, Present Findings, and Future Directions. *The Academy of Management Review, 20*(2), 343-378.

Broy, M. (2006). *Challenges in Automotive Software Engineering.* Paper presented at the International Conference on Software Engineering, Shanghai, China.

Burrell, G., & Morgan, G. (1979). *Sociological Paradigms and Organization Analysis*. London, UK: Heinemann Books.

Butor, M., & Guynn, N. (1994). Bricolage: An Interview With Michel Butor. *Yale French Studies, 84*, 17-26.

Cena, G., Valenzano, A., & Vitturi, S. (2005). Advances in automotive digital communications. *Computer Standards & Interfaces 27*(6), 665-678.

Chen, W., & Hirschheim, R. (2004). A paradigmatic and methodological examination of information systems research from 1991 to 2001. *Information Systems Journal, 14*(3), 197-235.

Christel, M., & Kang, K. (1992). *Issues in requirements elicitation.* Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

Chua, W. (1986). Radical Developments in Accounting Thought. *The Accounting Review, 61*, 601-632.

Ciborra, C. (1996). The Platform Organization: Recombining Strategies, Structures, and Surprises. *Organization Science, 7*(2), 103-118.

Ciborra, C. (1999). A Theory of Information Systems Based on Improvisation. In W. Currie & B. Galliers (Eds.), *Rethinking Management Information Systems*. Oxford, UK: Oxford University Press.

Ciborra, C. (2000). *From Control to Drift*. Oxford, UK: Oxford University Press.

Ciborra, C. (2002). *The Labyrinths of information*. Oxford, UK: Oxford University Press.

Ciborra, C., & Lanzara, G. (1994). Formative Contexts and Information Technology: Understanding the Dynamics of Innovation in Organizations. *Accounting Management and Information Technologies, 4*, 61-86.

CMMI® for Development, Version 1.2 (2006). *CMMI-DEV, V1.2* Retrieved 10th October, 2007, from http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf

Cockburn, A. (2001). *Agile Software Development: The Cooperative Game*. Boston, MA, USA: Addison-Wesley.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Software, IEEE, 34*(11), 131-133.

Cockburn, A., & Williams, L. (2001). *The costs and benefits of pair programming*. Boston, MA, USA: Addison-Wesley Longman Publishing.

Cohen, M., March, J., & Olsen, J. (1972). A Garbage Can Model of Organizational Choice. *Administrative Science Quarterly, 17*, 1-25.

Cooper, R. (2000). Information Technology Development Creativity: A Case Study of Attempted Radical Change. *MIS Quarterly, 24*(2), 245-276.

Coppola, N., & Elliot, N. (2005). Big Science or Bricolage: An Alternative Model for Research in Technical Communication. *Transaction on professional communication, 48*(3), 261-268.

Couger, J. D., Higgins, L. F., & McIntyre, S. C. (1993). (Un)Structured Creativity in Information Systems Organizations. *MIS Quarterly, 17*(4), 375-397.

Coulin, C., Zowghi, D., & Sahraoui, A. (2006). A Situational Method Engineering Approach to Requirements Elicitation Workshops in the Software Development Process. *Software Process: Improvement and Practice, 11*, 451-464.

Crosby, P. (1979). *Quality is Free*. New York, NY, USA: McGraw-Hill.

Crossan, M. (1998). Improvisation in Action. *Organization Science, 9*(5), 593-599.

Crotty, C. (2005). Infotainment invades the auto. *Electronic Business, September*, s3-s11.

Crouch, C., Streeck, W., Whitley, R., & Campbell, J. (2007). Institutional change and globalization. *Socio-Economic Review, 5*, 527-567.

Crutchfield, R. (1962). Conformity and creative thinking. In H. Gruber, G. Terrell & M. Wertheimer (Eds.), *Contemporary approaches to creative thinking*. New York, NY, USA: Prentice-Hall.

Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. New York, NY, USA: Harper & Row.

Cummings, L. (1965). Organizational Climates for Creativity. *The Academy of Management Journal, 8*(3), 220-227.

Cunha, M. (2003). Time Traveling: Organizational Foresight as Temporal Reflexivity. In H. Tsoukas & J. Shepherd (Eds.), *Managing the Future: Foresight in the Knowledge Economy*. Oxford, UK: Blackwell Publishing.

Cunha, M. (2005). *Bricolage in Organization.* Universidad Nova de Lisboa, Portugal.

Cunha, M., & Cunha, J. (2003). Organizational improvisation and change: two syntheses and a filled gap. *Journal of Organizational Change, 16*(2), 169-185.

Cunha, M., & Cunha, J. (2006). Towards a complexity theory of strategy. *Management Decision, 44*(7), 839-850.

Cunha, M., Cunha, J., & Kamoche, K. (1999). Organizational improvisation: what, when, how and why. *International Journal of Management Reviews, 1*(3), 299-341.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM, 31*(11), 1268-1287.

D'Agostino, F. (2006). Two conceptions of reason. *Economy and Society, 35*(1), 1-21.

Dahlbom, B., & Mathiassen, L. (1993). *Computers in context: The philosophy and practice of systems design*. Malden, MA, USA: Wiley-Blackwell.

Dehlin, E. (2008). *The Flesh and Blood of Improvisation.* Norwegian University of Science and Technology, Trondheim, Norway.

Dick, G. P. (2000). ISO 9000 certification benefits, reality or myth? *The TQM Magazine, 12*(6), 365-371.

du Plooy, N. (2002). Information Systems as Social Systems. In J. Cano (Ed.), *Critical reflections on information systems*. London, UK: Idea Group Inc.

Dvir, D., & Lechler, T. (2004). Plans are nothing, changing plans is everything: the impact of changes on project success. *Research Policy, 33*, 1-15.

Dybå, T. (2000). Improvisation in Small Software Organizations. *Software, IEEE, 17*, 82-87.

Eaglestone, B., Ford, N., Brown, G., & Moore, A. (2007). Information systems and creativity: an empirical study. *Journal of Documentation, 63*(4), 443-464.

Edmonds, E., & Candy, L. (2002). Creativity, Art Practice, and Knowledge. *Communications of the ACM, 45*(10), 91-95.

Eisenhardt, K., & Tabrizi, B. (1995). Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry. *Administrative Science Quarterly, 40*(1), 84-110.

Evans-Pritchard, E. (1940). *The Nuer: A Description of the Modes of Livelihood and Political Institutions of a Nilotic People*. Oxford, UK: Clarendon Press.

Fabbrini, F., Fusani, M., Lami, G., Sivera, E., & Sivera, E. (2007). A SPICE-based Software Supplier Qualification Mechanism in Automotive Industry. *Software Process: Improvement and Practice, 12*(6), 523-528.

Fayad, M. (1997). Software Development Process: A Necessary Evil. *Communications of the ACM, 40*(9), 101-103.

Fielden, K., & Malcolm, P. (2006). *Organisational pathways: Creativity to productivity.* Paper presented at the European Conference on Information Systems, Goteborg, Sweden.

Fitzgerald, B. (1997). The use of systems development methodologies in practice: a field study. *Information Systems Journal, 7*, 201-212.

Fitzgerald, B. (1998). An empirical investigation into the adoption of systems development methodologies. *Information and Management, 34*(6), 317-328.

Forsberg, K., & Mooz, H. (1997). System Engineering Overview. In R. Thayer & M. Dorfman (Eds.), *Software Requirements Engineering - Second Edition*. Washington, DC, USA: IEEE Computer Society Press Tutorial.

Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. *Software Development, 9*(8), 28-32.

Frank, R. (2005). Infotainment: Bringing Consumer Electronics Into the Vehicle. *Ward's Auto Electronics, January*, 22-28.

Gajski, D., & Vahid, F. (1995). Specification and design of embedded hardware-software systems. *Design & Test of Computers, 12*(1), 53-67.

Gallivan, M. (2003). The influence of software developers’ creative style on their attitudes to and assimilation of a software process innovation. *Information and Management, 40*, 443-465.

Garud, R., & Karnøe, P. (2003). Bricolage versus breakthrough: distributed and embedded agency in technology entrepreneurship. *Research Policy, 32*, 277-300.

Geertz, C. (1973). *The Interpretation of Cultures.* New York, NY, USA: Basic Books.

Geiger, W., & Kotte, W. (2005). *Handbuch Qualität; Grundlagen und Elemente des Qualitätsmanagements: Systeme - Perspektiven*. Wiesbaden, Germany: Vieweg & Sohn Verlag.

