MASTER

Harmonization options for the maintenance and development activities of applications (ALM) and the infrastructure (LCM) at SNS

van den Boogaart, N.C.J.M.

Award date:
2017

Eindhoven, December 2016

Harmonization options for the maintenance and development activities of applications (ALM) and the infrastructure (LCM) at SNS

by
N.C.J.M. (Nadine) van den Boogaart

BSc. Industrial Engineering for Healthcare
Student identity number 0752384

in partial fulfilment of the requirements for the degree of

Master of Science
in Operations Management and Logistics

Supervisors:

dr. ir. H. Eshuis (Rik)	Eindhoven University of Technology
Prof. dr. R.J. Kusters (Rob)	Eindhoven University of Technology
Dr. O. Türetken (Oktay)	Eindhoven University of Technology
E. de Bever (Ewald)	SNS
H. Beltz (Hugo)	SNS

TUE. School of Industrial Engineering

Series Master Theses Operations Management and Logistics

Subject headings: Business processes, process design, lifecycle (products)

Preface

This was the first time I had to perform a big project all by myself. This made me learn a lot about myself and what I can do. Even though I did most of the work, this report would never been finished without the help, guidance and tips from others.

First of all, I would like to thank my first supervisor, Rik Eshuis for his patience and input during this project. With your deep understanding of the topic you helped me find the scope for this project and the balance within the project. Thank you for the time you spend on the meetings and the effort you took in providing feedback. The focus on the report sometimes was hard, but with your positive directions it became a report I am really proud of. I also want to thank Rob Kusters for his clear vision on the method for my report and making me see things from another perspective. This led to a more structured report, which I am really grateful for.

I would also like to thank my company supervisors Ewald de Bever and Hugo Beltz. With a clear vision of what you wanted you guided me through this project. You made sure all information I needed was available and everybody I needed to speak made time for me. I really appreciated that. Furthermore you supported me on a personal level as well. When my motivation was as low at it could be, you both managed to get my spirit up and made me believe I could really finish this project. I would also like to thank the employees of the department I was located. You really accepted me and we had a lot of fun during my time at SNS. The many lunches and walks during the lunchbreak were always fun.

With the defense of my master thesis, my life as a student ends as well. I would like to thank all my friends for their help and love during my thesis. It was not always easy, but because of the nice evenings and talks I managed to stay motivated. I would also like to thank the other students in the K-gang for the cups of coffee, the studying and the motivational words when I deeded them most. Thanks to all friend who made my study the most wonderful time!

Last but not least I would like to thank my parents. It was not always easy when I was home. You were always there to support me and make me believe in myself. Without you my project would never have been finished. Thank you for your inspiration and love.

Thank you all!

Nadine van den Boogaart

Eindhoven, December 2016

Management summary

This master thesis contains the results of a study on harmonizing the work processes related to the development and maintenance of applications and infrastructure at the Samenwerkende Nederlandse Spaarbanken (SNS), 's Hertogenbosch. SNS is a financial institution that feels pressure and need to organize its processes more efficiently to save costs. The general management states that the company has to deliver the same high services and standards on budgets that decrease every year. The goal of this research is to help SNS harmonize their processes related to the maintenance and development of applications and the infrastructure to potentially reduce costs. This leads to the following research question:

How can harmonization between the maintenance and development activities of applications (ALM) and the infrastructure (LCM) be achieved?

SNS composed a taskforce for this study. The company supervisors selected employees with knowledge of the problem area. The taskforce was the main source of information for this study.

Research approach

In order to answer the research question, I used the research method depicted in Figure 1. During the analysis phase, information on the three aspects goals, processes and information was collected, represented and analyzed. These three aspects form the basis for redesigning an organization (Lankhorst, 2009). During the diagnosis phase, the relation between the aspects was examined. Besides needing to be aware of the different aspects, one needs to be aware of their interrelationships (Lankhorst, 2009). The boxes show the aspects that were analyzed during the analysis phase, the lines are the relations examined during the diagnosis phase. Based on the outcome of the analysis and diagnosis a design was created to harmonize the processes of ALM and LCM. The validation of the different models was done by the taskforce per phase of the research.

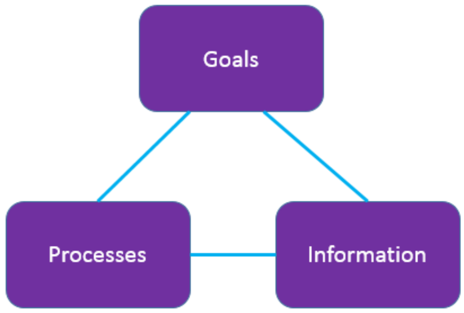


Figure 1: Research method

Analysis and diagnosis

For all three aspects identified in the research approach data was collected. This data was collected by conducting interviews and by reviewing company documentation. Goal models were created for different company levels. Processes were mapped for different sub-departments and based on these processes, data on the information aspect was collected. Based on this data conclusions were drawn on the three aspects (Table 1).

Table 1: Conclusions analysis phase

Aspect	Conclusion
Goals	The user goals with the division between ALM and LCM all relate to the same goals on the ITC level
	Goals of ALM and LCM are conflicting, so they can never both be achieved at the same time
	Identical goals stated the same are interpreted different by different departments
Processes	There is a lot of overlap between the four processes created for the way of working of ALM

	The processes created for the way of working for LCM differ a lot in sort of activities and number of activities
	Between the processes for ALM and LCM there are a lot of differences
Information	Parts of the data are not stored
	Parts of the department data are stored in internal documents (Excel sheets)

The relations between the aspects were also investigated. This was done by linking the collected information of two aspects. The conclusion on the aspects showed that the division between ALM and LCM also had to be taken into account. By investigating the links it became clear that the goals should not change during the design phase, because of the general nature of the goals. They will stay the same over time, which was supported by the interviews. For the processes and information aspect it became clear that the work processes are different for every sub-department. There are differences between the way of storing the data and the type of stored data.

The first step in harmonizing the processes would therefore involve the information aspect. Based on the analysis of the aspects this would be the least impactful change for the company, with the biggest result. With this in mind the design was created.

Design

The goal of the design phase was to create a plan for the harmonization of the processes of ALM and LCM for SNS. The design started with the outcome of the analysis and diagnosis phase, which stated that the first change should consider the information aspect. With this in mind a multiple step implementation was created to reach more harmonization between the processes over time. This multiple step design plan can be found in Figure 2.

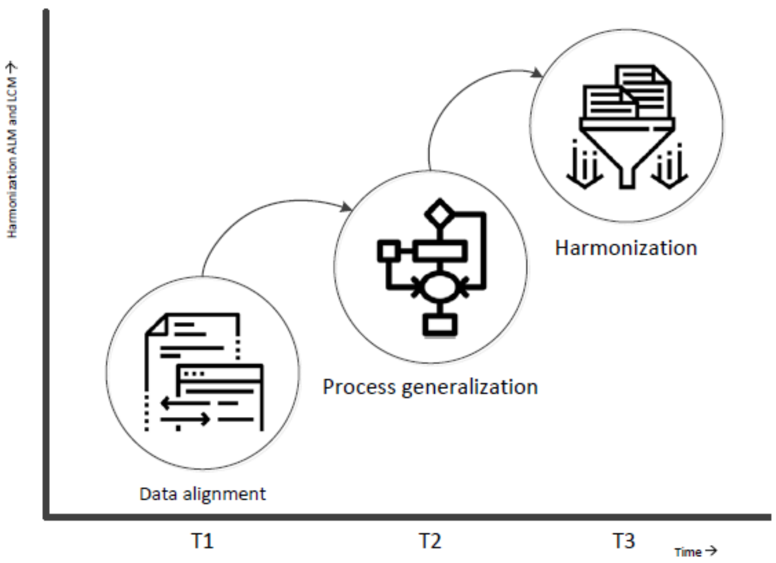


Figure 2: Design plan

The first step of the design is called *data alignment*. During this step all relevant data gets stored using the same format. This means that the data models created of the problem domain during the analysis phase become the actual data models. Now, data gets stored on all steps of the processes, but every sub-department still works by its own process. When all important data gets stored in the same format in the same location, it becomes important to store as much information as possible. This is when the *process generalization* starts. During step two, general processes are created, one for ALM and one for

LCM. Multiple versions of the generalized processes were created. The first version was created by me, which was reviewed by the taskforce. The creation of the generalized process models followed an iterative process. During every step the comments of the taskforce were used to adapt the general processes, until everybody agreed on the final design. By creating generalized processes, more data can get stored on the processes. When all sub-departments working on the same type of problem go through the same processes, this means that data collection goes even faster. Insights based on this data should be used for step three of the design.

During step three of the design, *the harmonization*, the generalized processes of ALM and LCM created during the previous step get harmonized. Until now the interaction or communication between ALM and LCM was not taken into account. During this final design step, the interaction between the processes of ALM and LCM is included. The first activities of the two processes are merged, which leads to an update planning that everybody agrees on. During the whole process it is important that there is communication between the different sub-departments. With better communication it becomes possible to be more efficient during updates and help each other better. When everybody knows what the others are doing, more understanding and insight is generated.

Analyzing data is important, because it gives companies insight into where things go wrong. It is also important that intuition is no longer leading in making changes. The harmonization of the processes of ALM and LCM is done based on the researched aspects. When SNS wants to implement this design other aspects might also be important and need to be taken into account.

Conclusion

This study developed a multiple step design for harmonizing the processes for ALM and LCM within SNS. The processes were created using the input of the taskforce, they also validated the designed process steps. The detailed design cannot be used by other companies than SNS, because SNS specific information is used for the creation of the design. The general steps and procedures can be used to investigate related problems. It is important to keep in mind that the design is created based on current information. The implementation of step one and two take time, that is why it is important that the changes based on the information get included in the design that will eventually be implemented. Not only the data can change, but also the environment itself. The financial sector is a sector that has changed rapidly over the past years and needs to change more towards the future, because customers need to become the core focus of financial institutions (De Nederlandse Bank, 2015). These changes also have to be checked to adapt the created harmonized process. Per step and per change SNS needs to reassess its work processes and see how this influences the outcome.

Table of contents

List of figures

List of tables

List of abbreviations

ALM	Application lifecycle management
BPMN	Business process model and notation
COMET	Component and Model-based development Methodology
I&C	Information & Change
I&O	Infrastructure & Operations
IT	Information technology
ITC	Information Technology and Change department
LCM	Life cycle management
PLC	Product lifecycle
PLM	Product lifecycle management
RACI	Responsible, Accountable, Consulted and Informed
RE	Requirements engineering
SLM	Service lifecycle management
SOA	Service oriented architectures
TAB	Technisch Applicatie beheer
UML	Unified Modelling language

1 Introduction

This master thesis contains the results of a study on harmonizing the work processes related to the development and maintenance of applications and the infrastructure at SNS, ‘s Hertogenbosch.

In a growing competitive market, financial institutions need to adapt their working method. Costs need to be reduced while the quality of financial products and services has to improve (De Nederlandse Bank, 2015). To be able to still have a market share in the future, this research focuses on identifying aspects related to harmonization of the development and maintenance of applications and the infrastructure. Based on found inefficiencies, scenarios for the future are introduced that manage the processes more efficiently, resulting in increased performance and possibly, lower costs.

1.1 Research context

The software industry has been identified as one of the most important industries in the world (Colomo-Palacios, Fernandes, Sabbagh, & Amescua Seco, 2012). Regardless of industry and organization size, information technology (IT) is fundamental for improving productivity and development of knowledge-intensive products and services (Soto Acosto, Placer Maruri, & Perez Gonzales, 2010). Costs of an organization, which main focus is IT, can be accounted for 70% to 80% by the IT services (Orlov, 2005), therefore problems with IT services are a relevant topic for research.

Where IT departments previously focused on the production of software applications, this has started shifting towards more service-focused operations (Marrone, M; Kolbe, LM, 2011). An increasing number of organizations are looking for more efficient and innovative technological services and solutions. This way of thinking followed after a time in which companies believed that every department could be seen as a specialism that is best at making certain decisions and doing specific tasks. Which led to problems within the company, because departments stopped working together and for instance had conflicted versions of the same data (Spark, 2015). Other problems that arose were envirionmental incompatibilities at department borders, waste, gaps, information silos, islands of automation, overlapping networks, ineffective fixes and product recalls. When the problems started to have a big impact on the organisations, the need for a new way of thinking and doing things more integrated was needed. The new way of thinking that arose was lifecycle mangagement (van den Boogaart, 2016) (Appendix C).

Lifecycle management (LCM) provides a generic frame of reference for systems and methods that are necessary for managing all product related data during the product’s lifecycle (Kaariainen & Valimaki, 2008). Lifecycle management can be used for many different types of processes within different fields of work. Implementing any form of lifecycle management gives more insight into project data and real time information can be used to make decisions and control the process. One of the latest adaptations of lifecycle management is application lifecycle management (ALM). ALM has the purpose to provide integrated tools and practices that support project cooperation and communication through an applications lifecycle process. At SNS these terms are used in a different way. For them, *Lifecycle management (LCM)* includes all the lifecycle management actions related to the infrastructure. *Application lifecycle management (ALM)* includes all lifecycle management actions related to the applications used within SNS. The definitions of SNS are used for this research.

This research focusses on how goals, processes and information can be combined to harmonize the processes for SNS in the future. Harmonization should make working together easier and potentially reduce the costs for maintenance and development of applications and the infrastructure. This research first focusses on collecting data on the goals, processes and information. With this input, the connections between the three are explained. Based on this information opportunities for the future are designed.

1.2 Company description

This research is conducted at SNS, at the Information Technology and Change department (ITC) located in s'-Hertogenbosch. SNS is part of SNS Bank N.V. that consists of ASN Bank, BLG Wonen, RegioBank, SNS and ZwitserlevenBank. SNS Bank N.V. has 'Bankieren met de menselijke maat' as a mission, and every one of the five brands fits a specific target group of customers. For SNS this means that they want to offer their services in a more 'human' and 'normal' way (menselijker en normaler). They believe that people have to be able to manage their money 24/7 by their computer, tablet or mobile. SNS has around 200 stores all over the Netherlands where people can get advice on savings, pensions and mortgages (SNS, 2016). Part of the company structure of SNS can be found in Figure 3. This project focusses on the ITC department. Within ITC there are three sub-departments: ITC Bank Data Office, Infrastructure & Operations (I&O) and Information & Change (I&C). I&O is the department that runs the business and who's primary responsibility is to ensure continuity of current services. I&C is the department that changes the business and who's primary responsibility is to realize change within the guaranteed continuity of service by I&O. The ITC Bank Data Office is out of scope for this project because they do not work with any of the processes in scope of this research. The main input for this research came from I&O, because they were the ones that initiated this research. Some input was given by I&C, but they did not really see the relevance of this project in the beginning. However once the first steps were taken more interest came from them, as well as documentation as input on some of the topics. The fact that that I&C got involved later did not lead to problems, because I&O works with all processes related to this research (colored boxes Figure 3).

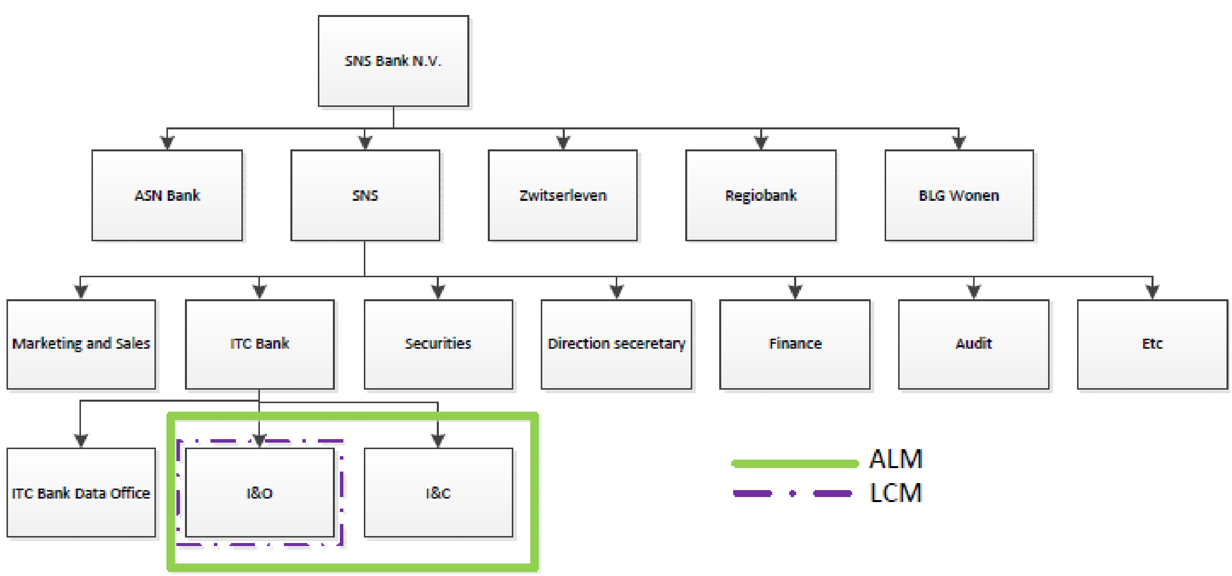


Figure 3: Part of the company structure of SNS

1.2.1 The taskforce

Within I&O, a taskforce was created to support this project for information or help. The most important role of the taskforce starts after this research. They have to translate the outcome of this research to specific solutions for the different sub-departments. During this research they were the main source of information. The members of this taskforce are selected by the company supervisors based on their knowledge of either ALM, LCM or a combination of the two. Other people involved do not work directly with ALM and/or LCM, but know about the creation of processes or have their focus on the security aspect of the systems. The background of all taskforce members is in different fields of knowledge that all influence the ALM and LCM processes. The list of roles of the members included in the taskforce is presented in Appendix A. Only the roles are given, because of privacy no names are stated. When only a selection of the taskforce is used for certain steps this will be explained in the methodology for that part. If this is not stated the whole taskforce was included.

1.3 Report structure

This report is structured in the following way. Chapter 2 gives a summary of the conducted literature study. Chapter 3 focusses on the problem definition and leads to the research questions. This chapter also contains a first introduction on the methodology. Chapter 4 gives a description of the analysis and diagnosis phase. During the first part of this chapter the methodology for the analysis and diagnosis of the research is explained further, which is followed by the results and a diagnosis of the results. This leads to implications for future design possibilities. In Chapter 5 the future design is presented. And finally conclusions are drawn and recommendations are given in Chapter 6.

2 Literature summary

Prior to this research, a literature study was conducted with the main research question; *What is application lifecycle management and how can it be implemented?* To be able to answer this question the paradigm of lifecycle management is researched. The directions that are included in the scope of the literature review are product lifecycle management (PLM), application lifecycle management (ALM) and service lifecycle management (SLM). The total literature review can be found in Appendix C.

Product lifecycle management

The product lifecycle (PLC) represents the unit sales curve for a product, extending from the time it is first placed on the market until it is removed (Rink & Swan, 1979). The product lifecycle portrays the evolution of product attributes and market characteristics through time, and the concept of PLC can be used in a prescriptive way in the selection of marketing actions and planning (Polli, 1968). When talking about a product this can be anything from a pencil to software to a truck. The bell-shaped PLC model is adopted by the field and has a four-stage cycle-introduction that include introduction, growth, maturity and decline.

Product lifecycle management (PLM) can be defined as a strategic business approach for the effective management and use of corporate intellectual capital (Sudarsan, Fenves, Sriram, & Wang, 2005). In other words, this means that PLM manages business activities in the most effective way all across the lifecycle of the product. From the very first idea for a product all the way through until it is retired or disposed of (Spark, 2015). Furthermore, Spark (2015) states that there are two important characteristics for PLM. The first is that when using PLM, the activities that manage a company's product must be defined and documented in cross-functional business processes across the product lifecycle. Furthermore, cross-functional product data are managed by a system that manages the data across the product lifecycle.

Because of global changes, the managing of information in the lifecycle of a product is a major challenge. The benefits of using PLM for this are fast and easy exchange of documents and expertise, real-time control, improved communication and accessibility of product related information. PLM is also a collaborative platform that can improve information access and sharing inside the company, but also between a company and its stakeholders (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016). Felic et al. (2014) add more benefits like reduced time to market, a better collaboration and savings. However, they are careful, because for a lot of companies implementing PLM still means that they have to make heavy changes to the company structure. On top of that, PLM solutions are based on an integrated model that stores product data that is shared with all contributors. Challenges arise when this information can only be interpreted by experts. PLM can also lead to communication overhead that leads to extra costs, extra product development time and therefore longer time to market (Felic, König-Ries, & Klein, 2014). It is important that during the product lifecycle a collaborative approach is used, because problems with using PLM can be categorized as product-centric, process-oriented or human-centric knowledge management (Felic, König-Ries, & Klein, 2014). To be able to manage all data when using PLM, different methods can be used that can be manual as well as software driven.

When thinking about implementing PLM, the cost for doing so have to be taken into account. The product lifecycle costs are an important measure for PLM implementation, because it can track and analyze the financial information of activities associated with each phase of a product's lifecycle (Xu, Chen, & Xie, 2006). Product lifecycle costs refers to all the costs that occur over the whole lifecycle of a single product (Artto, 1994). In general, product lifecycle costs are estimated by using one, or a combination, of the following two methods: the costs of a product are estimated in comparison to the cost of a similar product or component that was made in the past. Or the labor times and rates are estimated, material quantities and prices are calculated to determine the direct costs of a product. On top of this, an allocation rate is used to allow for indirect costs (Shields & Young, 1991).

Application lifecycle management

The application lifecycle consists of application development as well as service management. By using the application lifecycle, a more broad view can be given than when only software development processes are taken into account. To make the lifecycle economically and efficiently, it is necessary that information flows of semantically annotated information is retrievable in a diverse operational infrastructure across organization boundaries (Oberhauser & Schmidt, 2007). Oberhauser and Schmidt (2007) also identify two reasons why the implementation of an application lifecycle can rarely be fully exploited. The first point they mention is that there is a semantic gap between the abstract process descriptions and the executed processes. On top of this, best practices in other organizations cannot directly be reused for other instances. This means that executable processes have to be abstracted manually to get process descriptions that fit a specific company.

Application lifecycle management (ALM) deals with the way a software system or application is conceived, planned, developed, maintained and decommissioned (Rajlich & Bennett, 2000). Typical activities that are included in the lifecycle are requirements development and management, project planning, solution development, deployment and issue tracking. Doyle (2007) states that ALM is a set of tools, processes and practices that enable a development organization to implement and deliver to software lifecycle approaches. This means that some kind of solution for ALM exists in every company (Doyle, 2007). The purpose of ALM is to provide integrated tools and practices that support project cooperation and communication through a project's lifecycle. For management it provides an objective mean to monitor project activities and generate real-time reports from project data. When using ALM it is important to understand it's true scope (Schwaber, 2006). Firstly, ALM is a discipline, as well as a product category. It is sometimes hard to remember that ALM can be implemented using a tool, but also without supporting tools. Secondly, ALM does not support specific life-cycle activities; it rather keeps them all in synchronization. Finally, an ALM solution is the integration of life-cycle tools, not merely a collection thereof.

Chappell (2008) states that for ALM to be both accurate and useful, the view on it should be a broad one. He defines three distinct areas when talking about ALM. The defined areas are governance, development and operations. The purpose of *governance* is to make sure that the application provides what the business needs, and consists of business case development, project portfolio management and application portfolio management. The second area is *development*, which starts after approving the business case. It involves the initial version of the application and the updates and maintenance that keeps the application up to date. The final area is *operations*, which includes monitoring and managing the application. It can entail multiple iterations and is closely related to de development line, because from the moment the application is deployed it has to be monitored throughout its lifetime (Chappell, 2008). Whether ALM activities are faithfully executed remains an area of doubt to many organizations (Rossberg, 2008). Shaw (1990) states that the most common problems are related to the coordination and cooperation among the developers, that view these actions as non-technical overhead. Everybody that works with PLM needs the right information at the right time in the right context.

Service lifecycle management

All information given until now was mainly on the first phases of the lifecycle. However, the service management part of the lifecycle is just as important. A service-oriented business level enables an organization to expose and offer operations as business services to business partners in order to facilitate on-demand collaborations (Kohlborn, Korthaus, & Rosemann, 2009). The basis of the service lifecycle are Service oriented architectures (SOA). A SOA is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains (Kohlborn, Korthaus, & Rosemann, 2009). This is getting more and more important when talking about business models (Mueller, Viering, Legner, & Riempp, 2010), because when organizations also want to offer services, they have to combine the different SOAs (e.g. SAP, Oracle) that are used within the company. It is important that motivations for SOA and services get documented and business and IT imperatives that need to be resolved are mapped (Marks & Bell, 2006).

Software providers no longer offer their solutions solely as complete packages, but rather allow customers to use them in parts or as a whole on a pay-per-use basis (Fischbach, Puschmann, & Alt, 2013). This leads to higher complexity, due to heterogeneous service specifications, service development processes, service implementation and operating models (Puschmann & Alt, 2011). Different suppliers have heterogeneous platforms, which means dedicated management of services along the lifecycle (service lifecycle management - SLM) is needed. Two types of approaches can be used when implementing SLM (Fischbach, Puschmann, & Alt, 2013), the IT-oriented approach or the business-oriented approach. The *IT-oriented approach* or 'SOA Management' can be described as the management and monitoring of applications, services, processes, middleware, infrastructure and software in accordance with the business goals (Behara & Inaganti, 2007). The *business-oriented approach* aims at transferring approaches from industrial product development and product management to the service area, mostly by means of process-based models, because services often have business-oriented aspects that go beyond technical elements.

Both of the above mentioned approaches are integrated in the service lifecycle (Bardhan, Demirkan, Kannan, Kauffman, & Sougstad, 2010). This combination leads to the integrated SLM that can be used in combination with both PLM and ALM during the service/maturity phases. Implementing an integrated SLM solution can be a challenge, because implementing SLM requires extensive standardization with respect to the governance, processes, applications and service descriptions. If a company is able to do this, it can reduce costs for the organization and lead to time benefits. Finally, by creating standardized unified service level agreements, the quality of service definition, provision and enhancement can be increased (Fischbach, Puschmann, & Alt, 2013). However, Fischbach et al. (2013) state that there has not been done a lot of research in these areas, so future research is important on this matter.

This review is used for making a comparison between PLM and ALM to gain more insight in ALM, because of the lack of literature on ALM. The comparison is presented in Table 2.

Table 2: Literature conclusions

	PLM	*ALM*
Benefits	Takes into account the whole lifecycle (Spark, 2015)	Takes into account the whole lifecycle (Doyle, 2007)
	Tools can be used for support in implementation (Xu, Chen, & Xie, 2006)	Tools can be used for support in implementation (Schwaber, 2006)
	Real time insight into project data and control (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016)	Real time insight into project data and control (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016)
	Improved communication (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016)	Very flexible concept that can be implemented in every company (Doyle, 2007)
	Reduced time to market (Felic, König-Ries, & Klein, 2014)	It is a product category as well as a discipline (Schwaber, 2006)
	Savings (Felic, König-Ries, & Klein, 2014)	
	Overview of where different costs are made within the product lifecycle (Savinirs, 2012)	
	Can improve information access and sharing inside the company as well as between the company and its stakeholders (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016)	
	Self-developed PILM frameworks work the best, but take more time to develop (Yang, Moore, Wong, Pu, & Chong, 2007)	
Pitfalls	Heavy changes need to be made to the companies structure when implementing PLM (Felic, König-Ries, & Klein, 2014)	Has to be implemented over organizations boundaries (Oberhauser & Schmidt, 2007)
	Product information is shared with all contributors of the product which is a problem if information can only be interpreted by experts (Felic, König-Ries, & Klein, 2014)	There is a gap between the abstract process description and the executed processes (Oberhauser & Schmidt, 2007)
	Communication overhead (Felic, König-Ries, & Klein, 2014)	No generalizable model or framework available for implementation or re-use (Oberhauser & Schmidt, 2007)
	Lack of interconnectivity with other information systems (Vezzetti, Violante, & Marcolin, 2014)	There can be breaks in the information flow between software operations and software development (Oberhauser & Schmidt, 2007)
	Available software is very expensive (Vezzetti, Violante, & Marcolin, 2014)	

Based on the main research question for this literature review, literature was searched and with this information the review was written. The first part of the question is about what application lifecycle management is. This can be answered by using the definition of Doyle (2007) who states that 'ALM is a set of tools, processes and practices that enable a development organization to implement and deliver to software lifecycle approaches'. The second part of the research question was harder to answer. The answer to the question how application lifecycle management can be implemented is still not clear. Because of the fact that ALM has no standard format for every company to implement, it is not possible to make a general claim on how to implement ALM based on the current available literature. A lot of different factors have to be taken into account when implementing ALM, but how a specific company can implement ALM cannot be answered with a general statement at this moment in time.

The conclusion leads to some directions for further research. The first is an investigation of how ALM can be implemented in different companies. This research can be done using companies with the same characteristics or different ones, so that you can compare or make a more generalized framework for that company type.

Another approach is to see whether ALM can be implemented in combination with other processes a company already uses. This could be lifecycle management, lean, six sigma, etcetera. It would then be really important to see whether ALM can merge all systems using another way of working as a basis.

Furthermore, one could also state that any research that leads to a literary document is useful at this time, because of the lack of grounded research available until now. If more research becomes available, better comparisons can be made between situations and therefore better insight can be given into the concept of ALM.

3 Problem definition

The general explanation of this research can be found in this chapter. First the problem statement is given which leads to the cause-and-effect diagram. Based upon the cause-and-effect diagram, the research question and sub questions are introduced. Afterwards, a general description of the methodology followed in this research is given.

3.1 Problem statement

SNS feels pressure and need to organize its processes more efficiently to save costs. The general management states that the company has to deliver the same high services and standards on budgets that decrease every year. Within the company, projects started to search for possibilities to save costs. These projects involve different fields of work or parts of the company. One of these projects includes gaining more insight in why costs on maintenance and development of applications as well as the infrastructure are so high. Within this project the infrastructure represents all the layers of the stack excluded by the application layer. The layers involved are the databases, middleware, the operating systems and the hardware. Maintenance and development of applications and infrastructure are part of the core business of SNS, and therefore seen as an important focus point for the future. The goal of this research is to help SNS harmonize their processes related to the maintenance and development of applications and the infrastructure to potentially reduce costs.

Within SNS, they believe that there are multiple causes that influence the high costs for maintenance and development of applications and the infrastructure. During interviews with employees (Appendix A) from different fields of work, different causes were named. Based on these interviews a cause-and-effect diagram (Figure 4) is created to give a clear overview of the mentioned causes for this problem.

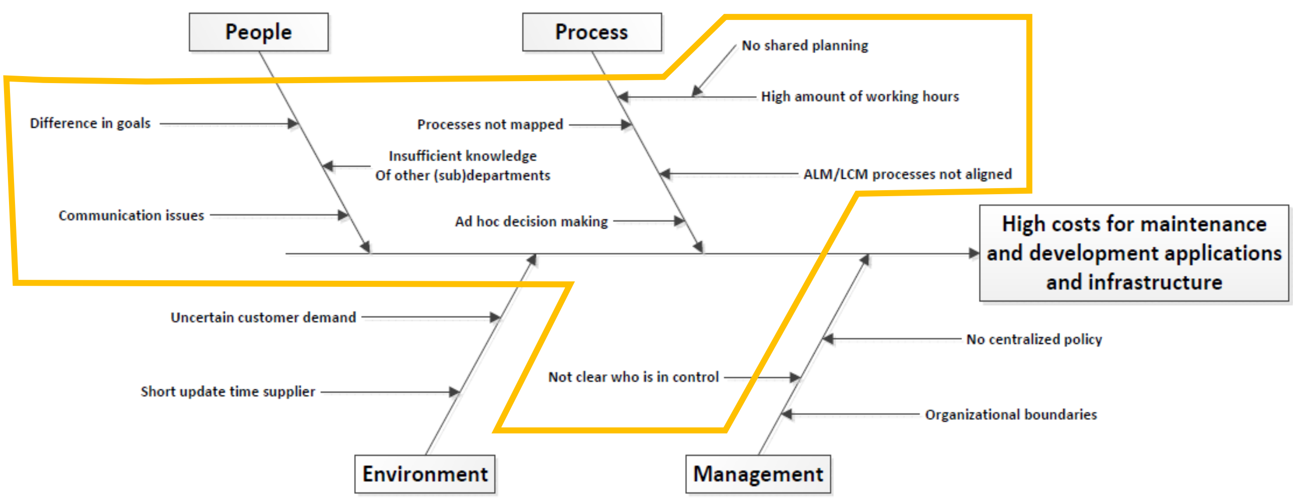


Figure 4: Cause-and-effect diagram

A detailed description of all causes can be found in Appendix B. The conclusion that can be drawn based on this cause-and-effect diagram is that everybody does what they think is best. There are differences in goals, processes are not mapped and not aligned, departments do not know each other's planning and tasks, there are communication issues and tasks take too much time. On top of that, it is not clear who is in control, also partly because there is no centralized policy. Combining these causes shows that there is no harmonization between the different work processes related to ALM and LCM. When harmonization can be achieved between the processes this will positively affect almost all causes (orange box) stated in the cause-and-effect diagram.

3.2 Research questions

The goal of this master thesis project is to create a more harmonized process scenario for the maintenance and development of applications and the infrastructure at SNS. Based on the problem statement it became clear that the harmonization between the different processes is missing. When the processes become more harmonized, this potentially reduces costs. The problem statement and information gained from interviews with the taskforce led to the following main research question:

"How can harmonization between the maintenance and development activities of applications (ALM) and the infrastructure (LCM) be achieved?"

Lankhorst (2009) states there are three aspects that form the basis for redesigning an organization. Sub-chapter 3.3.1 explains these aspects in more detail. In order to answer the main research question, the sub-questions presented below were created:

Q1: What are the relevant aspects for harmonizing ALM and LCM and how can they be evaluated?

Q2: With respect to the aspects, what problems occur at SNS?

Q3: What are the relationships between these aspects?

Q4: What are options for the improvement of harmonizing the processes of ALM and LCM?

Q5: How can a more harmonized future model be implemented?

Within SNS, only the ITC department is taken into account, because this is the leading department when talking about applications and infrastructure. Regulations and methods used companywide are also taken into account. The processes outside SNS are left out of scope, such as processes related to customers or suppliers. If the customers or suppliers are relevant within the processes this will be mentioned, but their own processes are not taken into account. Due to the time constraints on this project, the outcome of this master thesis report are high level harmonization options for the future.

3.3 Research approach

To be able to answer the main research question and the different sub questions the project followed the Regulative Cycle (Van Strien, 1997) that can be found in Figure 5. The focus for this master thesis was on the first three phases of the Regulative Cycle, which includes a validation step of the used methods per phase (returning arrows).

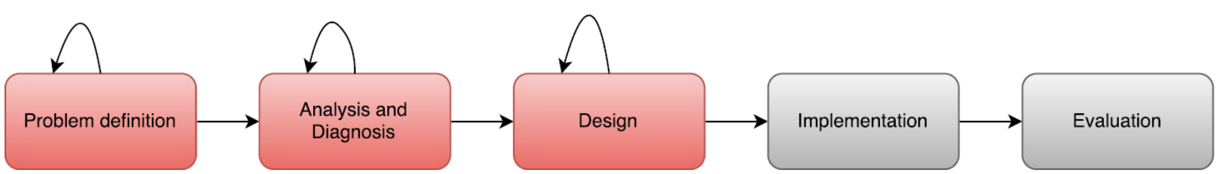


Figure 5: Regulative cycle

3.3.1 Analysis and diagnosis

The analysis and diagnosis phase is the analytical part of the project (Van Aken, Berends, & van der Bij, 2010). During this phase, research methods were used to analyze qualitative and/or quantitative data to diagnose and interpret the problem and gain insight into what causes the problem.