Ghezzi, C., Jazayeri, M., & Mandrioli, D. (1991). *Fundamentals of Software Engineering*. Upper Saddle River, NJ, USA: Prentice-Hall.

Giddens, A. (1984). *The Constitution of Society: Outline of the Theory of Structuration*. Cambridge, UK: Polity Press.

Glass, R. (1995). *Software Creativity*. Upper Saddle River, NJ, USA: Prentice Hall.

Goffman, E. (1959). *The presentation of self in everyday life*. Garden City, NY, USA: Doubleday.

Goossens, G., Van Praet, J., Lanneer, D., Geurts, W., Kifli, A., Liem, C., et al. (1997). Embedded software in real-time signal processing systems: designtechnologies. *Proceedings of the IEEE, 85*(3), 436-454.

Graaf, B., Lormans, M., & Toetenel, H. (2003). Embedded software engineering: state of the practice. *Software, IEEE, 20*(6), 61-69.

Greene, B. (2004). *Agile Methods Applied to Embedded Firmware Development.* Paper presented at the Agile Development Conference, Salt Lake City, UT, USA.

Gu, M., & Tong, X. (2004). Towards Hypotheses on Creativity in Software Development *Product Focused Software Process Improvement* (pp. 47-61). Berlin: Springer.

Guba, E., & Lincoln, Y. (1994). Competing Paradigms in Qualitative Research. In N. Denzin & Y. Lincoln (Eds.), *Handbook of Qualitative Research* (pp. 105-117). Thousand Oaks, CA, USA: Sage Publications.

Guler, I., Guillén, M., & MacPherson, J. (2002). Global Competition, Institutions, and the diffusion of organizational practices: The international spread of ISO 9000 quality certificates. *Administrative Science Quarterly, 47*(2), 207-232.

Handelsgesetzbuch (2008).

Hansen, B., Kautz, K., & Jacobsen, D. (2003). *Information systems developement methodologies in practice.* Paper presented at the IRIS 26 Conference, Helsinki, Finland.

Harrington, H., & Mathers, D. (1997). *ISO 9000 and Beyond: From Compliance to Performance Improvement*. New York, NY, USA: McGraw-Hill.

Hatch, M. (1999). Exploring the empty spaces of organizing: How improvisational jazz helps redescribe organizational structure. *Organization Studies, 20*, 75-100.

Hatch, M., & Weick, K. (1998). Critical Resistance to the Jazz Metaphor *Organization Science, 9*(5), 600-604.

Hatton, E. (1989). Lévi-Strauss's "Bricolage" and Theorizing Teachers' Work. *Anthropology & Education Quarterly, 20*(2), 74-96.

Heidegger, M. (1962). *Being and Time*. Oxford, UK: Basil Blackwell.

Hirschheim, R. (1985). Information systems epistemology: an historical perspective. In E. Mumford, R. Hurshheim, G. Fitzgerald & T. Wood Harper (Eds.), *Research Methods In Information Systems*. New York, NY, USA: North-Holland.

Hmieleski, K., & Corbett, A. (2006). Proclivity for Improvisation as a Predictor of Entrepreneurial Intentions. *Journal of Small Business Management, 44*(1), 45-63.

Hoerl, R., Snee, R., Czarniak, S., & Parr, W. (2004). The Future of Six Sigma. *Six Sigma Forum Magazine, August*, 38-43.

Hoffmann, H., Leimeister, J., & Krcmar, H. (2007). *A Framework for Developing Personalizeable Mobile Services in Automobiles.* Paper presented at the European Conference on Information Systems, St. Gallen, Switzerland.

Hoffmann, O., Cropley, D., Cropley, A., Nguyen, L., & Swatman, P. (2005). Creativity, Requirements and Perspectives. *Australasian Journal of Information Systems, 13*(1), 159-175.

Huarng, F., Horng, C., & Chen, C. (1999). A study of ISO 9000 process, motivation and performance. *Total Quality Management, 10*(7), 1009-1026.

Husserl, E. (1931). *Ideas: General Introduction to Pure Phenomenology*. New York, NY, USA: Macmillan.

Hutchins, E. (1991). Organizing work by adaptation. *Organization Science, 2*, 14-39.

Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA, USA: The MIT Press.

IEEE (1998a). *IEEE Recommended Practice for Software Requirements Specifications*. New York, NY, USA: The Institute of Electrical and Electronics Engineers, Inc.

IEEE (1998b). *ISO/IEC 12207 Standard for Information Technology - Software life cycle processes*. New York, NY, USA: The Institute of Electrical and Electronics Engineers, Inc.

Iivari, J., Hirschheim, R., & Klein, H. (1998). A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies. *Information Systems Research, 9*(2), 164-193.

Iivari, J., Hirschheim, R., & Klein, H. (2004). Towards a distinctive body of knowledge for Information Systems experts: coding ISD process knowledge in two IS journals. *Infomation Systems Journal, 14*, 313-342.

Innes, J., & Booher, D. (1999). Consensus Building as Role Playing and Bricolage. *Journal of the American Planning Association, 65*(1), 9-26.

ISO (2000a). Quality management systems - Requirements (Vol. ISO 9001:2000): International Organization for Standardization.

ISO (2000b). Quality management systems —Guidelines for performance improvements (Vol. ISO 9004:2000): International Organization for Standardization.

ISO (2005). Quality management systems - Fundamentals and vocabulary (Vol. ISO 9000:2005): International Organization for Standardization.

ISO (2008a). Implementation Guidance for ISO 9001:2008 (Vol. ISO/TC 176/SC 2/N 836): International Organization for Standardization.

ISO (2008b). Quality management systems - Requirements (Vol. ISO 9001:2008): International Organization for Standardization.

ISO/IEC (2001). Guidelines for quality management system documentation (Vol. ISO/TR 10013:2001): International Organization for Standardization.

ISO/IEC (2003). Information technology — Process assessment — Part 2: Performing an assessment (Vol. ISO/IEC 15504-2): International Organization for Standardization.

ISO/IEC (2004a). Information technology — Process assessment — Part 1: Concepts and vocabulary (Vol. ISO/IEC 15504-1): International Organization for Standardization.

ISO/IEC (2004b). Information technology — Process assessment — Part 3: Guidance on performing an assessment (Vol. ISO/IEC 15504-3): International Organization for Standardization.

ISO/IEC (2004c). Information technology — Process assessment — Part 4: Guidance on use for process improvement and process capability determination (Vol. ISO/IEC 15504-4): International Organization for Standardization.

ISO/IEC (2006). Information technology — Process assessment — Part 5: An exemplar Process Assessment Model (Vol. ISO/IEC 15504-5): International Organization for Standardization.

ISO/IEC (2008a). Information technology — Process assessment — Part 6: An exemplar system life cycle process assessment model (Vol. ISO/IEC TR 15504-6): International Organization for Standardization.

ISO/IEC (2008b). Information technology — Process assessment — Part 7: Assessment of organizational maturity (Vol. ISO/IEC TR 15504-7): International Organization for Standardization.

Jiao, J., & Chen, C. (2006). Customer Requirement Management in Product Development: A Review of Research Issues. *Concurrent Engineering, 14*(3), 173-185.

Jones, M. (1999). Structuration theory. In W. Currie & B. Galliers (Eds.), *Rethinking management information systems*. New York, NY, USA: Oxford University Press.

Jones, M., & Karsten, H. (2003). *Review: Structuration theory and information systems research.* Cambridge, UK: University of Cambridge.

Jones, R., Arndt, G., & Kustin, R. (1997). ISO 9000 among Australian companies: impact of time and reasons for seeking certification on perceptions of benefits received. *International Journal of Quality & Reliability Management, 14*(7).

Jovanovic, V., & Shoemaker, D. (1997). ISO 9001 standard and software quality improvement. *Benchmarking for Quality Management & Technology, 4*(2), 148-159.

Juran, J., & Godfrey, A. (1998). *Juran's Quality Handbook* (Vol. Fifth Edition). New York, NY, USA: McGraw-Hill Professional.

Kamoche, K., & Cunha, M. (2001). Minimal Structures: From Jazz Improvisation to Product Innovation. *Organization Studies, 22*(5), 733-764.

Kamoche, K., Cunha, M., & Cunha, J. (2003). Towards a Theory of Organizational Improvisation: Looking Beyond the Jazz Metaphor. *Journal of Management Studies, 40*(8), 2023-2051.

Kaplan, A. (1964). *The conduct of inquiry*. Scranton, PA, USA: Chandler.

Katayama, H., & Bennett, D. (1999). Agility, adaptability and leanness: A comparison of concepts and a study of practice. *International Journal of Production Economics, 60-61*, 43-51.

Kautz, K., Hansen, B., & Jacobsen, D. (2004). The Utilization of Information Systems Development Methodologies in Practice. *Journal of Information Technology Cases and Applications, 6*(4), 1-20.

Kettunen, P., & Laanti, M. (2005). How to steer an embedded software project: tactics for selecting the software process model. *Information and Software Technology, 47*, 587-608.

Khazanchi, D., & Munkvold, B. (2003). *On the Rhetoric and Relevance of IS Research Paradigms: A Conceptual Framework and Some Propositions.* Paper presented at the Hawaii International Conference on System Sciences, Waikoloa, HI, USA.

Kibler, T., Poferl, S., Böck, G., Huber, H., & Zeeb, E. (2004). Optical Data Buses for Automotive Applications. *Journal of lightwave technology, 22*(9), 2184-2199.

Kincheloe, J. (2001). Describing the Bricolage: Conceptualizing a New Rigor in Qualitative Research. *Qualitative Inquiry, 7*, 679-692.

Kincheloe, J. (2005). On to the Next Level: Continuing the Conceptualization of the Bricolage. *Qualitative Inquiry, 11*, 323-350.

Kirton, M. (1976). Adaptors and innovators: A description and measure. *Journal of Applied Psychology, 61*(5), 622-629.

Klein, H., & Myers, M. (1999). A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly, 23*(1), 67-93.

Kondo, Y. (1996). Are creative ability and work standardization in contradictory relationship? *Training for Quality, 4*(3), 35-39.

Lanzara, G. (1999). Between transient constructs and persistent structures: designing systems in action. *Journal of Strategic Information Systems, 8*, 331-349.

Larman, C., & Basili, V. (2003). Iterative and incremental developments. a brief history. *Computer, 36*(6), 47-56.

Lee, E. (2000). What's ahead for embedded software. *Computer, 33*(9), 18-26.

Lee, E. (2002). Embedded Software. In M. Zelkowitz (Ed.), *Advances in Computers*. London, UK: Academic Press.

Leidner, D., & Jarvenpaa, S. (1993). The information age confronts education: case studies on electronic classrooms. *Infomation Systems Research, 4*(1), 24-54.

Lessnoff, M. (1974). *The Structure of Social Science*. London, UK: George Allen and Unwin.

Lévi-Strauss, C. (1966). *The savage mind*. Chicago, IL, USA: University of Chicago Press.

Lewin, A. (1998). Jazz Improvisation as a Metaphor for Organization Theory. *Organization Science, 9*(5), 539.

Lewis, M., Welsh, M., Dehler, G., & Green, S. (2002). Product Development Tensions: Exploring Contrasting Styles of Project Management. *The Academy of Management Journal, 45*(3), 546-564.

Leybourne, S., & Sadler-Smith, E. (2006). The role of intuition and improvisation in project management. *International Journal of Project Management, 24*, 483-492.

Lincoln, Y., & Guba, E. (1985). *Naturalistic inquiry*. Beverly Hills, CA, USA: Sage.

Louridas, P. (1999). Design as Bricolage: Anthropology Meets Design Thinking. *Design Studies, 20*(6), 517-535.

MacKenzie, K. (1986). Virtual Positions and Power. *Management Science, 32*, 622-642.

Madsen, S., & Kautz, K. (2002, 10-13 August). *Applying System Development Methods in Practice : The RUP example.* Paper presented at the 25th information systems research seminar in Scandinavia, Bautah¢j, Denmark.

Mäkäräinen, M. (2000). Software change management processes in the development of embedded software. *Technical Research Centre of Finland, VTT Publications, 416*.

Malinowski, B. (1922). *Argonauts of the Western Pacific*. New York, NY, USA: E. P. Dutton & Co.

Mangham, I., & Pye, A. (1991). *The Doing of Managing*. Oxford, UK: Blackwell.

Manhart, P., & Schneider, K. (2004). *Breaking the Ice for Agile Development of Embedded Software: An Industry Experience Report.* Paper presented at the International Conference on Software Engineering, Edinburgh, UK.

March, J., & Simon, H. (1958). *Organizations*. New York, NY, USA: Wiley.

Marsh, D. (2005). Embedded Linux steals design wins. *EDN Europe, June*, 20-32.

McConnell, S. (1996). *Rapid development: taming wild software schedules*. Redmond, WA, USA: Microsoft Press.

Mello, A. (2005). The automotive infotainment quandary. *Electronic Business, September*, 6.

Miel, R. (2005). Companies drive for better MP3 integration. *Automotive News, 80*.

Miles, M., & Huberman, M. (1994). *Qualitative Data Analysis*. Thousand Oaks, CA, USA: Sage Publications.

Mills, H., O'Neill, D., Linger, R., Dyer, M., & Quinnan, R. (1980). The management of software engineering. *IBM Systems Journal, 19*(4), 414–477.

Miner, A., Bassoff, P., & Moorman, C. (2001). Organizational improvisation and learning: a field study. *Administrative Science Quarterly, 46*(2), 307-337.

Mintzberg, H., & McHugh, A. (1985). Strategy formation in an adhocracy. *Administrative Science Quarterly, 30*, 160-197.

Mirvis, P. (1998). Practice Improvisation. *Organization Science, 9*(5), 586-592.

Molnar, W., & Nandhakumar, J. (2007). *Managing a New Computer Device Development in a Creative ISO 9001 Certified Company: A Case Study.* Paper presented at the Hawaii International Conference on System Sciences, Waikoloa, HI, USA.

Molnar, W., & Nandhakumar, J. (2008a). *Dealing with Ambiguous and Fluctuating Requirements of Embedded System Development: A Case-Study.*

Paper presented at the International Conference on Information Systems (ICIS), Paris, France.

Molnar, W., & Nandhakumar, J. (2008b). *Embedded System Development Process: A Quest from Innovation to Adaption.* Paper presented at the 16th European Conference on Information Systems (ECIS), Galway, Ireland.

Molnar, W., & Nandhakumar, J. (2009). *Managing Projects in an Embedded System Development Context: An in-depth Case Study from an Improvisational Perspective.* Paper presented at the Hawaii International Conference on Systems Science, Waikoloa, HI, USA.

Montuori, A. (2003). The complexity of improvisation and the improvisation of complexity: Social science, art and creativity. *Human Relation, 56*(2), 237-255.

Moorman, C., & Miner, A. (1998a). Organizational Improvisation and Organizational Memory. *The Academy of Management Review, 23*(4), 698-723.

Moorman, C., & Miner, A. (1998b). The Convergence of Planning and Execution: Improvisation in New Product Development. *Journal of Marketing, 62*(3), 1-20.

Morgan, J. (2005). Portable Entertainment and Information in the Car. *Ward's Auto Electronics, July/August*, 41.

Motwani, J., Kumar, A., & Cheng, C. (1996). A roadmap to implementing ISO 9000. *International Journal of Quality & Reliability Management, 13*(1), 72-83.

Mouakket, S., Sillince, A., & Fretwell-Downing, F. (1994). Information requirements determination in the software industry: a case study. *European Journal of Information Systems, 3*(2), 101-111.

Myers, M. (1997). Qualitative Research in Information Systems. *MIS Quarterly, 21*(2), 241-242.

Myers, M. (1999). Investigating information systems with ethnographic research. *Communications of AIS, 2*(23), 1-19.

Nanda, V. (2003). *ISO 9001:2000: Achieving Compliance and Continuous Improvement in Software Development Companies* Boca Raton, FL, USA: CRC Press.

Nandhakumar, J., & Avison, D. (1999). The fiction of methodological development: a field study of information systems development. *Information Technology & People, 12*(2), 176-191.

Nandhakumar, J., & Jones, M. (1997). Too close for comfort? Distance and engagement in interpretive information systems research. *Infomation Systems Journal, 7*(2), 109-131.

Nandhakumar, J., & Jones, M. (2002). Development gain? Participant observation in interpretive management information systems research. *Qualitative Research, 2*(3), 323-341.

Newman, M., & Robey, D. (1992). A Social Process Model of User-Analyst Relationships. *MIS Quarterly, 16*(2), 249-269.

Nuseibeh, B., & Easterbrook, S. (2000). *Requirements engineering: a roadmap.* Paper presented at the International Conference on Software Engineering, Limerick, Ireland.

Orlikowski, W. (1995). *Improvising Organizational Transformation Over Time: A Situated Change Perspective*. Cambridge, MA, USA: Sloan School of Management, Massachusetts Institute of Technology.