To be able to answer the main research question, information was collected on different topics. Typically, the current system under consideration is analyzed in its organizational, operational and technical setting; problems are pointed out and opportunities are identified (van Lamsweerde A. , 2001). The aspects researched for this thesis were the goals, the processes and the information. These three aspects were selected because they form the basis for redesigning an organization (Lankhorst, 2009).

During the analysis, data on the three aspects is collected, represented and analyzed. The diagnosis was conducted combining the information collected on the three aspects during the analysis. There were no proven best practices found based on the earlier literature review (Appendix C) (van den Boogaart, 2016). Therefore a research method was developed by the researcher (Figure 6). During the diagnosis the relations between the aspects were examined, because besides needing to be aware of the different aspects, one needs to be aware of their interrelationships (Lankhorst, 2009). Figure 6 shows the research method explained. The boxes are the aspects that were analyzed during the analysis phase, and the lines are the relations examined during the diagnosis phase.

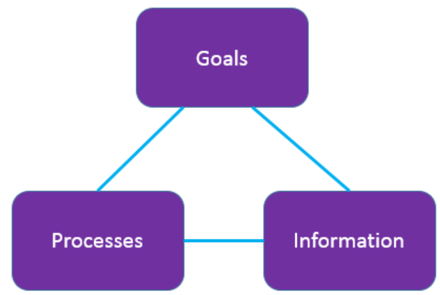


Figure 6: Research method

For both phases (analysis and diagnosis phase), the validation of the data and the created models was done by the taskforce. They stated whether things should change or were forgotten. Based on their feedback the models were updated. This iterative process was continued until the members of the taskforce stated that the models were a good representation of the way they work.

The outcome of the analysis and diagnosis phase was a deep understanding of the context of the problem. After the analysis and diagnosis it became clear what the problems with the current way of working were. Sub-question Q1, Q2 and Q3 were answered, which was the input for the design phase.

3.3.2 Design

During the design phase, a solution for the problem and an associated plan of change were created. For a business problem, this includes a redesign of a work process or organization structure and an implementation method (Van Aken, Berends, & van der Bij, 2010). Based on the features of this phase of the regulative cycle, sub-question Q4 and Q5 were answered.

Based on the outcome of the analysis and diagnosis phase, opportunities for future improvement were determined. These opportunities should lead to a better harmonization of the processes and positively affect multiple causes mentioned in the cause-and-effect diagram. During the design phase, options for change were presented using multiple time frames for implementations. Because change cannot happen overnight smaller steps for implementation make this better manageable. The models were validated by the taskforce that gave the input on the processes of ALM and LCM. They commented on the models and stated what could be changed.

3.3.3 Validation

As explained before, the validation of the different models was done by the taskforce. The members of the taskforce were selected by the company supervisors based of their knowledge of ALM or LCM. Based on the input of the taskforce, the models were validated, so the validation was done based on expert opinion. The arrows above the activities in Figure 5 resemble the validation, because this was done per step and not as a separate step at the end. The validation will not be described separately, but was discussed per section.

4 Analysis and diagnosis

This chapter explains the analysis and diagnosis phase. First a more specific and detailed explanation of the used methodology is given. Afterwards the results of the analysis are presented. Based on these results a diagnosis is conducted and conclusions are drawn. All this information will be input for the design phase.

4.1 Methodology

In this subchapter the method that is shortly explained in Chapter 3 is further specified. First, the data collection method is explained, followed by the method used for data analysis and the diagnosis.

4.1.1 Method of data collection

Multiple sources of information were used to collect information on the three selected aspects. Two methods of data collection were used, namely interviewing and document analysis.

Interviews were conducted to gain insight into the different ways of working. All employees were able to explain their view on the problem and explain their own way of working. Because the harmonization was missing, it was important to collect all the different views. During the interviews employees were questioned on their goals, processes and further knowledge on ALM and LCM. The interviews were conducted with members of the taskforce, of which the selection of the members is explained in Chapter 1. First, individual interviews were held with all members of the taskforce to gain a better understanding of the problems (Appendix D1). Afterwards, interviews were conducted with specific members of the taskforce to get a better understanding of specific sections of the problem. The interviews were semi-structured (questions can be found in Appendix D2) and the most important findings were written down during the interview. The semi-structured interview method was chosen, because these kind of interviews are particularly useful when the research problem refers to a wide-ranging problem area and it is required to detect and identify the issues relevant to understanding the situation (Blumberg, Cooper, & Schindler, 2008). The outcomes of the interviews were processed anonymously, because of privacy reason. Therefore the taskforce members will be referred to by the department they work for.

Secondly, company documentation was collected on the topics, like goals and processes. The company documentation was useful to gain more insight into the different levels of goals and the way of working of some of the members and departments. This documentation was also important to gain insight into the company's strategy.

4.1.2 Method of data analysis

During this phase the collected data was analyzed and categorized in accordance with the three aspects explained in Sub-chapter 3.3.1; the goals, processes and information.

4.1.2.1 Goal model

The goal models were created to gain a better understanding of the work environment at SNS. If goals of different (sub-) departments are compatible, they make working together easier, whereas if they are conflicting they make working together harder.

Goals capture, at different levels of abstraction, the various objectives the system under consideration should achieve. Goals can be used for eliciting, elaborating, structuring, specifying, analyzing, negotiating, documenting and modifying requirements. Goals cover different types of concerns; functional concerns associated with the services to be provided, and non-functional concerns associated with quality of service (van Lamsweerde A. , 2001). The system which a goal refers to may be the current one or the system-to-be. High-level goals often refer to both (van Lamsweerde A. , 2001). Berre et al. (2006) state different aspects that are important when creating a goal model. A goal model only describes the goals of the area of concern, taking into account all different stakeholders. Adding to that, when creating goals, they must be achievable, preferably measurable, not self-evident and have clear and detailed implications.

The goals of SNS were explained in general, after which the goals of the sub-department ITC were given. This led to the goals of the stakeholders. The goals of the stakeholders were linked to the goals of ITC. This was done using the Component and Model-based development Methodology (COMET) (Berre, et al., 2006).

The goal model of SNS was depicted using a balanced scorecard. The balanced scorecard is a strategic planning and management system that is used extensively in business and industry, government, and nonprofit organizations worldwide to align business activities to the vision and strategy of the organization, improve internal and external communications, and monitor organization performance against strategic goals (Balanced Scorecard Institute, 2016). The balanced scorecard includes four dimensions: financial, customer, internal processes and renewal and growth. The balanced scorecard created for SNS for this research has different dimensions (customer, social and environmental); these were already created as dimensions by SNS. The Balanced Scorecard for SNS does not state goals specifically related to the causes derived from the cause-and-effect diagram and can therefore be found in Appendix E. The first discussed model is of sub-department ITC. The balanced scorecard for ITC was created using the four traditional dimensions, because the dimensions created by SNS were not sufficient to cover all goals related to the problem. The most relevant goals for this research are covered by the 'internal process' category, which was not included in the three dimensions of SNS.

Based on the goals of ITC, the relationships between the goals were explained using a goal network. A goal network is a model that shows the relationships between the goals on a high level, represented by a plus or minus sign. In this case the model is the as-is as well as the to-be model, because during the interviews it became clear that the goals are quite generic and could be seen as fixed over time. The most important negative relationships were explained shortly. If goals with a negative relation would be changed, one needs to be careful and see what the effect is on the related goals.

To get a clearer distinction between the goals of different parties within ITC, user goal models were created. A user goal model shows the dependencies between the goals of the actual user and the main goal model (balanced scorecard of ITC). It is important to state what impact changes can have on higher level goals within the organization, because goals are not necessarily compatible (Berre, et al., 2006). Two different user goal models were created; the first one shows the deviation between users that work with ALM (applications) and LCM (infrastructure), the second one shows the division between the goals of I&O and I&C. These divisions were created to see where the goals differ and where they are the same, which could be used when harmonizing the processes in the future.

4.1.2.2 Process model

The current (as-is) situation had to be analyzed to be able to harmonize the processes in the future. Therefore the different processes for the maintenance and development of the applications (ALM) and the infrastructure (LCM) were collected and mapped.

During the analysis at SNS, it became clear that every part of the organization handles their processes related to ALM and LCM differently. To gain insight into the differences and overlap between the different processes, the processes for every specific department were mapped. To create the as-is models based on the input, the Business Process Model and Notation (BPMN) language was used. When using BPMN, standard methods of depicting all information need to be used. The used symbols are explained in Figure 7 (Dumas, La Rosa, Mendling, & Reijers, 2013). The BPMN language was selected, because within BPMN there are different standards available and the one that best fits this research is the collaboration/communication diagram. The relationships and roles, as well as the communication between different parties (pools or lanes), can be modeled on a higher level using this modeling method (OMG, 2011).

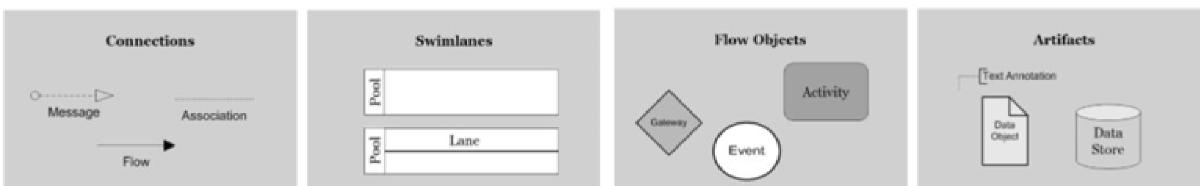


Figure **7: Overview BPMN symbols**

In order to highlight the similarities and differences between the processes, colors were used to indicate how often an action or task was mentioned. The colored boxes can be found in Figures 10 to 17, showing the overlapping steps. When a process step can be found in all generated processes it is colored green, when it can be found in three processes it is colored yellow, in two processes it is orange and in only one process it is colored red. The colors only show whether an activity is mentioned a certain amount of times, and does not say anything about the order of the activities. The order of the activities can be derived from the processes, based on the directions of the arrows between the activity boxes. If the arrow goes from one box toward another this means that these steps follow each other. No method was used to indicate if certain activities always follow each other, this had to be checked manually.

It was also relevant who performs the different steps in every process, this is why RACI matrices were created. The explanation and the actual RACI matrices can be found in Appendix G. The matrices were put in the appendix because they give a more in depth explanation which is not relevant for the design phase. The information from the RACI matrices was used during the implementation of the design.

4.1.2.3 Information model

The final aspect is the information aspect. This is an important aspect, because a lot of the information seen as important is different for every sub-department. The information was mapped using the processes generated during the previous step. Per process step the data that flows through it was mapped. To be able to harmonize the processes, the data had to be harmonized as well.

Information can be different things; here it includes information needed for different steps, updating of different types of information sources, outgoing information, and so on. Based on the mapped information, a data model was created. A data model looks at the aspects the information is about and uses this as a basis for structuring the data. If the things important to the business and the relationships between these important things are correctly identified, a data model can be developed. The goal of the data model is to control duplication to maximize data quality (West, 2011). The resources that are of concern to the business will mostly, but not entirely, be discovered from consideration of the things that have to happen in the business (as contained in the process model) (Berre, et al., 2006).

Data models can be created on three levels of detail. The first level is *the conceptual level*, in which the highest-level relationships between different entities are identified. As can be derived from Figure 8 the only information shown in the conceptual data model are the entities that describe the data and the relationships between those entities. The *logical level,* which follows from the conceptual level, adds primary keys for each entity as well as foreign keys. Normalization starts to occur at this level. The *physical level* represents how the model will be built in the database. All table structures, column names, data types, constraints, primary keys, foreign keys and relationships between tables are given. The physical level follows from the logical level, so all levels are related (Angelov, Eshuis, & Kusters, 2010).

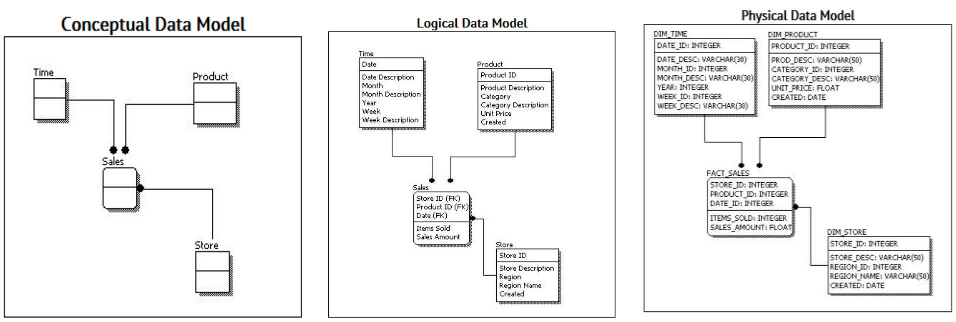


Figure 8: Data model levels (1keydata, 2016)

For this research the conceptual data model was selected to create the as-is data models. There is a lack of information that gets stored, which means that the data model was generated based on the interviews. The conceptual data model shows the relationships between different classes that can have attributes that represent the properties of the class. The language used to create these data models was the Unified Modeling Language (UML), which uses multiplicity constraints to relate the different classes (Angelov, Eshuis, & Kusters, 2010).

I decided to create an as-is data model of the problem domain. By representing the problem domain, all explained data important for the processes could be represented even if it was not stored anywhere.

Based on all this information a short conclusion was drawn that represents the main insights so far. These insights was relevant when there needs to be determined what design option is best.

4.1.3 Method of diagnosis

During the diagnosis, the information collected on the aspects got linked. So no new information was collected. These connections were investigated, because new problems can occur because of these connections. These connections were investigated using different models. First the link between ALM and LCM was investigated, after which the links between the different aspects was explained.

4.1.3.1 Link ALM and LCM
During the analysis phase, information was collected on the goals, processes and information. It became clear that all this data was dividable in information about ALM and LCM. This is why it was important to analyze the differences between ALM and LCM for the three aspects. This was done by comparing the output of the analysis phase for ALM and LCM for all three aspects and stating the biggest differences.

4.1.3.2 Link different aspects
As stated before, the links between the aspects can hold conflicts that need to be solved when creating harmonization between the processes. For each of the connections between the aspects, an analysis was conducted on how the aspects interacted in the current situation. This means that the diagnosis phase consists of the research on the link between the processes and goals, the goals and information and the processes and information.

The link between goals and processes was diagnosed by connecting the user goal model to the different created processes. To be able to say something about the interaction, goals needed to be linked to the processes or the processes needed to be linked to the goals. In this case the goals were linked to the processes, because for every process there was researched what goal it wanted to achieve. The user goal model was used, because this model was created based on the interviews.

The second link is the link between the information and the processes, which was analyzed by using CRUD matrices. A CRUD matrix exists of four values: create, read, update and delete. Within this matrix the entities of the data model are connected to the activities of the processes. By doing so, it can be determined whether information is updated that is never used, or if activities are not related to any of the entities. For every process a CRUD matrix was created, leading to eight matrices. There were only two data models, which meant that the four processes for ALM were related to the ALM data model. The same holds for LCM.

For the interaction between the goals and the information it was important to check whether information was available to evaluate the goals. It was only relevant to check goals related to this research, because the higher level goals, like the once of SNS in general, were not directly related to this research. This meant that the ITC goals were taken into account as well as the goals of the user goal model for ALM and LCM. For each of the goals there was stated whether there was information stored to check if the goal was met or not. When necessary, an explanation was given on what data should be needed to say something about the goal.

4.2 Results
With the use of the methodology described in sub-chapter 3.1, different results were generated. First the results of the data collection were explained, after which the results of the analysis and diagnosis were presented.

4.2.1 Results of data collection
This section describes the results of the data collection and reflects on the process. Both the outcome of the interviews and the company documentation is explained.

The employees were selected by the company supervisors based on their knowledge of the problems with the harmonization of the processes for ALM and LCM. They came from different disciplines and have different viewpoints.

To collect data, a first round of interviews was conducted with all members of the taskforce (Appendix A). The questions were used to get a broader insight into the problems. The general structure of the interviews was the same for every interviewee and can be found in Appendix D1. For the second round of interviews, specific employees were selected based on the first round of interviews. The selected employees work with ALM or LCM and had a clear view on the problems they encounter during these processes. The selected employees can be found in Appendix A with the X in the ALM or LCM column. The selected employees got a mail to prepare their working process before the interview. During the second interview round, different questions were asked by me that can be found in Appendix D2. Based on the questions the most important outcome regarding the processes was that almost all interviewees (9 out of 10) stated that the processes were not general for all employees. In addition ALM and LCM were not aligned. I mapped all different work processes and asked the interviewees to review these processes.

Company information was used as well, but the documents are confidential. For the goal models 'het Manifest' was used. This is a document of SNS that contains the vision for the future and current performance measures. This document also includes the goals of the company, which was used for creating the company goal model. There was also internal documentation (bedrijfsplan) from subdivisions of I&O and I&C used to create the lower level goal models. There were five of these documents (one for each subdivision) available. There were also Excel sheets available with information about every specific system or application, for example end of support dates, whether the systems are planned to be updated and so on. Schedules are made and updates are done based upon these Excel sheets for a particular part of the company. Another, more general document stated what systems are used for different processes. This document is high-level, so holds no specific information per process.

4.2.2 Results of data analysis

During this section the data analysis was discussed. This was done by dividing the topic into the three defined aspects; goals, processes and information.

4.2.2.1 Goal model

The results on the goals based on the methodology described in the previous sub-chapter are presented below. The balanced scorecard of SNS, and related information, based upon the Manifest can be found in Appendix E. As explained, a balanced scorecard for ITC is created that can be found in Table 3.