Orlikowski, W., & Baroudi, J. (1991). Studying Information Technology in Organizations: Research Approaches and Assumptions. *Infomation Systems Research, 2*(1), 1-28.

Orlikowski, W., & Hofmann, D. (1997). An Improvisational Model of Change Management: The Case of Groupware Technologies. *Sloan Management Review, 38*(2), 11-21.

Orlikowski, W., & Robey, D. (1991). Information Technology and the Structuring of Organizations. *Information Systems Research, 2*(2), 143-169.

Paulk, M. (1995). How ISO 9001 compares with the CMM. *Software, IEEE, 12*(1), 74-83.

Paulk, M., Konrad, M., & Garcia, S. (1995). CMM Versus SPICE Architectures. *Software Process Newsletter, Spring,* 7-11.

Pentland, B. (1992). Organizing moves in software support hot lines. *Administrative Science Quarterly, 37*, 517-548.

Pentland, B., & Feldman, M. (2005). Organizational routines as a unit of analysis. *Industrial and Corporate Change, 14*(5), 793-815.

Peplowski, K. (1998). The Process of Improvisation. *Organization Science, 9*(5), 560-561.

Plsek, P. (1997). *Creativity, Innovation and Quality*. Milwaukee, WI, USA: ASQ Quality Press.

Poole, M., & DeSanctis, G. (2003). Structuration Theory in Information Systems Research: Methods and Controversies. In M. Whitman & A. Woszczynski (Eds.), *The Handbook for Information Systems Research*. London, UK: Idea Group Publishing.

Powers, W., & Nicastri, P. (2000). Automotive vehicle control challenges in the 21st century. *Control Engineering Practice, 8*, 605-618.

Pozzebon, M., & Pinsonneault, A. (2005). Challenges in Conducting Empirical Work Using Structuration Theory: Learning from IT Research. *Organization Studies, 26*(9), 1353-1376.

Prasad, K. (2005). Emerging vehicle technologies and services. *Ward's Auto Electronics, March*, 40.

Pressman, R. (1997). *Software Engineering: A Practitioner's Approach*. New York, NY, USA: McGraw Hill.

Prigogine, I., & Stengers, I. (1984). *Order out of chaos*. Toronto, Canada: Bantam Books.

Rada, R. (1996). ISO 9000 Reflects the Best in Standards. *Communications of the ACM, 39*(3), 17-20.

Radice, R., Harding, J., Munnis, P., & R., P. (1985). A programming process study. *IBM Systems Journal, 24*(2), 91-101.

Rautiainen, K., Vuornos, L., & Lassenius, C. (2003). *An Experience in Combining Flexibility and Control in a Small Company's Software Product Development Process.* Paper presented at the ACM-IEEE International Symposium on Empirical Software Engineering, Rome, Italy.

Ronkainen, J., & Abrahamsson, P. (2003). *Software Development Under Stringent Hardware Constraints: Do Agile Methods Have a Chance.* Paper presented at the XP 2003, Genova, Italy.

Rout, T. (1998). *SPICE and the CMM: is the CMM compatible with ISO/IEC 15504?* Paper presented at the AquIS, Venice, Italy.

Rout, T., & Tuffley, A. (2007). Harmonizing ISO/IEC 15504 AND CMMI. *Software Process: Improvement and Practice, 12*, 361-371.

Royce, W. (1970). *Managing the development of large software systems: Concepts and techniques.* Paper presented at the IEEE Westcon, Los Angeles, CA, USA.

Rubin, I., & Rubin, H. (2005). *Qualitative Interviewing*. Thousand Oaks, CA, USA: Sage Publications.

Ryle, G. (1979). Improvisation. In G. Ryle (Ed.), *On Thinking*. London, UK: Blackwell.

Sauer, C., & Lau, C. (1997). Trying to adopt systems development methodologies - a case-based exploration of business users' interests. *Infomation Systems Journal, 7*, 255-275.

Sawyer, S., & Guinan, P. (1998). Software development: Processes and performance. *IBM Systems Journal, 37*(4), 552-569.

Shehabuddeen, N., Hunt, F., & Probert, D. (2002). *Insights into embedded software sourcing decisions: practical concerns and business perspectives.* Paper presented at the IEEE International Engineering Management Conference IEMC'02, Cambridge, UK.

Singels, J., Ruël, G., & van de Water, H. (2001). ISO 9000 series – Certification and performance. *International Journal of Quality and Reliability Management, 18*, 62-75.

Smith, T. (2002). The future of management system standards. *ISO Management Systems, November-December,* 21-25.

Sommerville, I. (1996). Software process models. *ACM Computing Surveys, 28*(1), 269-271.

Sommerville, I. (2007). *Software Engineering* (Vol. 8th Edition). Essex, UK: Pearson Education Limited.

Sorensen, R. (1995). A comparison of software development methodologies. *CrossTalk, 8*(1).

Stahl, B. (2005). The obituary as bricolage: the Mann Gulch disaster and the problem of heroic rationality. *European Journal of Information Systems, 14*, 487-491.

Steen, O. (2007). Practical knowledge and its importance for software product quality. *Information and Software Technology, 49*, 625-636.

Stienen, H. (1999). Nach CMM und BOOTSTRAP: SPICE - Die neue Norm für Prozessbewertungen. *Industrie Management, 15*(4), 67-71.

Stolterman, E. (1991). How system designers think about design and methods. *Scandinavian Journal of Information Systems, 3*, 137-150.

Sutcliffe, K., Sitkin, S., & Browning, L. (2000). Tailoring process management to situational requirements. In R. Cole & W. Scott (Eds.), *The Quality Movement and Organization Theory* (pp. 315-330). London, UK: Sage.

Taramaa, J., Khurana, M., Kuvaja, P., Lehtonen, J., Oivo, M., & Seppanen, V. (1998, 25-27 Aug). *Product-based software process improvement for embedded systems.* Paper presented at the Euromicro Conference, Vasteras, Sweden.

Teece, D. (1996). Firm organization, industrial structure, and technological innovation. *Journal of Economic Behavior and Organization, 31*(2), 193-224.

Townsend, S. (2007). Over the waterfall. *ITNOW, 49*(1), 9.

Truex, D., Baskerville, R., & Klein, H. (1999). Growing systems in emergent organisations. *Communications of the ACM, 42*(8), 117-123.

Truex, D., Baskerville, R., & Travis, J. (2000). Amethodical systems development: the deferred meaning of systems development methods. *Accounting Management and Information Technologies, 10*, 53-79.

Turk, D., France, R., & Rumpe, B. (2002). *Limitations of Agile Software Processes.* Paper presented at the Third International Conference on eXtreme Programming and Agile Processes in Software Engineering, Alghero, Italy.

Turk, D., France, R., & Rumpe, B. (2005). Assumptions Underlying Agile Software-Development Processes. *Journal of Database Management, 16*(4), 62-87.

Turner, R., & Boehm, B. (2003). People factors in software management. *CrossTalk, 16*(12), 4-8.

Tyler, S., & Tyler, M. (1991). Foreword Therapy. In B. Keeney (Ed.), *Improvisational Therapy: A Practical Guide for Creative Clinical Strategies* (pp. ix-xi). New York, NY, USA: Guilford Press.

van der Wiele, A., & Williams, A. (2000). ISO 9000 series registration to business excellence: the migratory path. *Business Process Management, 6*(5), 417-427.

Vera, D., & Crossan, M. (2004). Theatrical Improvisation: Lessons for Organizations. *Organization Studies, 25*, 727-749.

Vera, D., & Crossan, M. (2005). Improvisation and Innovative Performance in Teams. *Organization Science, 16*(3), 203-224.

Verjans, S. (2005). Bricolage as a way of life - improvisation and irony in information systems. *European Journal of Information Systems, 14*, 504-506.

Walsham, G. (1993). *Interpreting Information Systems in Organizations*. New York, NY, USA: John Wiley.

Walsham, G. (1995). Interpretive Case Studies in IS Research: Nature and Method. *European Journal of Information Systems, 4*(2), 74-81.

Wang, Y., Court, I., Ross, M., Staples, G., King, G., & Dorling, A. (1997). *Quantitative Evaluation of the SPICE, CMM, ISO 9000 and BOOTSTRAP.* Paper presented at the International Software Engineering Standards Symposium, Walnut Creek, CA, USA.

Wastell, D. (1996). The fetish of technique: methodology as a social defence. *Infomation Systems Journal, 6*, 25-40.

Weick, K. (1993a). Organizational redesign as improvisation. In H. Huber & W. Glick (Eds.), *Organizational Change and Redesign*. New York, NY, USA: Oxford University Press.

Weick, K. (1993b). The collapse of sensemaking in organizations: The Mann Gulch disaster. *Administrative Science Quarterly, 38*, 628-652.

Weick, K. (1995). Creativity and the Aesthetics of Imperfection. In C. Ford & D. Gioia (Eds.), *Creative Action in Organizations. Ivory Tower Visions and Real World Voices* Thousand Oaks, CA, USA: SAGE Publications.

Weick, K. (1998). Improvisation as a Mindset for Organizational Analysis. *Organization Science, 9*(5), 543-555.

Weick, K. (1999). The aesthetic of imperfection in orchestras and organizations. In M. Cunha & C. Marques (Eds.), *Readings in Organization Science*. Lisbon, Portugal: ISPA.

Weick, K. (2001). *Making sense of the organization*. Maldon, MA, USA: Blackwell Publishing.

Weick, K. (2003). Positive organizing and organizational tragedy. In K. Cameron, J. Dutton & R. Quinn (Eds.), *Positive organizational scholarship*. San Francisco, CA, USA: Berrett Koehler.

Weick, K., & Roberts, K. (1993). Collective Mind in Organizations: Heedful Interrelating on Flight Decks. *Administrative Science Quarterly, 38*, 357-381.

Yusof, S., & Aspinwall, E. (2000). Total quality management implementation frameworks: comparison and review. *Total Quality Management, 11*(3), 281-294.

Zheng, Y., Venters, W., & Cornford, T. (2007). Distributed Development and Scaled Agility: Improvising a Grid for Particle Physics. *Information Systems and Innovation Group Working Paper Series 163* Retrieved 4th April, 2009, from http://is2.lse.ac.uk/wp/pdf/wp163.pdf

9 Appendix

9.1 Implementation timetable for ISO 9001:2008

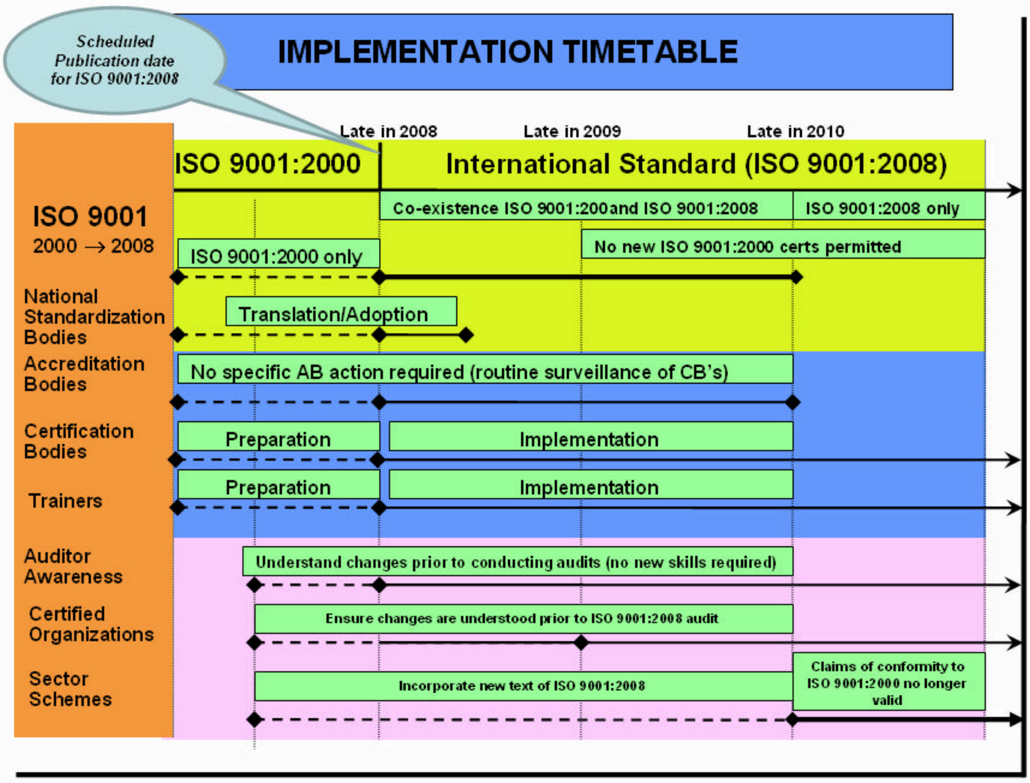

Figure 9-1 Implementation timetable for ISO 9001:2008

9.2 Questionnaire guides

9.2.1 Phase 1 interviews in April 2006

1) Describe your tasks for EC and project *Theta*? Concerning:
    a. What have you done before you joined EC?
    b. Previous tasks within EC, and
    c. Current tasks.
2) As a developers, how would you describe creativity?
    a. What has been very creative and why?
    b. How was this creative solution achieved?, and
    c. Do you get any feedback (compliment, and so on) for creative solutions?
3) What methods and tools do you use and why? Concerning:
    a. Day-to-day project tasks,
    b. Programming,
    c. Documenting, and
    d. Communication.
4) Tell me more about the history of the EC and the *Theta* development? Concerning:
    a. Project characteristics,
        i. What triggers can you identify for the various issues?
        ii. What tools do you use to support the project management?
        iii. What could be improved?
    b. Requirement elicitation and management, and
        i. What triggers can you identify for the various issues?
        ii. How was the elicitation performed?
        iii. What tools do you use to manage the requirements?

        iv. What could be improved?
    c. Concurrent development projects.
        i. What do you think about the concurrent projects?
        ii. What could be improved?
5) What opinion do you have about the perspective of that development project?
6) How would you describe your team environment within EC? Concerning:
    a. Have been conflicts?,
    b. What situations have been very exciting?, and
    c. Are there any outstanding characters within the development team?.
7) What do you think about the quality thinking within EC? Concerning:
    a. How do you define quality?,
    b. What is your share in this quality thinking?, and
    c. What does EC to improve quality?
8) What do you think about EC's policy concerning Kaizen? Concerning:
    a. Why does EC have Kaizen?, and
    b. How does Kaizen influence your day-to-day activities?
9) Why do EC have ISO 9001 certification, what do you think?
10) What do you think about ISO 9001 as part of EC's policies?
11) Did the experiences that ISO 9001 changed anything in the development setting?

9.2.2 Phase 2 interviews in January 2008

1) How would you describe your approach to do your development / management tasks?
    a. How would you describe an engineer?
    b. How would you describe a tinkerer?

c. Have you experienced changes in approaching your tasks, or how the team coped with challenges?

2) Did the way, how you perform your tasks change? Concerning:
    a. Use of programmes for development and management?,
    b. Routines of your day-to-day activities?,
        i. Meetings
        ii. Practices
3) Did you experience any change in your environment during the *Theta* project? Concerning organisation of EC (development, project management, marketing, quality management) and the economic setting:
    a. What sort of change?
    b. Why was this change necessary?
    c. What impact had this change?
4) Lets talk about project *Theta*'s development phase:
    a. How were requirements elicited?
    b. What sort of problems did you experienced and why?
    c. Because you know how things developed, what would you like to do differently?
    d. In what fields do you identify possible challenges?
        i. Why could those challenges raise?
        ii. Involvement of customers,
            - How would you describe the customer: As the final user, marketing people, head of departments of the customer organisations?
        iii. Requirements:
            - Have there been too less?
            - Too ambiguous?
            - Were some issues treated over-ambitioned?
            - Were some issues overlooked?
            - How was the whole planning?

iv. Tell me more about the requirements:
    - How have they been managed?
    - Have they been complete?
    - How consistent were they?
    - How negotiable were they with the customer?
    - Was it possible to trace them?

v. Tell me more about the prototype development:
    - What purpose had this prototype?
    - What pieces did you re-use?

e. If you need to describe the development phase, how would you describe the events at that time?
    i. Your tasks,
    ii. Situation of the development department, and
    iii. Situation of EC.

5) Tell me more about the change-control-board meeting:
    a. Before it started, what the usual way to decide development issues?
    b. How did it start?
    c. What do you think about this meeting concerning your day-to-day work?

6) Do you think, that more or less ISO (managed processes) would have helped the development of *Theta*?
    a. Why and how?

7) What do you think about the future of *Theta*?
    a. Product *Theta*,
    b. Successor products developed by EC, and
    c. What would you like to change for successor developments?