Table 3: Balanced scorecard ITC

Dimension	*Object*	*Name*	*Description*	*Measure*	*Target*
Financial perspective	F1	Healthy balance sheet	There is a responsible profitability	Return and cost	Cost/Income: < 50%, Cost/Assets: < 75%
	F2	Low cost level	The services (IT and business) are affordable	Costs	As low as possible given that it still meets all standards

	F3	Moderate risk profile	A healthy Tier1 core capital ratio	Return and cost	Cost/Income: < 50%, Cost/Assets: < 75%
Customer perspective	C1	Customer experience	Customer gets explicit attention	Customer survey	Customer satisfaction > 8
	C2	Reliability	Keep promises	Customer survey	Reliability of 100%
	C3	Employee involvement	Win clients trust so that they choose for SNS	Employee survey	Employee satisfaction > 8
Internal process perspective	I1	Continuity	Continuous delivery, so deliver and improve IT services	uptime	Availability of 24x366
	I2	Simplicity	Flawless and tight organized business	Employee survey	Grade of simplicity of processes needs to be >= 8
	I3	Quality	IT services have to be of high quality	Employee survey	Grade of quality of systems needs to be >=8
	I4	Adaptability	Be able to adapt to changes from inside as well as outside	Employee survey	Grade of adaptability of processes needs to be >=8
Renewal and growth perspective	R1	Knowledge up to date	Employees have to be up to date with new innovations	Certificates	All employees have to pass a test on programs they work with

The balanced scorecard for ITC represents a lot of different goals. For example, the internal process perspective shows goals directly related to the problem. The customer in the ITC balanced scorecard is the employee that uses the system within the company, where in the balanced scorecard for SNS the customer was the actual buyer of products from SNS. The customer as defined for ITC can be divided in two ways: 1) the employees working within I&O or I&C, 2) the employees working on applications (ALM) or the infrastructure (LCM). Both divisions have been evaluated, the division between ALM and LCM can be found below and the division between I&O and I&C can be found in Appendix E. The division between ALM and LCM is most relevant for this research, because these are the goals related to the process harmonization.

To show the relationships between the goals of ITC, a goal network (Figure 9) is created. Showing the relationships between the goals is important, because friction between goals can lead to problems with harmonization options for the future. The goal network is created by the researcher in collaboration with the company supervisors.

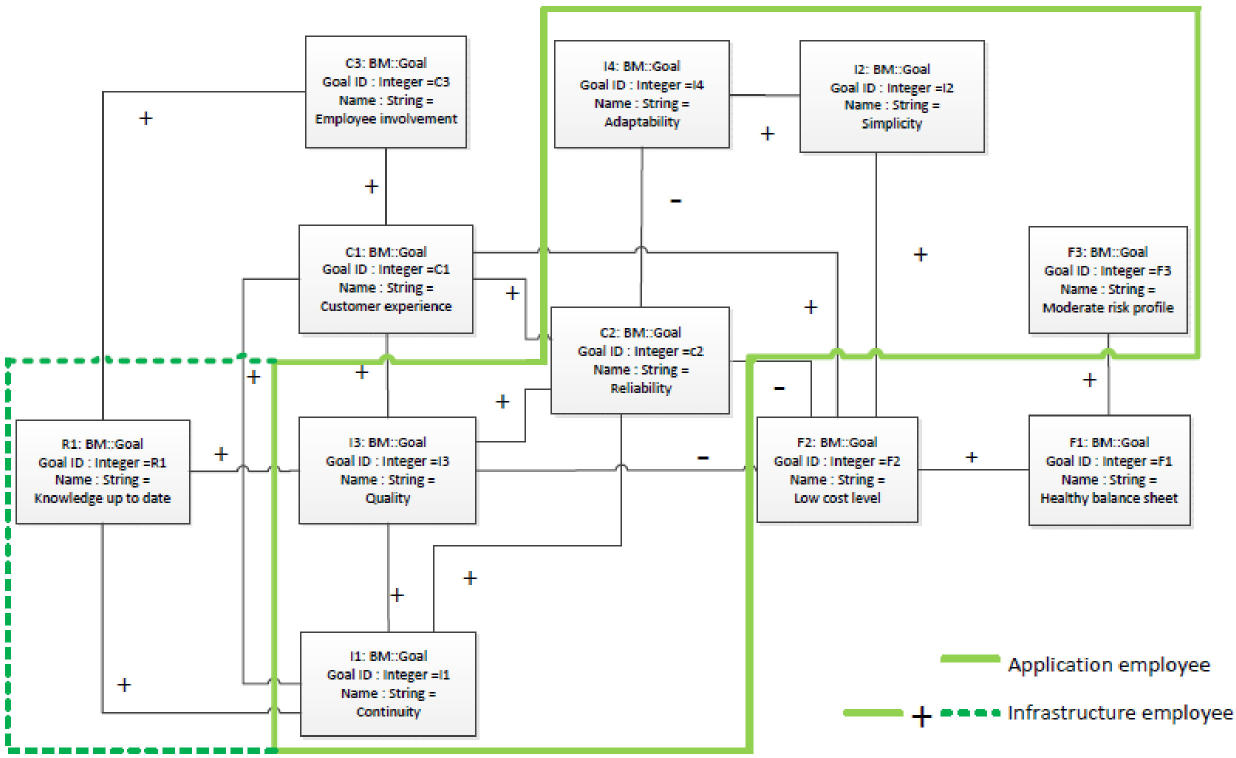


Figure 9: Goal network ALM LCM

From Figure 9 can be derived that most of the goals have a positive connection. All departments have goals that are more important than others, but these goals are different per department. This is why the goals are ranked the same in importance and the goal network becomes symmetric. The negative connections will be further explained, because a negative connection might mean that a solution or idea in one field causes problems in another.

- The first negative relation is the relation between quality and low cost level. To improve the quality, one needs to put more effort into the product which leads to a higher price for the final product. When excellent quality is wished for, low costs will not be realistic. A lot of decisions are now made based on costs. Almost in all cases the cheapest option is selected taken into account the requirements the solution needs to have. SNS has a very cost driven environment, whereas some of the interviewees stated that quality should be the most important aspect.
- The second negative relation is the one between reliability and adaptability. Reliability is often related to stability and a predetermined set of features. However, when a system needs to be able to adapt easily the reliability would become lower. Right now reliability is the most important aspect of the two. As a financial institution, governmental rules apply to the company that need to be met. These rules are reached best when reliability is high and systems are safe (reliable).
- The final negative relation exists between low cost level and reliability. A reliable system comes with higher costs. A reliable system for SNS means that there are no bugs, no data leakages and that it is up and running. To be able to do this, working hours are needed as well as maintenance costs. If a company wants to have a low cost level, this means that reliability cannot be

guaranteed. Right now they will always pick reliability over costs. This means that the costs for the company can rise quickly when problems occur with the systems. From the interviews was derived that the reliability is very important, but that a lot could be done to make the processes and handling of problems less costly.

The relationships between the different goals of ITC are made clear, but the figure also includes colored lines. The colored lines are added to show which goals are more important to ALM and which are more important to LCM. Based on the user goals presented in Table 4, the lines are created. The division between the application employee (ALM) and the infrastructure employee (LCM) is shown this way.

Table 4: User goal model ALM LCM

User	*User goal*	*Goal ITC*
Application employee (I&O and I&C)	**A1: Stay above minimal availability**	I1, I3
	A2: Stay below the maximum permissible outage	I1, I3
	A3: Stay below the maximum loss of data during a calamity	I3, C2, F3
	A4: Vouch for continuity of critical business processes	I1, I2, I4, R1
	A5: Protect systems and processes	C2, I1, I3
	A6: Keep application controls up and running	I1, I4
Infrastructure employee (I&O)	**S1: Vouch for continuity of critical business processes**	I1, I2, I4, R1
	S2: Stay above minimal availability	I1, I3
	S3: Keep all data- and log files protected	C2
	S4: Vouch that every production environment has a backup/restore option	I1, I4
	S5: Vouch for integrity of databases	C2, I3, I1
	S6: Keep all software (versions) up to date	C2, I3, I1, F3
	S7: Keep security measures up to date	I3, C2

As can be derived from Figure 9, there is a lot of overlap between the goals of the application employee and the infrastructure employee at the ITC level. Looking at the user goals there are some conflicts (bold goals in Table 2). For instance, both the infrastructure and the application employee have 'vouch for continuity of critical business processes' as a goal. However, a critical business process can be different for both. For the infrastructure this can be related to the fact that all needed data needs to be retrievable, where for the application this might be related to customers that need to be able to reach the website. For the goal 'stay above minimal availability' the conflict can happen between the two users. To stay above minimal availability is related to a lot of aspects, but in general a department wants to make sure the system is available. Because the infrastructure and the applications are related to each other, this means that the two can interfere with each other. Keeping one system up could lead to another going down.

The validation of the created models was done in two ways. The company supervisors showed the balanced scorecard of ITC to the other department heads and asked them to comment on it. Because the balanced scorecard was created with the use of documentation, not a lot of comments were made. Only the targets stated in the table were changed slightly. Some of the grades were too low, so they were changed to an eight or higher. For the user goal model and de related lines in the goal network the taskforce performed the validation. Based on their input the user goal model was created. They understood the user goal model, but were surprised by the small difference between ALM and LCM in the goal network. After explaining why the lines were this way, they agreed upon this representation and understood why the overlap existed.

4.2.2.2 Process model

The output generated on the processes is presented below. Appendix F shows the processes without color and larger. First the processes generated from the input on ALM are presented. This is done, as explained, relatively shortly, because all activities will be explained in more detail in the sub-chapter on the information aspect. For every activity there is indicated who is in control using the RACI matrix, which can be found in Appendix G.

The colored boxes that can be found in the figures, show the overlapping steps. When a process step can be found in all processes it is colored in green, when it can be found in three processes it is colored yellow, in two processes it is orange and in only one process it is colored in red, see Table 5.

Table 5: Color overview process models

Color	*Found in # processes*
(green)	4
(yellow)	3
(orange)	2
(red)	1

ALM

Process input technical application management – document management

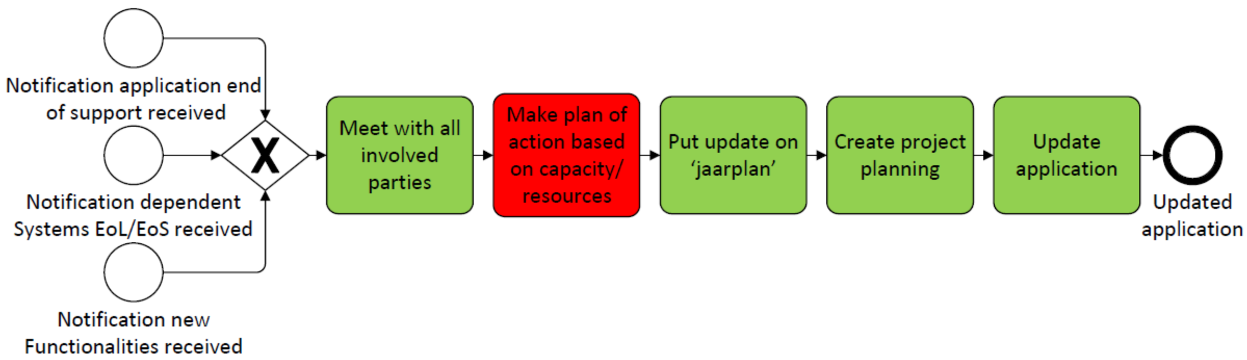


Figure 10: Process TAB document management

Process input technical application management - back office

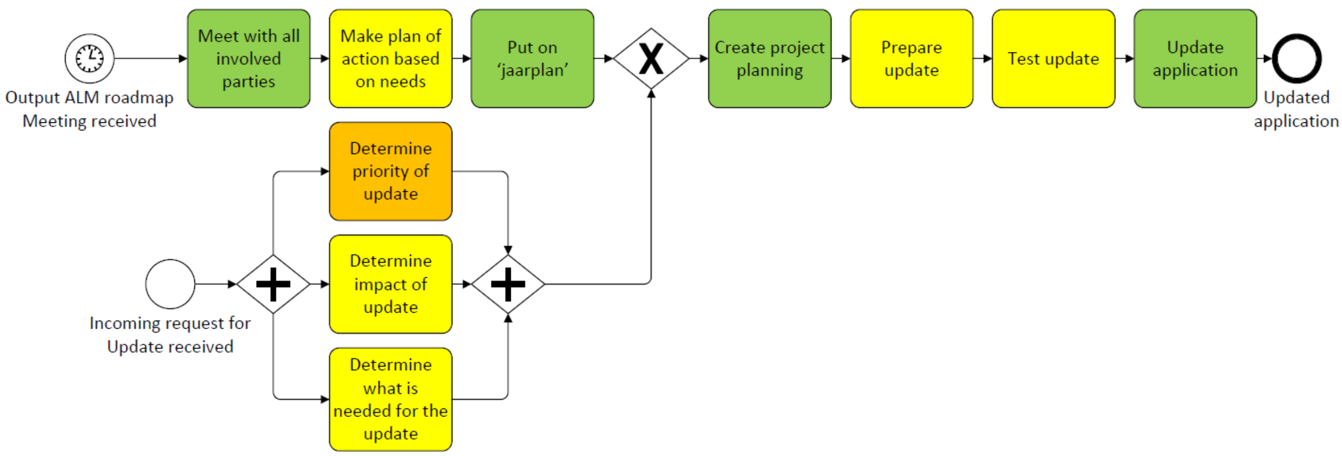


Figure 11: Process TAB back office

Process input technical application management – front office

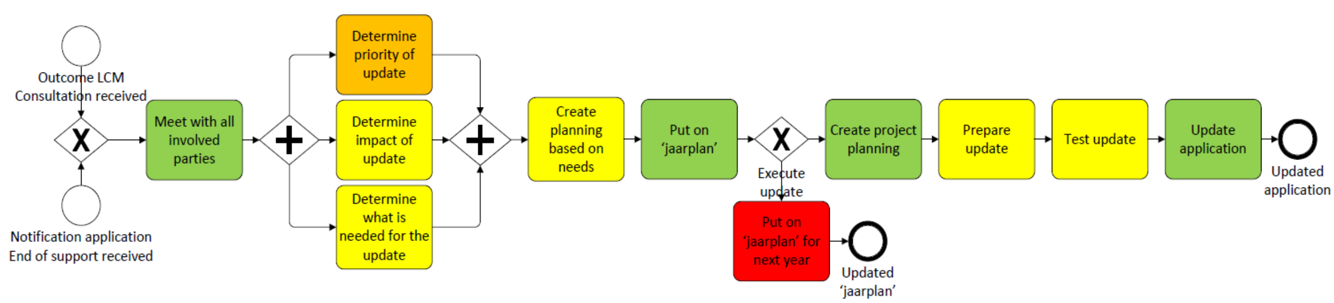


Figure 12: Process TAB front office

Process input technical application management – windows applications

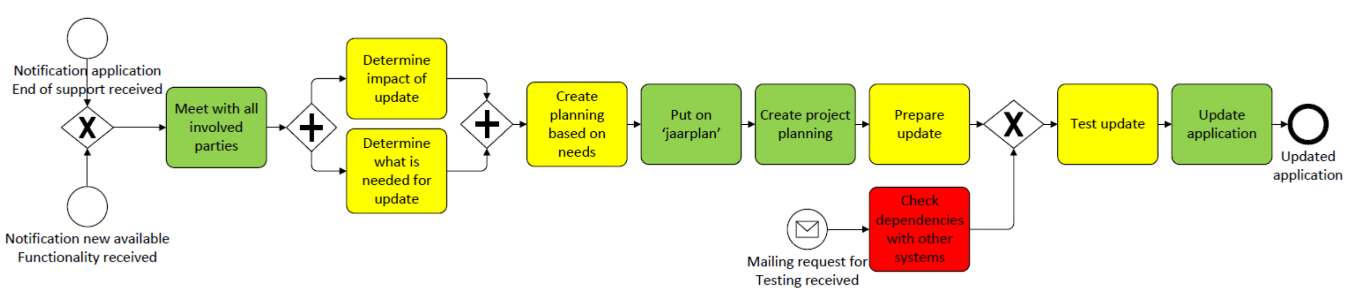


Figure 13: Process TAB windows applications

The input processes for ALM show that the general order of activities (way of working) in the processes is the same for all processes. This can be derived from the green boxes that are in the same order in all ALM processes. In some cases, different steps are taken in between, but the green colored activities are executed by everybody and in the same order. The biggest difference between the processes is the section where they determine the impact of the update, what is needed for the update and the priority of the update. Not all three aspects were mentioned during all the interviews and as can be derived from the figures, determining these aspects does not happen at the same moment during the process. There is also a lot of difference between the triggers defined by the interviewees. Combining this information leads to the conclusion that the processes for ALM already have a general flow. This is convenient for the design, because when the processes look alike, combining them becomes easier.