9.3 Photographs

9.3.1 Workplaces of developers

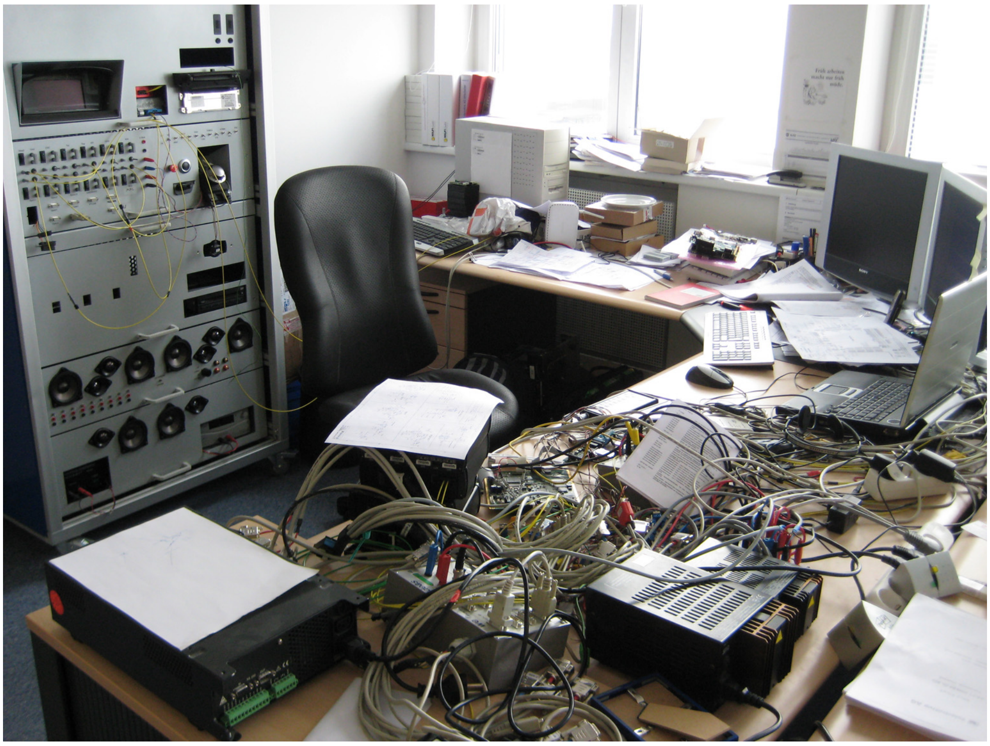

Figure 9-2 Raphael's workplace

Figure 9-3 Junior developers' workplace

Figure 9-4 Test bench of Theta

9.3.2 *Theta* mainboards

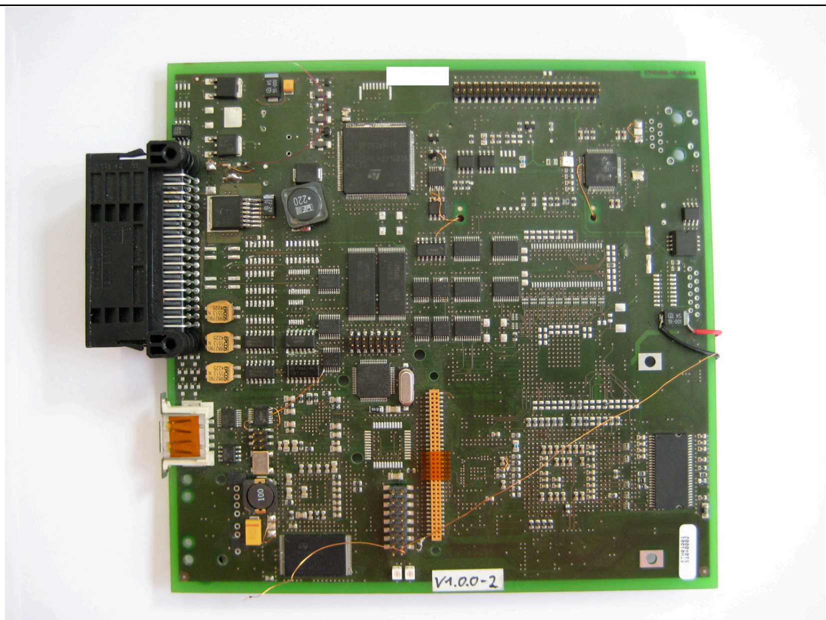

Figure 9-5 Prototype mainboard of Theta

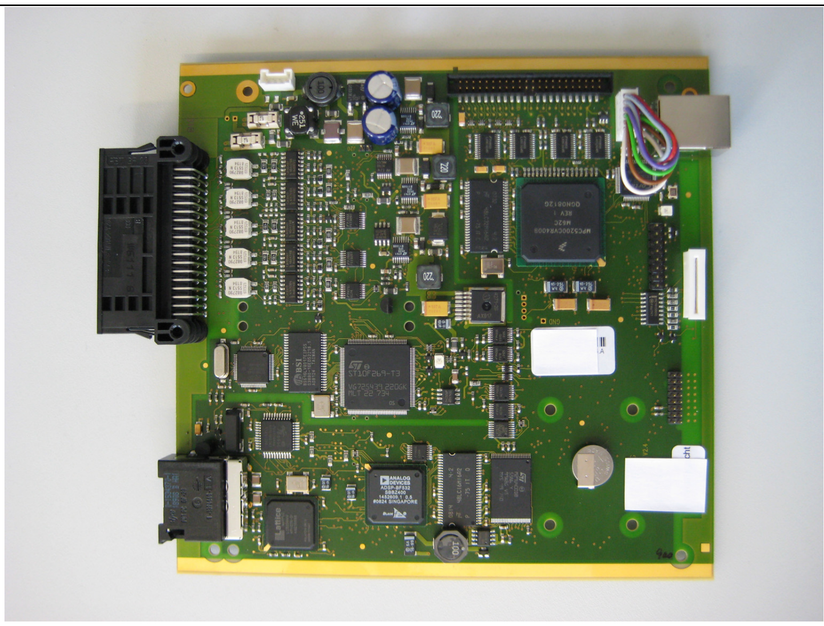

Figure 9-6 Final mainboard of Theta

9.4 Analytical tables

Table 9-1 Analytical table: structured processes	
Analytical focus	**Data gathered from the case study context pertaining the particular analytical focus**
Structured processes	Data from the appearance of EC (Internet, flyers) - advertising with the ISO 9001 certificate - emphasis on certified and well-rehearsed processes - experience with several development projects - focus on the automotive industry - usage of high-tech techniques and tools - value of tailor made products - reference to Kaizen as continuous improvement efforts - reference to SPICE as part of the quality management Data from the top-level management - positive attitude concerning ISO 9001 - understanding, that EC fulfils is fully compliant to ISO 9001 processes - valuation of Kaizen as core of any improvement efforts - consideration of SPICE as valuable part in the future of the quality management - development activities are structured, and involve a lot of planning - developers follow the ISO 9001 standards, like other employees of EC - customers value the high flexibility of EC, in developing tailored solutions Data from the developers management

(continued on next page)

	- mixed attitude concerning ISO 9001
	- understanding that ISO 9001 is necessary for the success of EC
	- understanding that ISO 9001 constrains development activities
	- consideration of SPICE as framework for selective help for EC
	- initial phase of project *Theta* (proof of concept) was not structured at all
	- development activities changed over time
	    o at the start of project *Theta*, less influenced by structured processes
	    o at the end of project *Theta*, processes were better managed
	- requirements were partly unclear, ambiguous and fluctuating; structured processes could not provide help with those circumstances
	- issues with the hardware development were better structured than the software development efforts
	
	Data from the quality managers of EC
	- ISO 9001 is a valuable part of the identity of EC
	- process management is demanded by the customers of EC
	- process management is desired by the managers of EC
	- developers need to try harder to reach higher conformity with the ISO 9001 rules
	- Kaizen is contrasting to the improvement efforts by ISO 9001
	- fulfilment of ISO 9001 processes would not need other ideas such as Kaizen or SPICE
	
	Data from the employees working in the marketing department
	- developers seem to work not really according to structured processes
	- some interaction processes, could be improved
	- development processes appear to be like a “black box”
	- transparency of some processes improved over time
	