LCM

Process input infrastructure services

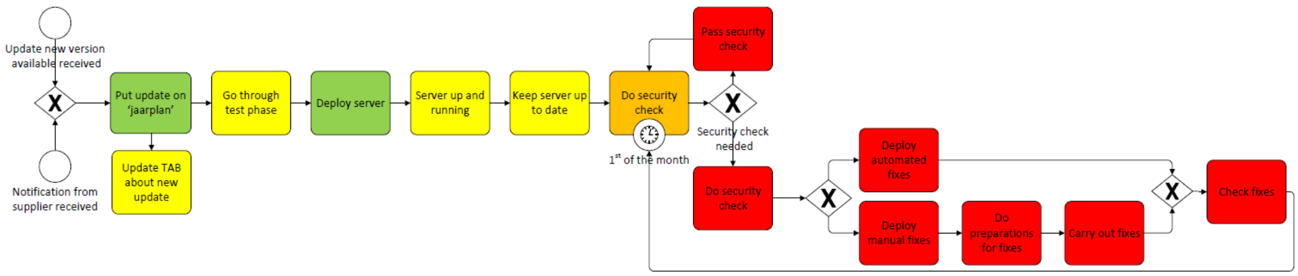


Figure 14: Process infrastructure services

Process input data services – data warehouse

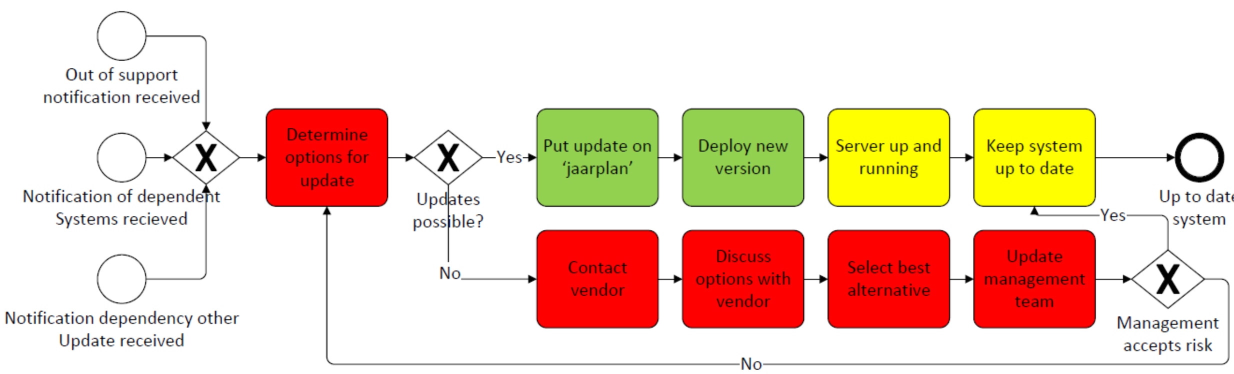


Figure 15: Process data warehouse

Process input data services – database administration

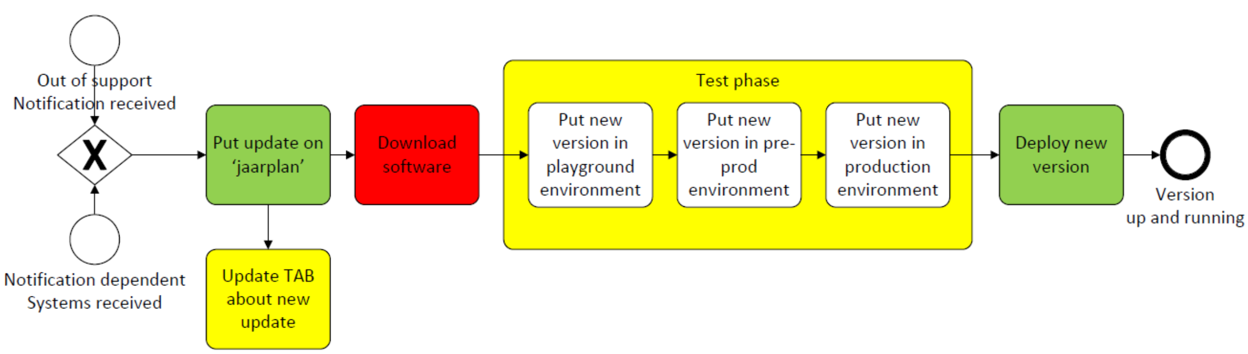


Figure 16: Process database administration

Process input system information management

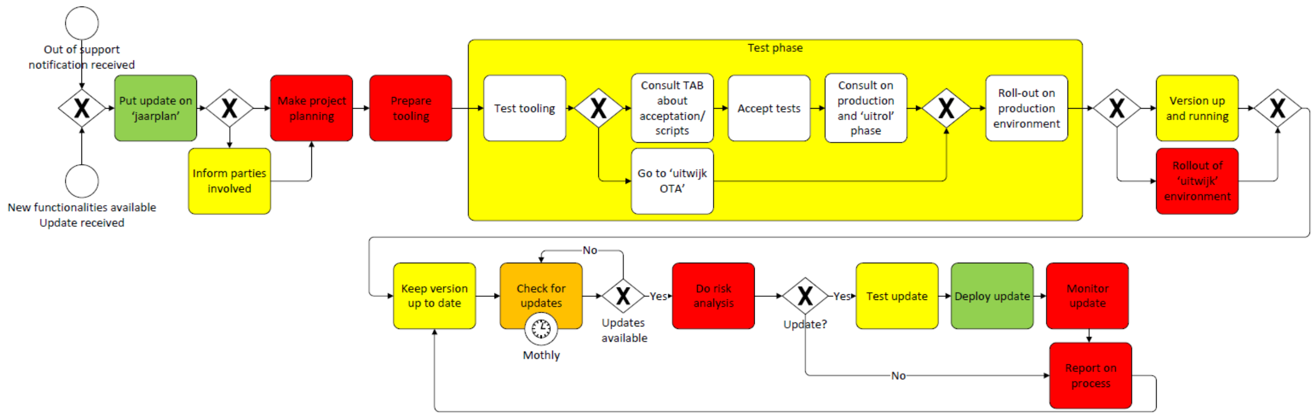


Figure 17: Process system information management

The input on the LCM processes show that there is less overlap between the processes than in the case of ALM. Only two green colored boxes are present, which means that the processes given as an input are quite different from each other. This can also be concluded from the high amount of red and orange boxes. Another difference between the LCM input processes is the number of steps that need to be taken. There can also be concluded that there are a lot of different triggers that lead to starting the process. The limited overlap means that the generalization of the LCM process might be more difficult because the processes are further separated from each other.

The validation of the models was done by the employee that gave input for the model. During the interviews the employee drew the process on a whiteboard and I modeled them in Visio. When the first version of the process was ready this was send to the interviewee for feedback. There were small comments on the terms used for the activities, but because the drawing of the processes was done together no comments were made on the order of the activities or the way the process looks.

4.2.2.3 Information model

The output generated on the data flows using the processes is presented in Appendix H. In the Appendix a detailed descriptions of the different activities per input process are explained. Based on these descriptions a table was created, for ALM as well as LCM, that represents the information mentioned as important. These tables are represented below.

It is important to state that not all information is actually registered and/or stored somewhere. Most of the time the mentioned information is shared and distributed via conversations or meetings and decisions are made then. These decisions are not being put on paper, everybody just knows what is decided. Whether information is stored (in what form whatsoever) is indicated in the table, in the column named 'stored'. For example, the person linked to a certain update is known by everybody, but this is not documented anywhere. This is why, the selected data model is a representation of the problem domain.

ALM: From the process and data input on ALM the important information about the data is given below. To make the table more clear, the names of the departments are abbreviated; TAB document management = DM, TAB back office = BO, TAB front office = FO and TAB window applications = WA.

Table 6: Data ALM

Information	Stored	Where	Named by			
Available resources	No		DM	BO		
Detailed plan of action	No		DM	BO	FO	WA
Employee planning	No		DM	BO		
End of life date	Yes	Excel sheet	DM	BO	FO	WA
End of support date	Yes	Excel sheet	DM	BO	FO	WA
Impact	Yes	RFC (request for change) Word or Excel sheet		BO	FO	WA
Needed resources	No		DM	BO	FO	WA
'Ok' on tests	Yes	RFC test report				WA
Planning based on needs	No				FO	WA
Planning upcoming year	Yes	Excel sheet	DM	BO	FO	WA
Prepared update	No			BO	FO	WA
Priority	No			BO	FO	
Specifics of the update	Yes	Vendor document (website or PDF)	DM	BO	FO	WA
Request to test application	No					WA
Tested application	Yes	RFC		BO	FO	WA
Time period update will happen (Q1-Q4)	Yes	Excel sheet	DM	BO	FO	WA
Triggers for the update	No			BO	FO	
Update type	No			BO	FO	WA
Updated application	Yes	RFC and in application	DM	BO	FO	WA

Based on the table for the information of ALM there can be concluded that most of the information is mentioned by at least 2 out of 4 departments. This means that all mentioned information is important to take into account for the data model. There are two data inputs mentioned by only one party, which are left out of the data model. Based on the input a data model is created (Figure 18).

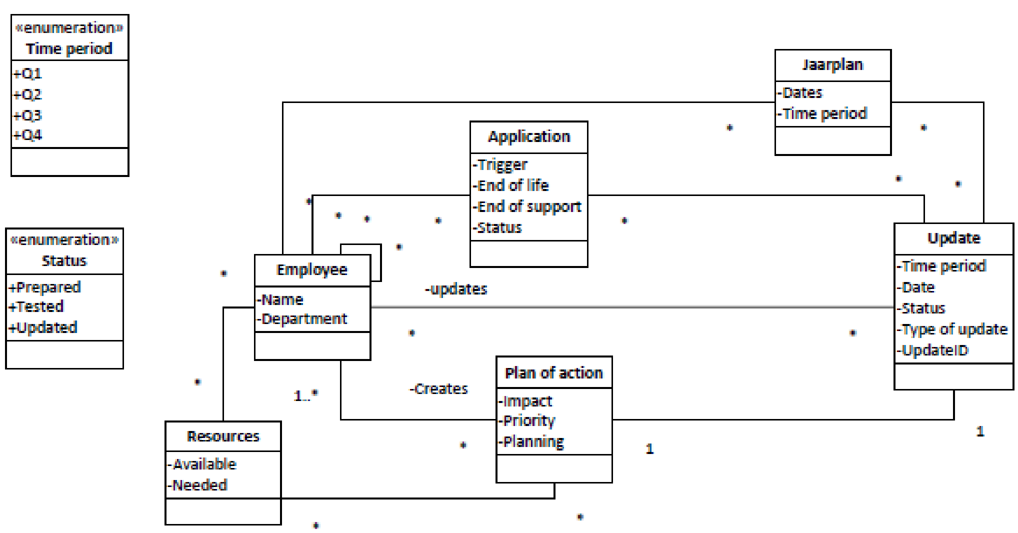


Figure 18: Data model ALM

Based on the data model presented in Figure 18 it seems like everything works as wished for, but this is not the case. The table that presents the data for ALM shows what information is actually stored and what information is not stored. Around half of the mentioned information is not stored anywhere, which means that this information is shared and exchanged via mail, telephone or life conversations and meetings. This is done in good faith, but miscommunications might happen earlier if certain data is not stored.

LCM: From the process and data input on LCM the important information is given below. To make the table more clear the names of the departments are abbreviated; Infra structure services = infra, data warehouse = DW, database administration = DA and system information management = SIM.

Table 7: Data LCM

Information	*Stored*	*Where*	*Named by*			
Check update	No		Infra			SIM
Contacted vendor	Yes	Vendor overview		DW		
Deployed update	Yes	RFC	Infra	DW	DA	SIM
Downloaded fix	No		Infra			SIM
End of life date	Yes	Excel sheet	Infra	DW	DA	SIM
End of support date	Yes	Excel sheet	Infra	DW	DA	SIM
Management team takes risks or not	Yes	RFC		DW		
Occurred problems	No		Infra	DW		SIM
'OK' on al tests	Yes	RFC	Infra	DW	DA	SIM
Planning upcoming year	Yes	Excel sheet	Infra	DW	DA	SIM
Prepared fix	No		Infra			SIM
Prepared software update	No				DA	SIM
Project planning	No					SIM
Risk level	Yes	RFC				SIM
Selected vendor option	Yes	Vendor doc		DW		
Status of fix	No		Infra			SIM
Tested software update	Yes	RFC			DA	SIM
Time period update will happen (Q1-Q4)	Yes	Jaarplan	Infra	DW	DA	SIM
Trigger update	No			DW		SIM
Update on process	Yes/No	Activity plan			DA	SIM
Update ready to test	No		Infra		DA	SIM
Update to TAB	No		Infra			SIM
Vendor options for update	Yes/No	Vendor doc		DW		
Working new version	Yes	RFC and in infrastructure	Infra	DW	DA	SIM

Based on the table for the information of LCM can be concluded that there is much more division between the mentioned information than in case of ALM. There are only seven data points mentioned by all sub-departments and there are six data points mentioned by only one sub-department. From this input the data model is created that is presented in Figure 19.

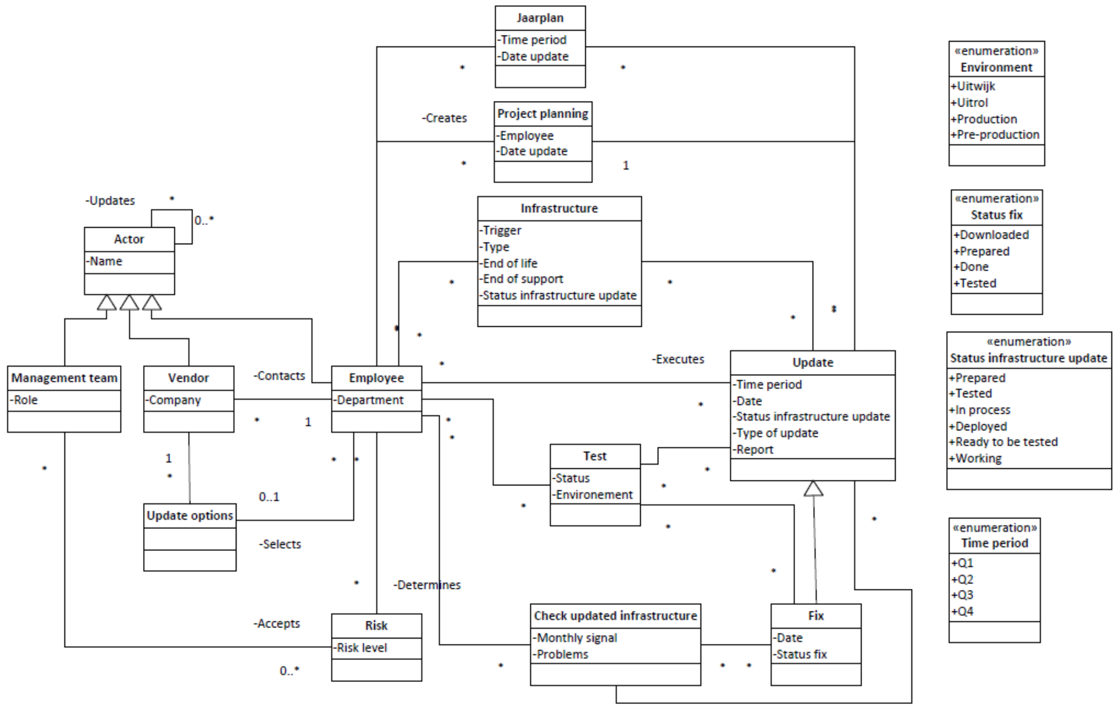


Figure 19: Data model LCM

The first conclusions drawn from the table including the LCM data is that more data is stored, but there is still a lot of data not stored. There are two data points mentioned that are stored by some cases but not stored in others. For instance the update on the process is stored when changes have been made that led to problems or new insights during the process. However, when this does not happen nothing about the process is registered or updated. For LCM holds the same as for ALM, the data that is not stored is known by people because of other contact moments. The data model for LCM has a bigger variety of classes than the data model for ALM, but there are several classes that are the same.

The validation of the data models was done by one of my company supervisors, who is head of all sub-departments related to the databases. He created data models before and uses them for the current work activities. During an review session, the data models were discussed and comments were made. The most important comment was that a number of entities represented in the data model are not present in their current system. After explaining again that this is a representation of the problem domain, the data models were accepted.

4.2.3 Results of diagnosis

This section explains the results of the diagnosis phase. The diagnosis includes the link between ALM and LCM as well as the links between the three aspects goals, processes and information.

4.2.3.1 Link ALM and LCM

Based on the data collected for the analysis phase it became clear there are differences between the goals, processes and information when talking about ALM versus LCM. For each of the three aspects the biggest differences are explained here. This information can be used to draw conclusions.

Goals: The user goal model with the divide between the ALM and LCM employee is the table used for this comparison. On ITC level there was almost no difference between the two, which means that the differences get pinpointed at the user goal level.

The user goals are presented in Table 8 with an indication of how important the goal is for the employee. The importance was determined based on the interviews. The amount of times a goal was mentioned led to the assignment of the number of stars. Stars are used to indicate the importance of the goal. One star is not important, up to four stars for very important.

Table 8: Importance different goals

	User goal	Importance
ALM	A1: Stay above minimal availability	***
	A2: Stay below the maximum permissible outage	**
	A3: Stay below the maximum loss of data during a calamity	**
	A4: Vouch for continuity of critical business processes	****
	A5: Protect systems and processes	*
	A6: Keep application controls up and running	***
LCM	S1: Vouch for continuity of critical business processes	****
	S2: Stay above minimal availability	**
	S3: Keep all data- and log files protected	***
	S4: Production environment has backup/restore option	**
	S5: Vouch for integrity of databases	*
	S6: Keep all software (versions) up to date	***
	S7: Keep security measures up to date	****

As explained before, some goals have the same name but do not mean the same. For both ALM and LCM, the continuity is an important goal. However these two can contradict each other, which leads to a lower continuity for one of the two. Differences between the goals of ALM and LCM are present, which are related to the interpretation of the goals.

Processes: Comparing the processes of ALM and LCM means that the four generated processes for ALM are compared to the four processes generated for LCM. The biggest difference between the two is that the ALM processes consist of one 'update', where the LCM processes have (3 out of 4) some sort of update loop. This means that there can be multiple updates during one LCM process, against only one update for an ALM process. LCM update processes usually take long periods of time (in extreme cases multiple years), whereas an ALM update usually is executed within a couple of months.

Comparing the processes is hard, because there are big differences between the different LCM processes. More overlap exists between the different ALM processes, which suggest that they already have a more standardized way of working. However all interviewees stated that they created their way of working by themselves, without consideration of other departments.

Information: A general conclusion that can be drawn on the information aspect is that parts of the data are not stored anywhere. This can become a problem when parties have to start working together, because part of why and how things are done cannot be explained by data.

The biggest difference between the two is the amount of information stated as important. For LCM more different types of data were mentioned than for ALM. There are data classes mentioned for ALM that are not mentioned for LCM and vice versa. It is hard to state further differences, because the data models are not representing the actual stored data at this point. They represent the environment if all data mentioned would be stored.

4.2.3.2 Link different aspects

Goals and Processes: To say something about the first interaction, the goals need to be linked to the processes or the processes need to be linked to the goals. In this case the goals are linked to the processes, because for every process there is researched which goal it wants to meet. The processes are linked to the goals of the user goal model, because this is the level of goals stated by the actual employees. The bold user goals (Table 9) are contradicting goals for the same process.

Table 9: Link processes and goals

Process	type	User goal application and infrastructure
TAB DM	ALM	**A1**, A2, A4, **A6**
TAB BO	ALM	**A1**, A2, A3, A4, **A5**
TAB FO	ALM	**A2**, A4, **A5**, A6
TAB WA	ALM	**A1**, A3, A4, **A6**
Infra	LCM	S1, **S2,** S4, **S6, S7**
DW	LCM	**S1**, S3, S4, **S7**
DA	LCM	S1, **S2**, S3, S4, S5**, S6, S7**
SIM	LCM	**S1**, S3, S4, **S6, S7**

From the table can be derived that the employees that work with ALM all mentioned application goals and the employees that work with LCM all mentioned infrastructure goals. It can also be derived (Table 9) that every process has contradicting user goals linked to it.

When processes have contradicting goals, the priority of the goals (as determined in Table 8 by the stars) decides which goal is the most important. For ALM this means that A5 (protect systems and processes) is least important. This might seem strange, but for ALM it is more important that everything works than that everything is protected 100%. For LCM it is more important that the software is up to date and secure (S6 and S7) than that they stay above minimal availability (S2).

There exists a difference between ALM and LCM in what is considered most important. The application employee believes availability is most important, where the stack employee believes security and up to date of systems is most important. These contradictions have to be taken into account when harmonizing the processes.

Data and processes: The second interaction is the interaction between the process and information aspect. This is evaluated for all eight processes created after the interviews, because everybody works in its own way. It is still important to remember that not all data is actually stored, but based on the information mentioned as important during the interviews, data models were created.

The processes were linked to the data entities per activity. As explained, CRUD matrices are used to explain this connection. All entities of the data model are represented horizontally against the activities vertically. To explain which activities belong to which process, all processes are numbered.

Number	ALM process	Number	LCM process
1	TAB Document management	1	Infrastructure services
2	TAB Back office	2	Data warehouse
3	TAB Front office	3	Database administration
4	TAB Windows applications	4	System information management

The CRUD matrix for ALM is created first, after which the matrix for LCM can be found. Whether data is stored or not is indicated in the tables by making the letters bold. When a letter is bold, this means this data is not stored anywhere.

Table 10: CRUD ALM

Activity ALM		Entity					
		Employee	Resources	Plan of action	Application	Year planning	update
1	Make plan of action based on capacity/resources		**R,U**	**C**			
1	Put update on 'jaarplan'				C	U	
1	Create project planning			**R,U**		R	
1	Update application			**R**	R,U		U
2	Make plan of action based on needs		**R,U**	**C**			
2	Put on 'jaarplan'				R	U	
2	Determine priority of update			**U**			
2	Determine impact of update			**U**			
2	Determine what is needed for the update		**R**	**U**			
2	Create project planning			**R,U**		R	
2	Prepare update			**R**	R		C
2	Test update						U
2	Update application				U		U
3	Determine priority of update			**U**			
3	Determine impact of update			**U**			
3	Determine what is needed for the update		**R**	**U**			
3	Create planning based on needs		**R,U**	**C**			
3	Put on 'jaarplan'					U	
3	Put on 'jaarplan' next year					U	
3	Create project planning			**R,U**		R	
3	Prepare update			**R**	R		R,U

3	Test update						U
3	Update application				U		U
4	Determine impact of update			**U**			
4	Determine what is needed for update			**U**			
4	Create planning based on needs		**R**	**U**			
4	Put on ‘jaarplan’				R	U	
4	Create project planning			**C**		R	
4	Prepare update			**R**	R		R,U
4	Test update						U
4	Update application				U		U

The CRUD matrix for ALM shows that the actions related to the data look quite similar for all four processes. A lot of information is updated to the plan of action, which is used when the actual update of the application is prepared. Form the table can be derived that the plan of action does not get stored anywhere. Often activities are executed based on experience with previous updates, so specifics for the current update are not stored. This is a problem, because the plan of action is used during almost all steps of the process. The resources are also not saved. This is not a big problem for one process, but when processes get combined this means that resources need to be shared. When there is no insight in these resources and how they are divided, this leads to problems with resource division. There is nothing stored on the entity employee. This entity is necessary for the process, because this is the actor that executes the update and all related steps.

Table 11: CRUD LCM

Activity LCM		**Entity**												
		Employee	Actor	Vendor	Management team	Update options	Risk	Jaarplan	Project planning	Infrastructure	Test	Check update infrastructure	Update	Fix
1	Put update on ‘jaarplan’							U		R				
1	Update TAB about update	**U**	**U**											
1	Go through test phase									**R**	**R**		**U**	
1	Deploy server									U				
1	Server up and running									U				
1	Keep server up to date											R		
1	Do security check									R		R		
1	Deploy automated fixes													U
1	Deploy manual fixes													U
1	Do preparations for fixes													U
1	Carry out fixes									U			U	
1	Check fixes											**R**		

2	Determine options update					R				R				
2	Put update on ‘jaarplan’							U		R				
2	Deploy new version							R			R		U	
2	Server up and running									U			R	
2	Keep system up to date											R		
2	Contact vendor	**R**		**U**										
2	Discuss options with vendor			R		C								
2	Select best alternative					R								
2	Update management team				R	R								
3	Put update on ‘jaarplan’							U		R				
3	Update TAB about update	**U**	**U**											
3	Download software									R			U	
3	Put new version in playground environment										R		U	
3	Put new version in pre-prod environment										U		U	
3	Put new version in production environment										U		U	
3	Deploy new version									U			R	
4	Put update on ‘jaarplan’							U		R				
4	Inform parties involved	**U**	**U**											
4	Make project planning							R	**C**	R				
4	Prepare tooling									R			U	
4	Test tooling										**R**		**U**	
4	Consult TAB about acceptation/scripts	**U**	**U**											
4	Accept tests										R		U	
4	Go to ‘uitwijk OTA’										U		U	
4	Consult on production and ‘uitrol’ phase	**U**									U		U	
4	Roll-out on production environment										U		U	
4	Version up and running									U			R	
4	Rollout of ‘uitwijk’ environment										U			
4	Keep version up to date									R		R		
4	Check for updates											**R**		**R**
4	Do risk analysis						U							
4	Test update										**R**			**U**
4	Deploy update									U	R			U
4	Monitor update									R		R		

In comparison to the ALM processes, there are more process steps that do not use the same entity. This was expected, because comparing the processes created for LCM showed less overlap than the processes created for ALM. The bold letters in the table indicate that the updating of other parties involved in the process is not documented. It is not necessary to store the actual conversation, but who is informed, when and about what would be useful to store. This is important to ensure everybody knows who is involved with the project and what their role is. Again, the project planning is not stored, but it is only used during one process. The fact that this document is not seen as important means that employees go through the steps based on experience. When harmonizing the processes, all departments should agree on one CRUD matrix.

Goals and information: The third and final interaction is the interaction between the goals and the information. It is important to check whether the information is available to evaluate the goals. The goals of ITC and the user goal model with the divide between ALM and LCM are included, because these goals are directly related to this research. For each goal there is indicated whether information is available and an explanation is given why this information is available or not.

Table 12: Link goals and data

Goal model	*Goal*	*Info*	*Explanation*
Application and infrastructure	Stay above minimal availability	No	There is not monitored how much time systems are available
	Stay below maximum permissible outage	No	There is not monitored what amount of time the system is down
	Stay below the maximum loss of data during a calamity	No	It is often not clear what the precise loss of data is after a calamity.
	Vouch for continuity of critical business processes	No	It is not monitored how much time the systems are up or down
	Protect systems and processes	No	There would be information required about hacks, attacks, fraud etcetera to say something about how well the systems are protected
	Keep application controls up and running	Yes	They save the end of life dates and end of support dates to be able to check whether updates are needed
	Keep all data- and log files protected	No	See protect systems and processes
	Vouch that every production environment has a backup/restore option	Yes	During the update this is done during the test phase. When a backup/ restore is needed this information is saved
	Vouch for integrity of databases	No	See protect systems and processes
	Keep all software (versions) up to date	Yes	In the Excel sheets with the end of life and end of support dates. If these dates have passed and no new version is online this means the version is no longer up to date
	Keep security measures up to date	No	See protect systems and processes

ITC	Healthy balance sheet	Yes	The data on the costs and income of SNS are saved, but not only of ALM and LCM
	Low cost level	Yes	Prices for different options of updates are checked when options from the vendor or the needs are discussed
	Moderate risk profile	Yes	When the risks are determined for the updates
	Customer experience	Yes	But this is not saved during the updating or for one of the other causes. This is saved outside of this project
	Reliability	Yes	There is checked whether stored data is the correct data. This is done preventively by the arrangement of the systems and applications
	Employee involvement	No	There is an employee survey every year, but this can be filled in if one wants to. And this survey is not specific for these processes within SNS
	Continuity	No	Availability is measured, but not the failovers of the system.
	Simplicity	Yes	There is stored whether systems differ from the standard
	Quality	Yes	But not specifically for the processes important for this project, so that is why this is not mentioned in the data models
	Adaptability	Yes	By creating standards and staying a small and compact organization
	Knowledge up to date	Yes	This is saved under the employee, because here new diplomas are saved and lessons that still need to be passed are also depicted

From the table can be derived that there are multiple goals that cannot get evaluated by information stored by SNS. At the application and infrastructure level there are more goals that are not evaluated by the information than goals that are measured. For now, the data that SNS stores is not sufficient for drawing conclusions on reaching goals. This does not mean that the information mentioned is not important, but for smoother processes even more and different information might be needed. It is important that data gets stored that can help with the evaluation of goals. When goals get evaluated based on numbers this gives a stronger view than if this is done only based on intuition.

4.3 Chapter conclusion

This section states the conclusions that can be drawn from the information collected in this chapter. The conclusion is divided into two sections. First the conclusions based on the analysis of the data are presented, which are followed by the conclusions based on the diagnosis. Finally a general conclusion of all information is presented.

4.3.1 Conclusions of data analysis

To be able to answer the main research question, information was collected on three aspects. For every aspect the most important conclusions are presented in Table 13.

Table 13: Conclusions data analysis

Aspect	Conclusion
Goals	The user goals with the division between ALM and LCM all relate to the same goals on the ITC level
	Goals of ALM and LCM are conflicting, so they can never both be achieved at the same time
	Identical goals stated the same are interpreted different by different departments
Processes	There is a lot of overlap between the four processes created for the way of working of ALM
	The processes created for the way of working for LCM differ a lot in sort of activities and number of activities
	Between the processes for ALM and LCM there are a lot of differences
Information	Parts of the data are not stored
	Parts of the department data are stored in internal documents (Excel sheets)

The goal of this research is to harmonize the processes of ALM and LCM. The conclusions (Table 13) are used to identify where improvements can be made. The most important conclusions to use are the once related to the information aspect. The missing of data, or it being stored in different ways in different locations, needs to be dealt with to be able to harmonize the processes. Information is really important, because when decisions can be made based on information, instead of experience and intuition, this leads to more insight and a true representation of where things go wrong. This is why the information needs to be included in the design for the future.

The conclusions based on the processes lead to the general conclusion that everybody does what they think is best. For the harmonization it is important where the processes are the same and how they could be combined. The final conclusions are on the goals. For the goals it was stated that they will not change in the future. The stated goals are quite general and thus will be the same. The outcome based on the goals needs to be taken into account when the processes are harmonized, but the aspect itself will not be changed.

4.3.2 Conclusions of diagnosis

During the diagnosis the links between the different aspects of the analysis were investigated. The investigated links are goals and processes, processes and information and data and information. For all these links a conclusion is drawn, taking into account the differences and similarities between ALM and LCM.

Goals and processes: The link between goals and processes shows that one process can be linked to multiple goals. However, these goals can contradict each other. This contradiction happens in almost all processes. This leads to the conclusion that even though the processes of ALM and LCM have the same general goal of updating the system, the sub-goals per process are different. Because the goals stay the same for the design, this conflict might not change with a new design.

Processes and information: The combination of the processes and data was the hardest one, because the data models are still very conceptual and the process for ALM and LCM is described in four different ways. It can be concluded that the processes for ALM show more overlap within their data storage and use than the processes for LCM. Because the data for this link is usually stored in personal Excel files this means that the data can be updated, but nobody sees that this information is updated. To be able to harmonize the processes, the data and the processes should be better aligned.

Information and goals: For the link between the goals and the data it is researched whether the goals stated by the company are measured by the data they store. Because a lot of the data is not stored this resulted in multiple goals that could not be measured by the data. There are goals that are really broad, so it might be that only part of the goal is measured by the data.

4.3.3 General conclusion

Based on the cause-and-effect diagram the goal of this project became the harmonization of the processes related to the maintenance and development of the applications and the infrastructure. Based on the conclusions for the analysis and diagnosis phase, some directions for improvement are determined. The most important change needs to happen to the information aspect. It is important that the data collection gets standardized, for SNS to be able to collect more data on the current processes. The data and the processes are deeply connected, because the collected data dependents on the process the department goes through.

To harmonize the processes for ALM and LCM it is important to keep these two things in mind. With these things taken into account, the design will be created.

5 Design

Based on the conclusions of the previous chapter a design is created for the future. The method for creating the design will be discussed first, which is followed by the results. Finally a conclusion is drawn.

Based on the conclusions of Chapter 3 multiple possibilities for change are selected, but they all have different timeframes. When looking at the different outcomes of the conclusion, the conclusion was drawn that the information aspect is most important to start with. Because no company can change overnight, I decided to create a design that includes multiple steps and starts with looking at the information aspect.

5.1 Method for design

This section explains the method for creating the different steps of the design. This method was created in collaboration with the taskforce and the company supervisors.

From the beginning of the project, SNS wanted to harmonize their processes of ALM and LCM. Based on the information collected during the analysis and diagnosis phase harmonization of the processes could not be achieved. There were many differences between the work processes, and problems with data storage made harmonizing the processes in one step impracticable. This was why a design with multiple steps became favorable. The first step was selected based on the conclusions of Chapter 4 and harmonization between the processes was the aim for the final step.

Based on the conducted interviews with the taskforce for the analysis and diagnosis phase, it became clear that a more general working process for ALM and LCM could contribute to a better collaboration between sub-departments. This led to the three step design for creating harmonization between the processes for ALM and LCM. Each step is shortly explained.

5.1.1 Step 1: Data alignment

Based on the conclusions drawn in Chapter 4, the information aspect was the aspect that could be changed most easily. This change could include storage place and storage format, both were used for the redesign. By standardizing the data and storing it in one central location, communication becomes clearer and employees have more insight into the planning of other departments. Storing data in a standardized way also meant that information could be used for making better supported decisions on where things go wrong.

The information for this step was already collected during the analysis and diagnosis phase. Information collected based on the interviews with the taskforce was used again.

5.1.2 Step 2: Generalized processes

After step 1 of the design the data gets stored in the same way, but employees still have different work processes. This is why step 2 involves the generalization of the processes for ALM and LCM. If employees follow the same standardized process, more data can be collected. This can lead to the use of more advanced data mining solutions, which give more insight into the problems and strongpoints of your processes.

The generalized processes were created based on the processes generated during the analysis phase. The four work processes for ALM were used as input for the generalized process of ALM and the four work processes for LCM were used to create the generalized process for LCM. The generalized processes were created by me. Afterwards the generalized processes were shown to the members of the taskforce who gave input on their work processes (Appendix A). These interviews were conducted one on one, so the comments and remarks could be discussed. Based on the comments and remarks changes were made to the generalized processes until everybody agreed.

5.1.3 Step 3: Harmonizing the processes

With step 2 in place, step 3 is the final step of this design. During this step the processes of ALM and LCM get harmonized. This means that both ALM and LCM have to collect the same sort of data and that their data has to be in one system. Step 3 is the most hypothetical step, because the company, the environment, the customer and so on, could still change over time. The methodology used for the design phase can be found in Figure 20.

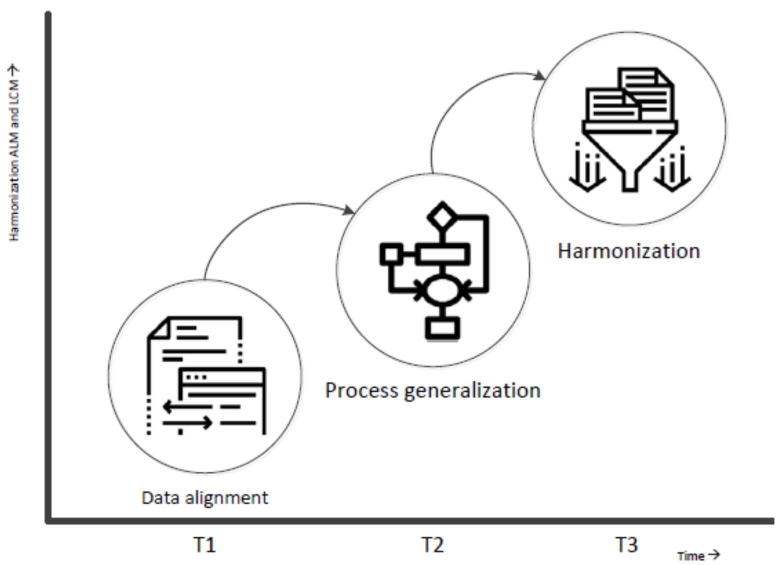


Figure 20: Step by step design method

With this step by step design, SNS has a general guideline for handling their problems with the harmonization of the processes. The outcome of the three steps is explained in the result section.