	Data from the project manager

	- mixed attitude concerning ISO 9001
	- some practical solutions might lead to better results
	- prescribed processes are constraining
	- however, developers should follow some simple standards
	- another standards (SPICE) would not help to improve the organisation
	- initial phase of project *Theta* (proof of concept) was not structured at all
	- with various new practices, project manager tried to manage the project better
	    o change-control-board meeting
	    o tools: Subversion, Bugzilla
	- change-control-board meeting with participants from marketing department, project managers, developers
	- requirements were partly unclear, ambiguous and fluctuating; structured processes could not provide help with those circumstances
	- requirement management through setting baseline of requirements
	- the ISO processes were better for the hardware development activities; software development activities was depicted wrong by the organisational ISO 9001 framework
	- requirement collection in detail was not practical; used table became difficult to manage
	
	Data from the developers
	- most developers are critical concerning ISO 9001
	- some details of the organisational ISO 9001 processes are just wrong
	- actions such as the annual ISO 9001 audit, the SPICE audit remembered the importance of managed procedures
	- however, perception that this certificate is necessary for marketing reasons
	- initial phase of project *Theta* (proof of concept) was not structured at all

	- no notice of explicit Kaizen procedures
	- perception that tailored SPICE could improve some procedures
	- some procedures changed over time
	    o interaction with customer
	    o interaction within EC
	- "common sense and dedication" would be better than ISO 9001
	- requirements were partly unclear, ambiguous and fluctuating; structured processes could not provide help with those circumstances
	- issues with the hardware development were better structured than the software development efforts
	- Raphael mentioned "structured solution finding," but he could not describe this better
	- state of the art design of software; modular design of software
	
	Data from external ISO 9001 auditors
	- ISO 9001 is important for organisations
	- many cases it is a legal requirement
	- in addition, it helps the organisations to be effective and efficient
	
	Data from the customers of EC
	- value of tailor made products
	- it is not apparently, EC works according to highly standardised principles
	- it is not apparently, how EC develop its products
	
	Data drawn from a comparison of SPICE engineering processes and EC's processes
	- EC would achieve the SPICE maturity level 0 (incomplete process)
	- lack of documentation in every stage of the development process
	- lack of reviews in every stage of the development process
	- insufficient testing efforts of development results
	- lack of conformity between organisational standard of ISO 9001 and the practices of the developers

	Data drawn from field notes
	- some procedures changed over time
	    o interaction with customer
	    o interaction within EC
	- differences in the perception of management processes
	- internal and external ISO 9001 auditors looked only at the surface of development activities
	- some processes were followed more rigidly on the end of the development process, such as testing
	- conformity between organisational standards and practices
	    o generic steps of embedded system development
	- many non-formal standards, that were not part of the organisational ISO 9001 documentation; some emerged over time during the project
	    o change-control-board meeting
	    o tools: Bugzilla, Subversion
	- lack of conformity between company standards and practices

Table 9-2 Analytical table: improvisation and bricolage	
Analytical focus	**Data gathered from the case study context pertaining the particular analytical focus**
Improvisation and bricolage	Data from the appearance of EC - no indication of improvisation / bricolage - necessity of creative efforts, reasoned in the nature of EC's products Data from the top-level management - no indication that top-level enhanced or managed improvisation / bricolage activities from employees - belief of prevailing following of management processes according to the ISO 9001 standard - phrases like tinker and improvisation have an unprofessional quality and they avoided any association with the work practices of EC - however, the phrase creativity is positively associated with the engineering capabilities of EC employees Data from the developers management - awareness that improvisation and bricolage were vital parts of work practices by developers - awareness that the quality of improvisation and bricolage altered over time - tried to enhance and also to manage improvisation and bricolage of developers - understanding that improvisation was necessary, to fill the gaps of ISO 9001 procedures - understanding that EC's size is reasonable, to use improvisation and bricolage intensively - initial phase of project *Theta* (proof of concept) was basically hacked

(continued on next page)

	- requirements were partly unclear, ambiguous and fluctuating; hacking was a way to cope with that circumstances - issues with the hardware development needed less improvisation than the software development - practical learning and doing, through project work - focus of development activities altered Data from the quality managers of EC - improvisation and tinker are the practices imperative to avoid from EC's employees - belief that creative activities (develop high-tech devices) were not constrained but enhanced through ISO 9001 regulations - phrases like tinker and improvisation have an unprofessional quality and they avoided any association with the work practices of EC Data from the employees working in the marketing department - necessity of creative efforts, reasoned in the nature of EC's products - some indications that developers may not work entirely according to standard processes - probably at the start of development, the developers improvised a lot - work style would remain confidential - in fairs we concentrate to present EC as organisation with high flexibility to accommodate customer desires Data from the project manager - mixed attitude concerning improvisation and bricolage - tinker was related with unprofessional work style - hacking was sometimes necessary, to cope with various circumstances - project managers encouraged developers to avoid hacking - initial phase of project *Theta* (proof of concept) was basically

	hacked - vague and unclear requirements reasoned to progress with development activities without final clarification - after the project managers got an impression that the gathered requirements were enough, they set this as an initial baseline of requirements - requirements fluctuated, for short term solutions, we needed to hack; long term solutions required sophisticated planning - focus of development activities altered - first, we set up the change-control-board meeting at a whim, with time it became more regular Data from the developers - tinker was strongly related with unprofessional work style - improvisation was also perceived as work style to avoid - hacking was positively associated with working styles at EC - constraints made it necessary to hack in order to cope with the circumstances - project *Theta* was a result of hacking, particularly at the beginning - over time, the amount of hacking reduced in project *Theta* - in later project phases, developers fixed previous hacks with sophisticated solutions - actions such as the annual ISO 9001 audit, the SPICE audit remembered the importance of managed procedures and to avoid hacking - the level of detail in the prescribed ISO 9001 procedures provides in some fields a lot of space for hacking - initial phase of project *Theta* (proof of concept) was basically hacked - requirements were partly unclear, ambiguous and fluctuating; hacking was a way to cope with that circumstances - issues with the hardware development needed less improvisation than the software development

	- practical learning and doing, through project work
	- use of "common sense" during development (initial phase of recording the MOST signal)
	- focus of development activities altered
	- modular design of software was not always practical; in finding solutions quickly, software developers hacked
	
	Data from external ISO 9001 auditors
	- improvisation and tinker are the practices imperative to avoid from ISO 9001-certified organisations
	- belief that creative activities (develop high-tech devices) were not constrained but enhanced through ISO 9001 regulations
	- no indication of improvisation / bricolage concerning EC
	
	Data from the customers of EC
	- no indication of improvisation / bricolage
	- necessity of creative efforts, reasoned in the nature of EC's products
	
	Data drawn from a comparison of SPICE engineering processes and EC's processes
	- direct indication of improvisation / bricolage (data drawn from documentations of work practices)
	- however, prescribed ISO 9001 processes were not entirely fulfilled
	    o new regular meetings were established
	    o new tools were used
	- lack of conformity between organisational standard of ISO 9001 and the practices of the developers; improvisational and bricolage activities as a way to cope with difficulties
	
	Data drawn from field notes
	- some procedures changed over time
	    o interaction with customer

	o interaction within EC - differences in the perception of management processes - focus of development activities altered - lack of conformity between company standards and practices; improvisation and bricolage activities to cope with challenges - Bugzilla was used not as bug tracking tool, but as database for other incidents of product *Theta* also - later non-formal standards emerged from improvisational activities over time during the project, but were not part of the organisational ISO 9001 documentation o change-control-board meeting

Table 9-3 Analytical table: reflexivity and human agency	
Analytical focus	**Data gathered from the case study context pertaining the particular analytical focus**
Reflexivity and human agency	Data from the project *Theta* team - questionnaire to evaluate the work processes during project *Theta* - dedicated team members - gathered experiences from previous development projects - many decisions were made together in meetings - in critical phases, decisions were made by Michael - initial phase of project *Theta* (proof of concept) was learning by doing - during the development, some errors remained unclear, because hardware as well as software was not error free - focus of development activities altered - adapting to challenges o human resources (increased) o different management (sub-projects) Data from the quality managers of EC - ISO 9001 requires to assess organisational processes regularly, in order to improve its effectiveness and efficiency - EC is structured around the ISO 9001 standard and the top-level management is committed to its validity Data from the top-level management - at the beginning, *Theta* was just another project; later *Theta* was the most important project - focus of development activities altered (e.g.: new customers) - several critical decisions concerning project *Theta*, were made by Michael - Michael asked the project *Theta* team to answer the

(continued on next page)

	questionnaire concerning work processes
	
	Data from the project manager
	- new meetings and tools were introduced to cope with various challenges
	- vital issues of the project management (e.g.: management of human resources) was constrained
	- initial phase of project *Theta* (proof of concept) was learning by doing
	- vague and unclear definition of requirements reasoned to progress the project from a perspective I thought would be best
	- focus of development activities altered
	- after the project managers got an impression that the gathered requirements were enough, they set this as an initial baseline of requirmeents
	- adapting to challenges
	    o human resources (increased)
	    o different management (sub-projects)
	- questionnaire to evaluate the work processes during project *Theta*
	