5.2 Results

Based on the methodology described in the previous section, the results are explained step by step.

5.2.1 Step 1: Data alignment

The first step of change has to be a small step that can be executed in a short period of time. In this case, the information aspect is the one where the biggest improvements are possible within the shortest time frame.

A lot of information mentioned as important is not stored, or stored in department specific documents. To improve this, the data models created during the analysis phase (Figure 18 and 19) should represent the actual stored data. Some entities are the same for ALM and LCM, but have different attributes. Data on all attributes should be stored by the processes, because both processes need different information.

When enough data is stored there can be decided which attributes are most relevant for the harmonization of the processes. With this step nobody has to change their way of working, just the place for storing the data they use. ALM and LCM are not merged in any way, all processes related to ALM store their data in the same place and all processes related to LCM do the same.

Storing the data in the same place does not mean that all problems are solved. Everybody can store their data together, but the data may have a different structure. When the data has a different structure (for the same entity) employees might not understand the data of the other department. It also means that the data cannot be compared easily. To make data comparison easy, all data should be in the same format. For example; one department notes the end of life dates as 10-08-2020, where another department saves it as 10th of august 2020. By registering them in the same format, this would reduce the amount of work when the data is needed for analysis.

Making sure that everybody uses the same format should be stimulated by the company. The manager in charge of this change process has to make sure that the new storage space is easily accessible and easy to use. There can be ensured that data can only be stored in a certain format, so employees are 'forced' to save the data in a certain way. To ensure that the employees understand why this is necessary, they have to be included in the process of creating the new storage. Employees know why data is stored in a specific way, and know how easy or hard it is to change this. By working together, a way of working can be selected that everybody supports.

5.2.2 Step 2: Generalizing the processes

When step 1 is implemented, the data gets stored in a central storage location and the data is stored using the same format. To generate and store more data, the processes for ALM and LCM get generalized during step 2.

To gain more insight in the problems that occur, generalizing the processes for ALM and LCM it is necessary that the collected data is the same for all departments. When more data is collected, more advanced data analysis methods, like advanced data mining techniques can be used. This leads to better insight and opportunities for improvement can be found.

The old processes were used as input for creating the generalized processes. To be able to generalize them, it was checked what the amount of green and yellow colored activities was. For ALM this was the main part of the processes, which meant that these activities were used as the basis for the generalized process. For LCM this were less than half of the activities, which meant that it was harder to create a generalized model. To make the generalization easier, specific parts of the processes sould be simplified before creating the generalized processes (one for ALM and one for LCM). These simplifications are explained in the preliminary steps.

5.2.2.1 Preliminary steps

During the analysis phase the test procedure was explained with different gradations of detail for each process. Afterwards, it became clear that there is a standard way of testing that everybody at SNS needs to follow. This is called the OTAP-street (Figure 21), which stands for development (Ontwikkeling), Testing, Acceptation and Production. The OTAP-street is known by the whole company. There were differences between the mentioned test procedures, because not all steps of the OTAP-street have to be executed for all updates. All interviewees stated that the OTAP-street is the right representation of the test phase for the generalized processes.

So the first preliminary step is the introduction of the simplified activity that states ‘go through test phase’, which includes all steps of the OTAP-street. When executing the process every employee knows which (sub)-steps are necessary for a specific update.

O	T	A				P	
→	→	→			→		
O	T	R1	R2	QF	PP	P	U
Ontwikkel	Test	Acc	Acc	Quick Fix	Pre-Prod	Productie	Uitwijk
Security Baseline van toepassing							
Ontwikkeling		DBA					

Figure 21: OTAP-street

The second preliminary step is related to the mentioned inputs. A generalized process would become confusing and unreadable if all mentioned inputs have to be depicted. Therefore the triggers were divided into categories that represent all mentioned triggers. As can be derived from Table 14 , these generalized triggers are the same for ALM and LCM, but hold different mentioned triggers. The selection of the generalized triggers was discussed with the taskforce, which led to these three generalized triggers.

Table 14: Generalized triggers

Generalized trigger	ALM	LCM
Functional	New functionalities Dependencies with other systems	New version available Notification dependencies other systems (2x) Better functionalities available
Technical	Application end of life (2x) Application end of support (3x)	Notification supplier about end of support (3x) Current version out of support
Ad hoc	Issue fixing	Issue fixing

5.2.2.2 The generalized processes

The generalization of the process for ALM is explained first, followed by the generalized process for LCM.

ALM: Based on the explained steps, a first version of the generalized process for ALM is created (Figure 22).

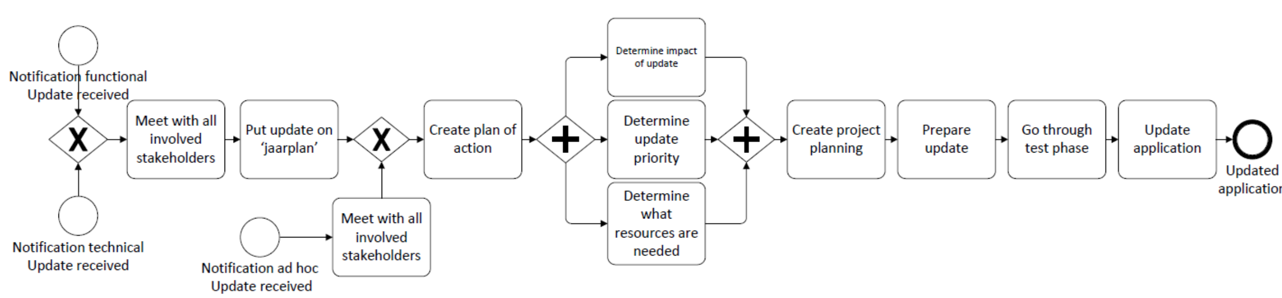


Figure 22: First version generalized ALM process

This model was reviewed by the taskforce. Based on their comments the final version of the generalized process for ALM was created (Figure 23). Because the processes for ALM already had a lot of overlap, the mentioned changes are small (see colored box in Figure 23). Larger images of the processes can be found in Appendix I.

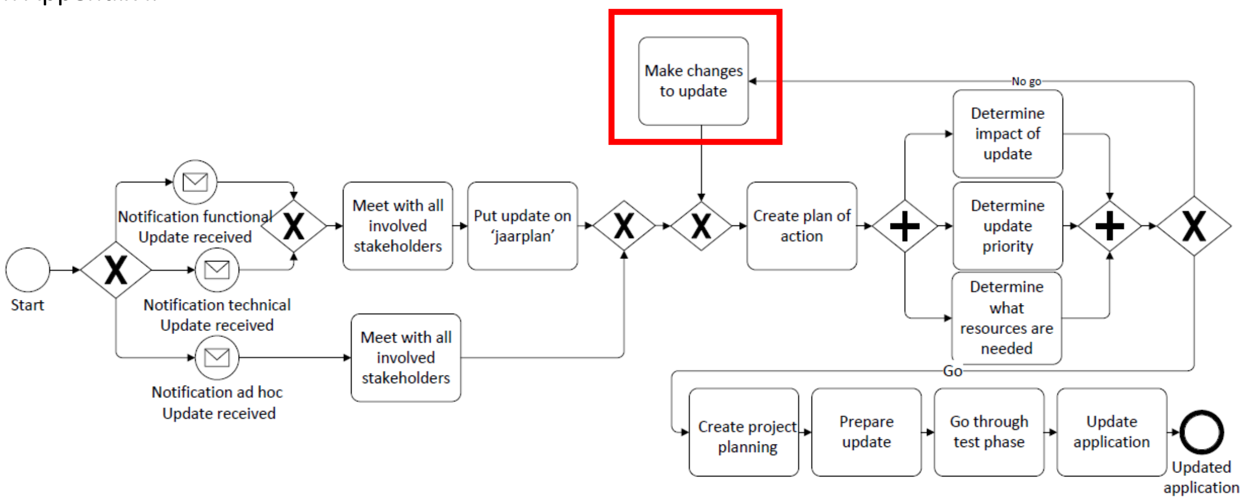


Figure 23: Final generalize ALM process

The only change added to the process is the loop back from after determining the update, priority and resources needed. There is a check to see whether all values are ok. If this is not the case, changes to the update are made and a new plan of action is created/the old one is updated. The interviewed members of the taskforce agreed that Figure 23 is a good representation of the process for ALM.

The changes made to the process have effect on the other aspects and the links between the aspects. These effects can be found in Appendix J.

LCM: Based on the explained steps, a first version of the generalized process for LCM was created (Figure 24). From the preliminary step can be derived that the ad hoc trigger has no link to a process step. To make sure there really is no ad hoc trigger present, this was questioned during the next round of interviews.

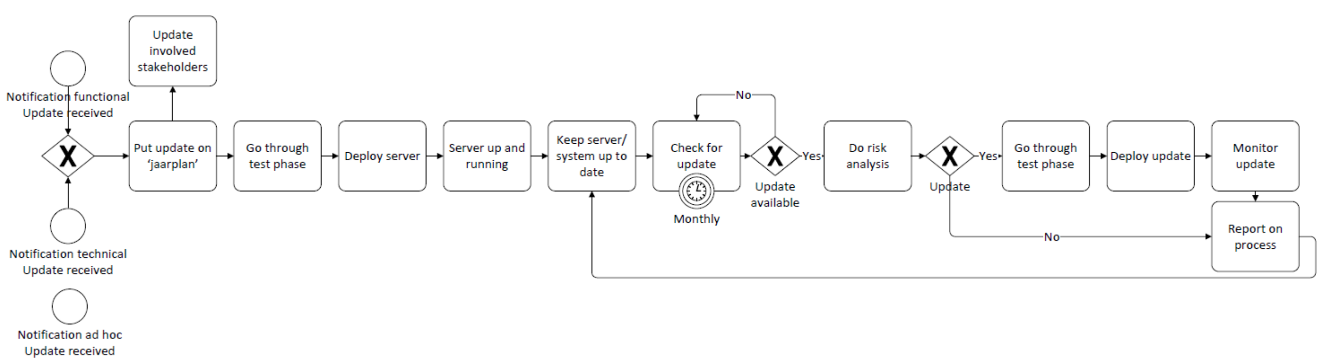


Figure 24: First version generalized LCM process

This model was reviewed by the taskforce. Based on their comments the final version of the generalized process for LCM was created (Figure 25). Larger images of the processes can be found in Appendix I.

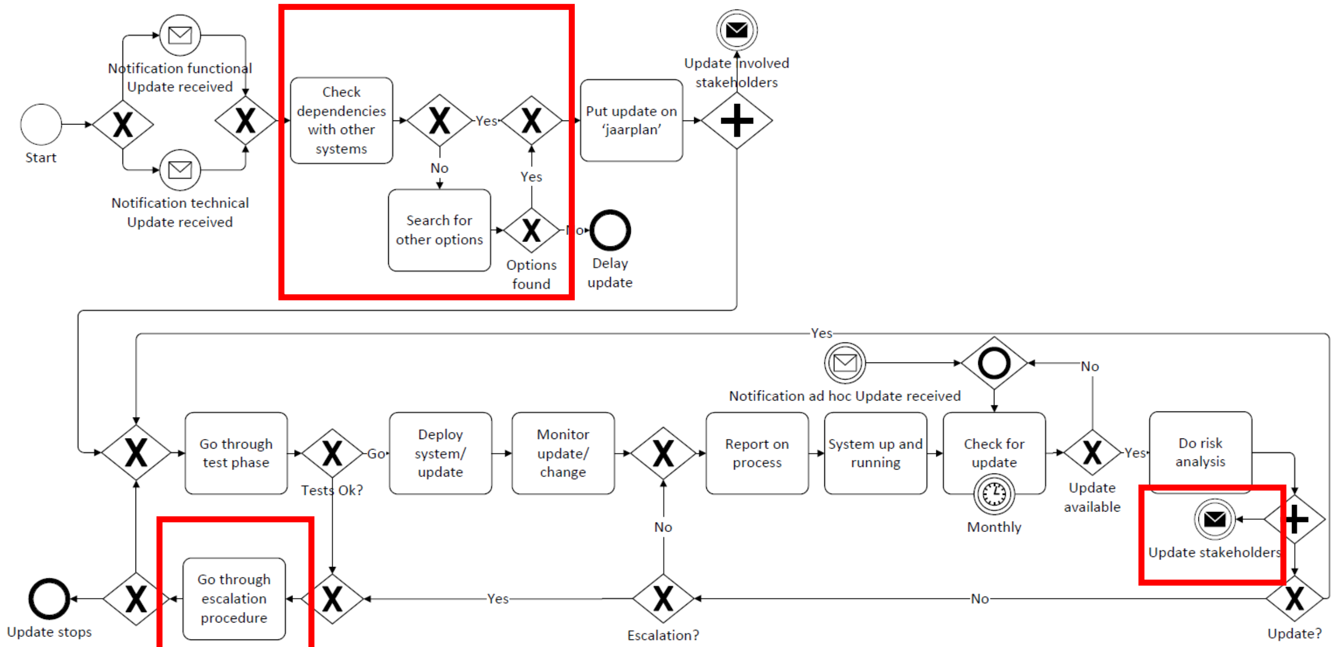


Figure 25: Final generalized LCM process

By looking at the two processes it becomes clear that multiple changes were made to reach the general process for LCM (see colored boxes Figure 25). The first change is related to the first step of the process. After a trigger comes in, the dependencies with other processes have to be checked. If this check is 'ok' the process continues as before. If this is not the case, other options are searched for the update. If these options are found the process continues as before, but if these are not found the process ends. This means that the update will be delayed. The second aspect mentioned by all interviewees was the need for an escalation procedure. This can be different for every specific case, but usually includes the acceptation of some sort of negative consequences (risks). This can occur when the update is not done completely or the risk of doing the fix is too high. During this step the management has to accept the risk and has to manage to keep an 'up and running' system. The final important aspect mentioned was that stakeholders also need to be updated during the monthly cycle of updates, so this was added. The interviewed members of the taskforce agreed that Figure 25 is a good representation of the process for LCM.

The changes made to the processes have effects on the other aspects and the links between the aspects. These effects can be found in Appendix J.

When step 2 is implemented all departments working with ALM have the same basic work process and the once working with LCM also have a basic work process. The data models are updated to enable more advanced data analysis methods (Appendix J).

5.2.3 Step 3: Harmonizing the processes

Step 3 includes the harmonization of the processes for ALM and LCM. Until now the interaction or communication between the processes was not included in the design in any way. During this final step of the design the interaction between the processes of ALM and LCM was included.

To be able to combine or connect the two processes, overlap between the processes needed to be found. This information was collected based on the earlier conducted interviews with the taskforce. Explanations of all activities can be found in Appendix H and during the creation of the generalized processes more information was collected. If it was unclear whether activities were alike this was questioned during the interviews. The processes were combined with these findings in mind. The processes created during the analysis phase were based on a one year time period, this was done for step 3 as well.

Both processes begin with 'meet with stakeholders' and 'check dependencies with other systems'. These steps can be seen as equal, because during the meeting with the stakeholders the dependencies and interactions with other systems are evaluated. When the processes interact, this means the employees of ALM and LCM have to check dependencies with each other. Afterwards, both employees for ALM as well as for LCM put the update on the 'jaarplan'. When the processes are connected, this means that they update the same 'jaarplan'. It can therefore be concluded that the activities of both processes can be done together. We are only talking about the triggers in the categories functional and technical, because the ad hoc triggers are not included on the 'jaarplan'. The creation of the 'jaarplan' happens once a year.

The second change of the new process is that both parties need to keep each other updated during the process. They also work together on certain updates, because the 'jaarplan' is combined. With better communication it becomes possible to be more efficient during updates and help each other. If everybody knows what the others are doing, this generates more understanding and insight. With the changes mentioned, the combined process is generated that can be found in Figure 26. The detailed view of the process ALM and process LCM activities can be found in Figure 27.

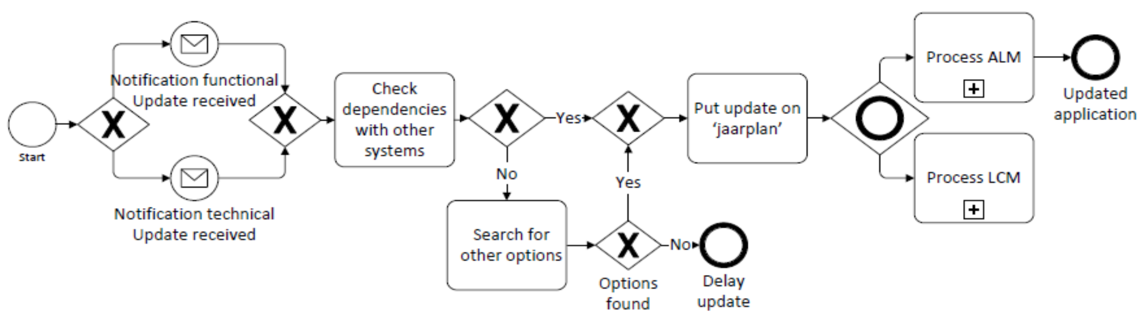


Figure 26: Harmonized process design

Figure 27 shows the communication between ALM and LCM. LCM needs to update ALM every monthly review cycle to make sure all applications have the correct version when (a part of) the infrastructure is updated. ALM needs to update LCM when they created their project planning. The LCM employee can then check if all requirements are met for the update of the application. If an infrastructure component does not meet the requirements, this change is included in the next monthly update cycle.

When SNS implements this process, issues in communication and other soft skills necessary for collaboration need to be evaluated. Because during different interviews was stated that the collaboration with other parties often was hard.