	Data drawn from field notes
	- hardware was differently developed than software
	- questionnaire to evaluate the work processes during project *Theta*
	- company standards were not followed; practices involved improvisation and bricolage; though, results were OK
	- Bugzilla was used as a multipurpose tool, in order to record incidents of product *Theta*
	- improvisational activities became non-formal standards, because those activities were useful enough to retain them
	    o change-control-board meeting
	- raised conflicts provoked new approaches to cope with difficulties

	o management of customers o management of requirements o lack of conformity with organisational standards and software development reasoned SPICE assessment, in order to find the issues that need improvement - after the SPICE assessment, the developers and project managers documented more, but the second lack (conducting reviews) was not improved

9.5 Results of SPICE assessment

F = Fully achieved

L = Largely achieved

P = Partly achieved

N = Not achieved

BP = Base practice

ENG = Engineering

Table 9-4 Comparison of organisational ISO 9001 standard with SPICE specification (level 1)

SPICE requirement	**Result**
ENG.1 Requirements elicitation	L
BP1: Obtain customer requirements and requests	F
BP2: Understand customer expectations	F
BP3: Agree on requirements	F
BP4: Establish customer requirements baseline	F
BP5: Manage customer requirements changes.	F
BP6: Establish customer query mechanism	F
Example work product: Customer request	N
Example work product: Customer requirements	F
ENG.2 System requirements analysis	P
BP1: Establish system requirements	F
BP2: Optimise project solution	N
BP3: Analyse system requirements	F
BP4: Evaluate and update system requirements	F
BP5: Ensure consistency	N
BP6: Communicate system requirements	N

(continued on next page)

Example work product: Traceability record	N
Example work product: Interface requirements	N
Example work product: System requirements	F
ENG.3 System architectural design	N
BP1: Describe system architecture	N
BP2: Allocate requirements	N
BP3: Define interfaces	N
BP4: Verify system architecture.	N
BP5: Evaluate alternative system architectures.	N
BP6: Ensure consistency	N
BP7: Communicate system architecture design	N
Example work product: System architecture design	N
Example work product: Traceability record	N
Example work product: Verification results	N
ENG.4 Software requirements analysis	P
BP1: Specify software requirements.	L
BP2: Determine operating environment impact	N
BP2: Determine operating environment impact	N
BP3: Develop criteria for software testing	L
BP4: Ensure consistency	N
BP5: Evaluate and update software requirements	L
BP6: Communicate software requirements	N
Example work product: Traceability record	N
Example work product: Interface requirements	N
Example work product: Software requirements	L
ENG.5 Software design	N
BP1: Describe software architecture	N
BP2: Define interfaces	N
BP3: Develop detailed design	N
BP4: Analyse the design for testability	N
BP5: Ensure consistency	N

Example work product: Database design	N
Example work product: High level software design	N
Example work product: Low level software design	N
Example work product: Traceability record	N
ENG.6 Software construction	L
BP1: Develop unit verification procedures	L
BP2: Develop software units	F
BP3: Ensure consistency	L
BP4: Verify software units	L
Example work product: Unit test plan	N
Example work product: Software unit	F
Example work product: Test incident report	L
Example work product: Test case specification	L
ENG.7 Software integration	L
BP1: Develop software integration strategy	N
BP2: Develop tests for integrated software items	P
BP3: Integrate software item.	F
BP4: Test integrated software items	F
BP5: Ensure consistency	L
BP6: Regression test integrated software items	L
Example work product: Build list	N
Example work product: Regression test plan	N
Example work product: Test procedure	L
Example work product: Test incident report	L
ENG.8 Software testing	P
BP1: Develop tests for integrated software product	L
BP2: Test integrated software product	L
BP3: Regression test integrated software	L
Example work product: Software test plan	N
Example work product: Software integration test plan	N
Example work product: Regression test plan	L

Example work product: Software test plan	N
Example work product: System integration test plan	N
Example work product: Test procedure	P
Example work product: Test case specification	P
Example work product: Test incident report	L
Example work product: Defect report	L
ENG.9 System integration	L
BP1: Develop system integration and regression test strategies	L
BP2: Develop tests for system elements	L
BP3: Integrate system elements	F
BP4: Test system elements	L
BP5: Regression test system elements	L
BP6: Ensure consistency	N
BP7: Build complete system of system elements	F
Example work product: System integration test plan	N
Example work product: Test procedure	N
ENG.10 System testing	L
BP1: Develop tests for system.	L
BP2: Test integrated system.	L
BP3: Regression test integrated system	L
BP4: Confirm system readiness	L
Example work product: Regression test plan	L
Example work product: System test plan	P
Example work product: Test procedure	L
Example work product: Test incident report	L
Example work product: Defect report	L
MAN.3 Project management	L
BP1: Define the scope of work	L
BP2: Define project life cycle	L
BP3: Evaluate feasibility of the project	L
BP4: Determine and maintain estimates for project attributes	P

BP5: Define project activities and tasks	L
BP6: Define needs for experience, knowledge and skills	L
BP7: Define project schedule	L
BP8: Identify and monitor project interfaces	L
BP9: Allocate responsibilities	L
BP10: Establish project plan	L
BP11: Implement the project plan	L
BP12: Monitor project attributes	L
BP13: Review progress of the project	L
BP14: Act to correct deviations	L
BP15: Perform project close-out review	N
Example work product: Work breakdown structure	N
Example work product: Schedule	L
Example work product: Project status report	N
Example work product: Project plan	L

Table 9-5 Detailed comparison of EC's practices with SPICE level 1

SPICE requirement	**Results**	
	Project 1	**Project 2**
ENG.1 Requirements elicitation	F	F
BP1: Obtain customer requirements and requests	F	F
BP2: Understand customer expectations	F	F
BP3: Agree on requirements	F	F
BP4: Establish customer requirements baseline	F	F
BP5: Manage customer requirements changes.	F	F
BP6: Establish customer query mechanism	F	F
ENG.2 System requirements analysis	P	P
BP1: Establish system requirements	F	F
BP2: Optimise project solution	N	N
BP3: Analyse system requirements	L	L
BP4: Evaluate and update system requirements	F	F
BP5: Ensure consistency	N	N
BP6: Communicate system requirements	F	F
ENG.3 System architectural design	P	P
BP1: Describe system architecture	F	F
BP2: Allocate requirements	F	F
BP3: Define interfaces	F	F
BP4: Verify system architecture.	N	N
BP5: Evaluate alternative system architectures.	N	N
BP6: Ensure consistency	N	N
BP7: Communicate system architecture design	F	F
ENG.4 Software requirements analysis	N	N
BP1: Specify software requirements.	P	F
BP2: Determine operating environment impact	N	N
BP3: Develop criteria for software testing	N	N
BP4: Ensure consistency	N	N
BP5: Evaluate and update software requirements	N	N

(continued on next page)

BP6: Communicate software requirements	F	F
ENG.5 Software design	N	N
BP1: Describe software architecture	P	P
BP2: Define interfaces	N	N
BP3: Develop detailed design	N	N
BP4: Analyse the design for testability	N	N
BP5: Ensure consistency	N	N
ENG.6 Software construction	N	N
BP1: Develop unit verification procedures	N	N
BP2: Develop software units	L	L
BP3: Ensure consistency	N	N
BP4: Verify software units	N	N
ENG.7 Software integration	N	N
BP1: Develop software integration strategy	N	N
BP2: Develop tests for integrated software items	N	N
BP3: Integrate software item.	L	L
BP4: Test integrated software items	P	P
BP5: Ensure consistency	N	N
BP6: Regression test integrated software items	N	N
ENG.8 Software testing	N	N
BP1: Develop tests for integrated software product	N	N
BP2: Test integrated software product	P	N
BP3: Regression test integrated software	L	N
ENG.9 System integration	N	N
BP1: Develop system integration and regression test strategies	N	N
BP2: Develop tests for system elements	N	N
BP3: Integrate system elements	L	L
BP4: Test system elements	N	N
BP5: Regression test system elements	L	N
BP6: Ensure consistency	N	N
BP7: Build complete system of system elements	L	L
ENG.10 System testing	N	N

BP1: Develop tests for system.	N	N
BP2: Test integrated system.	N	N
BP3: Regression test integrated system	F	N
BP4: Confirm system readiness	F	N