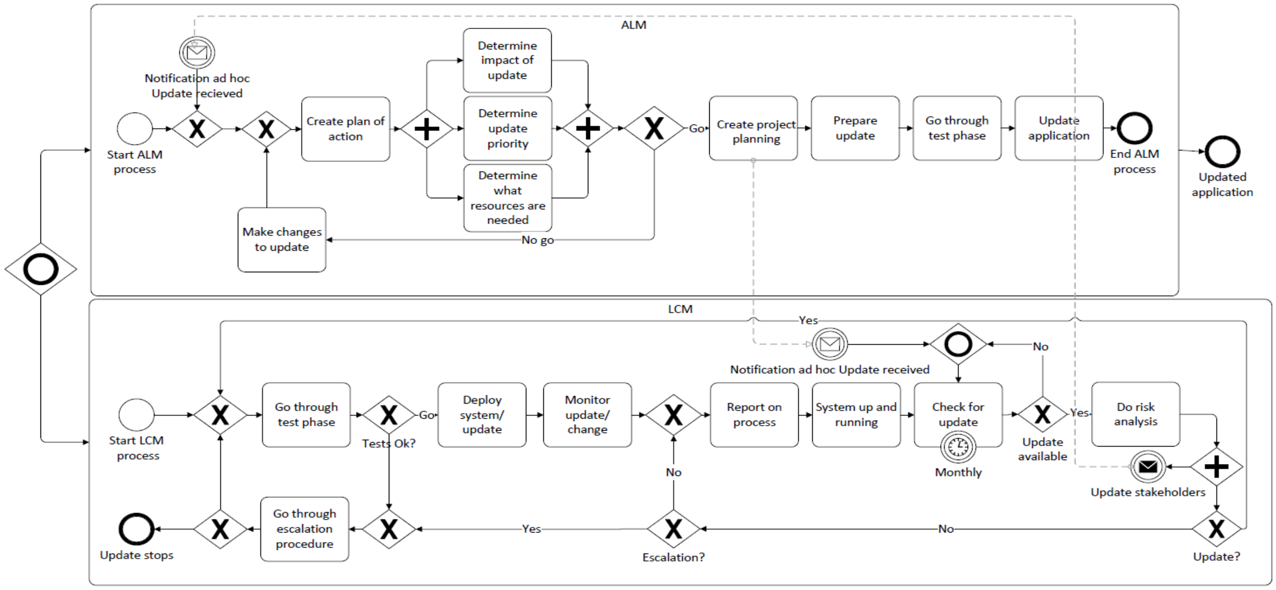


Figure 27: Detailed part harmonized process design

For SNS to reach step 3, a lot needs to happen. It is still a question if step 3 will ever be useful in this form, because the company can still change a lot before step 3 would be even considered. Not changing and adapting would be even worse, so this is a good direction for change to pursue.

5.3 Conclusion

Step 1 of the design proposes to change the information aspect. During this first step only the information is changed, to make sure that this step can be executed in the near future. When implementing the data models, this means employees can see what others are working on and check whether this influences their own work. By storing all data, it also becomes possible to make decisions based on facts and information and not on intuition.

Step 2 of the design proposes to change the process aspect. The new design of the process is a generalization of the processes for both ALM and LCM. Consequently, the data model was changed in order to facilitate central data sharing, which is required for the proposed process changes. The goal model was not changed as it was required that the goals stay the same in this research. Regarding the links between the aspects, only the link between process and data changed, resulting in a single CRUD-matrix.

The benefits of the proposed changes solve or improve causes from the cause-and-effect diagram. For example, 'non-alignment of ALM an LCM processes'. This way, the proposed changes contribute to better/easier maintenance and development of applications and the infrastructure.

Step 3: A process is developed where ALM and LCM are combined. To be able to combine or connect the two processes, it is important where overlap exists between the processes. For this design other aspects are also important. When people need to work together their soft skills become important too.

For the company to be more adaptable and efficient, more data needs to be stored in a central location. When SNS starts storing more data by generalizing the processes, a lot can be done. With the use of data analysis methods like advanced data mining techniques, problems can be derived from the data. Analyzing this information provides the company with a working method that is set. This means that analysis of multiple periods can be compared and intuition is no longer leading in making changes.

Within SNS the design leads to multiple improvements besides the harmonization of the processes for ALM and LCM. After step one, actual data gets stored, which can be used for problem solving. When the general processes get implemented this leads to more data collection and more insight into where things go wrong. When the basic processes are the same, determination of bottlenecks and strongpoints gets easier. This leads to change based on numbers instead of intuition. It makes prioritizing changes easier, because costs and benefits can be quantified.

SNS is happy with the outcome of the research. They believe that if nothing changes, the design would work for harmonizing the processes of ALM and LCM. The feedback of the taskforce and the company supervisors was positive. They believe that the most important problems were taken into account and fixed within the design. This report is going to be used to as a starting point for change for different projects related to ALM and LCM within SNS.

6 Conclusion and discussion

The conclusion and discussion chapter is the final chapter of this research. First the conclusions are explained by briefly returning to the main research question of this research. Afterwards the discussion section includes the relevance for research and practice, the limitations of the research and the recommendations for future research.

6.1 Conclusion

During this section the research question formulated in Sub-chapter 3.2 gets answered.

Research question: *How can harmonization between the maintenance and development activities of applications (ALM) and the infrastructure (LCM) be achieved?*

This study developed a multiple step design for harmonizing the processes for ALM and LCM within SNS. The processes were created using the input of the taskforce, they also validated the designed process steps. The detailed design cannot be used by other companies than SNS, but the general steps and procedures can be used to investigate related problems.

During the analysis and diagnosis phase the goal was to identify the differences between the processes for ALM and LCM based on the three selected aspects and their links. Interviews were conducted with members of the taskforce. The information from these interviews was used to create different models that gave insight into the differences between the aspects. Based on these differences, different options for improvement were found. The information aspect was selected as a starting point for the design, because this change would be relatively easy for the company and lead to a big improvement towards more harmonized processes.

The first step of the design is the *data alignment*. During this step all important data gets stored using the same format. This means that the data models created during the analysis phase become the actual data models. Once the data would be standardized using data models, it becomes important to store as much data as possible. To be able to do so, the work processes need to be the same. When work processes are the same, more data can be collected per process step, because they all go through the same process steps. This is why step two of the design, *process generalization,* includes the generalization of the processes for ALM and LCM. The generalized processes were created by the researcher and reviewed by the taskforce, until the taskforce stated that the new process represents everybody's work processes.

Both steps of the design lead to harmonization between the processes, but the real *harmonization* happens during step three of the design. Here the generalized processes created in step two are linked to each other. Based on the conducted interviews it became clear that working together on the 'jaarplan' would really help with gaining more insight into each other's work. One general process was created, but this did not mean that both processes were totally merged. The taskforce and the company supervisors stated that merging them is not possible jet, because the processes are too different from each other. The differences are in sort of activities that need to happen, as well as in length of the update. They did state that the harmonized process created by the researcher is a point they want to work towards.

SNS is happy with the outcome of the research. They believe that if nothing changes, the design would work for harmonizing the processes of ALM and LCM. The feedback of the taskforce and the company supervisors was positive. They believe that the most important problems were taken into account and fixed with the design. By creating a step by step design they believed that the implementation was doable and reachable. It is important to keep in mind that the design is created based on current information. Because the implementation of steps one and two takes some time, it is important that the changes to the data get included in the design that will eventually be implemented. Not only the data can change, but also the environment itself. The financial sector is a sector that has changed rapidly over the last years and needs to change more towards the future, because customers need to become the core focus of financial institutions (De Nederlandse Bank, 2015). These changes also have to be checked to adapt the created harmonized process. Per step and per change SNS needs to adapt its work processes and see how this influences the outcome.

6.2 Discussion

This section first explains the relevance of the research, after which the limitations of the research are discussed. Finally recommendations for future research are given.

6.2.1 Relevance for research

Based on the literature review (van den Boogaart, 2016) conducted as a preparation for this research it was concluded that there is not a lot of information available on ALM. How application lifecycle management can be implemented is still not clear. And ALM has no standard format for every company to implement, which makes it impossible to make a general claim on how to implement ALM based on the current available literature.

For this research ALM is combined with another process, namely LCM. Combining ALM and LCM has not been done a lot in literature until now, and has never been done in the financial sector. Furthermore, the problem SNS faces is one all financial institutions struggle with. With this research a direction is given for tackling problems between the harmonization of ALM and LCM. Most relevant for other research is the method of analyzing the data and what data is important to analyze.

The methodology used for reaching the design can be used by other companies that face the same problems. Other financial companies can use this method for identifying their own problems with the harmonization of processes related to applications and the infrastructure. This methodology can also be useful for companies in other sectors like the service sector. On can think of the telecommunication sector where services get sold online and apps are used for updates. For companies like this it can also be a problem to harmonize the processes between the infrastructure and the applications.

6.2.2 Relevance for practice and recommendations

This research is practically relevant to organizations in multiple ways. This research provides a method of analyzing the information available at the company to gain insight into where things go wrong. In doing so everybody knows where things go wrong, also things that might not come to mind immediately. Based on the findings, a more grounded direction can be selected for improvement in the future.

Besides the general contribution, this study provides SNS with information on where their problems occur. They gained insight into where problems within the company occur based on models and information, not based on intuition. They also learned that using models and research methods for finding problems leads to more understanding and intentions to tackle some of these problems. By creating an iterative implementation plan for the harmonization of the processes, it became clear how they could actually implement the changes. They can use this way of thinking for other projects or to extend this project in the future.

It is recommended that SNS starts implementing the first step of the iterative design, in which the data gets standardized. There should be tested whether this new method of storing data leads to problems with employees, security or other parties. It is important that besides this, I&O tries to involve I&C with the project. The processes of I&C can then be added to the problem, which can lead to changes in the second and third step of the design. When the processes of I&C become clear, the data storage might need some changes. After this is done and tested, the second and third step of the design can be implemented. It is important that before the processes get implemented, there is investigated how the high level design can be translated to the actual working processes for each specific sub-department. Sub-departments should be involved with this process to make the willingness to change higher.

The recommendations made here might only be part of the solution, because in the future more integrated design options might be possible. The financial sector is changing rapidly to a sector where the customer is key. This needs changes in the organizational model, the motivation and the company culture (De Nederlandse Bank, 2015), which might influence the design created for this research.

6.2.3 Limitations

In this section the most relevant limitations of the research are discussed.

- The first limitation is the generalizability of the research. The research is conducted at SNS, which is a relatively small financial institution in the Netherlands. The validation of the models was done by expert opinion from the company, which does not mean that this is the way other companies work. To be able to generalize the research, information should be collected on the same aspects and processes in other companies to see what the overlap and differences are.
- Interviews were only conducted with employees of I&O. Employees from I&C were asked to collaborate with the project, but they did not see the need. If their side of the story would have been included in the research, the outcome could have been more divided. Certainly for the processes of ALM, where only I&O employees elaborated on, but this does not mean that the employees of I&C have the same way of working.
- Due to time restrictions, the created harmonized design is on a high level. This means that there is no method for working created for all sub-departments. The high level harmonization process is a first step for implementation, but employees of all sub-departments need to be involved to get the details clear.
- In this research the processes for ALM and LCM are investigated separately from other processes important at SNS. In this research the employee only needs to work on either ALM or LCM, but there are related tasks they have to do that can influence the processes. It could happen that the outcome of an update for ALM or LCM leads to conflicts with other projects or tasks that need to be done.

6.2.4 Future research

In this section some recommendations are given for future research.

- It should be researched whether the steps taken for creating the future design can be used in other financial companies to develop a more generalized procedure for handling these types of problems. One could use similar size companies or larger companies and see what the differences are and how this influences the outcomes of the research.

- From the literature review conducted as a preparation for this master thesis it became clear that there is not a lot of research available on the topic of ALM. It would be interesting to conduct a broader research on the topic to be able to create a general framework for implementing ALM. It would also be interesting to see how ALM interacts with other common company processes and how these processes interact with each other.

Bibliography

1keydata. (2016). *Data modeling - conceptual, logical and physical data models*. Retrieved from 1Keydata: http://www.1keydata.com/datawarehousing/data-modeling-levels.html

Angelov, S., Eshuis, R., & Kusters, R. (2010). *Infromation modeling.* Eindhoven: TUE/Faculty IE&IS.

Artto, K. A. (1994). Life cycle cost comcepts and methodologies. *Cost manage*, 28-32.

Balanced Scorecard Institute. (2016). *Balanced Scorecard Basics*. Retrieved from Balanced Scorecard Institute; Strategy Management Group: http://balancedscorecard.org/Resources/About-the-Balanced-Scorecard

Bardhan, I. R., Demirkan, H., Kannan, P. K., Kauffman, R. J., & Sougstad, R. (2010). An interdisiplinary perspective on IT servicies management and service science. *Jounal of Management Information Systems*, 13-64.

Behara, G. K., & Inaganti, S. (2007). Approach to service management in SOA space. *Business Process Trends Journal*, 1-9.

Berre, A.-J., Elvesaeter, B., Oyvind, A., Oldevik, J., Solberg, A., & Nordmoen, B. (2006). *Component and Model-based developement Methodology (COMET).* SINTEF ICT.

Blumberg, B., Cooper, D. R., & Schindler, P. S. (2008). *Business Research Methods.* New York: McGraw-Hill Companies.

Chappell, D. (2008). *What is application lifecycle management.* Chappell & Associates.

Colomo-Palacios, R., Fernandes, E., Sabbagh, M., & Amescua Seco, A. D. (2012). Human and intellectual captial management in the cloud: Software vendors perspective. *Journal of Universal Computer Science*, 1544-1557.

De Nederlandse Bank. (2015). *Verandervermogen in de financiele sector.*

Doyle, C. (2007). The importance of ALM for aerospace and defence. *Embedded system engineering magazine*, 28-29.

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2013). *Fundamentals of Business Process Managment.* Berlin: Springer.

Felic, A., König-Ries, B., & Klein, M. (2014). Process-oriented semantic knowledge mangement in product lifecycle management. *8th international conference on Digital Enterprise Technology*, 361-368.

Fischbach, M., Puschmann, T., & Alt, R. (2013). Service Lifecycle Management. *Business & Information Systems Engineering*, 45-49.

Haughey, D. (n.d.). *RACI matrix*. Retrieved from Project Smart: https://www.projectsmart.co.uk/raci-matrix.php

Kaariainen, J., & Valimaki, A. (2008). *Enterprise interoperability III: new chalanges and industrial approaches.* Berlin: Springer.

Kohlborn, T., Korthaus, A., & Rosemann, M. (2009). Business and Software Service Lifecycle Management. *13th International Conference on Enterprise Computing.* Auckland: QUT Digital Repository.

Lankhorst, M. (2009). *Enterprise architecture at work; Modelling, communication and analysis.* New York: Springer.

Marks, E. A., & Bell, M. (2006). *Service-oriented architecture. A planning and implementation guide for business and technology.* Hoboken: John Wiley & Sons.

Marrone, M; Kolbe, LM. (2011). Impact of IT service managment framewroks on the IT organization. An empirical study on benefits, challenges and processes. *Springer*, 5-18.

Mueller, B., Viering, G., Legner, C., & Riempp, G. (2010). Understanding the sources of the economic potential of service-oriented architecture: finding from the automotive and banking industry. *Journal of Management Information Systems*, 147-182.

Oberhauser, R., & Schmidt, R. (2007). Towards a holistic intergration of software lifecycle processing using the semantic web. *2nd international conference on software and data technologies*, (pp. 137-144). Barcelona.

OMG. (2011, Januari 3). *Documents Associated with Business Process Model and Notation (BPMN) Version 2.0.* Retrieved from OMG-we set the standard: http://www.omg.org/spec/BPMN/2.0/

Orlov, L. (2005). Make it matter for business innovation. *Forrester*.

Polli, R. (1968). A test of the classical product life cycle by means of actual sales histories. . *PhD dissertation*.

Puschmann, T., & Alt, R. (2011). Überbetriebliches Service Lifecycle Management. *HMD-Praxis der Wirtschaftsinformatik*, 58-67.

Rajlich, V. T., & Bennett, K. H. (2000). A staged model for the software lifecycle. *IEEE computer*, 66-71.

Rink, D. R., & Swan, J. E. (1979). Product life cycle research: a literature review. *Journal of Business Research*, 219-242.

Rossberg, J. (2008). *Pro visual studio team application lifecycle management.* CA: Apress.

Savinirs. (2012). *Savinirs Driven by Vision and Innovation*. Retrieved from PLM service: http://www.savinirs.com/plm.php

Schwaber, C. (2006, August 18). The changing Face Of Application Life-Cycle Management. *Forrester*.

Shields, M. D., & Young, S. M. (1991). Managing product life cycle costs: an organization model. *Journal of cost management*, 39-42.

SNS. (2016). *Waar staan we voor?* Retrieved from SNS: http://www.snsbank.nl/particulier/over-sns/missie-sns.html

Soto Acosto, P., Placer Maruri, E., & Perez Gonzales, D. (2010). A case analysis of a product lifecycle information management framework for SMEs. *International Journal of Information Management*, 240-244.

Soto Acosto, P., Placer Maruri, E., & Perez Gonzalez, D. (2016). A case analysis of a product lifecycle information management framework for SMEs. *Interntional Journal of Information Management*, 240-244.

Spark, J. (2015). *Product lifecycle management.* Switzerland: Springer.

Sudarsan, R., Fenves, S. J., Sriram, R. D., & Wang, F. (2005). A product information modeling framework for product lifecycle management. *Computer-Aided Design*, 1399-1411.

Van Aken, J. E., Berends, H., & van der Bij, H. (2010). *Problem Solving in Organizations.* Cambridge: Cambridge University Press.

van den Boogaart, N. (2016). *Application lifecycle management.* Eindhoven: Eindhoven University of Technology.

van Lamsweerde, A. (2001). *Goal-Oriented Requirements Engineering: A guided tour.* Belguim: Louvain-la-Neuve.

van Lamsweerde, A. (2003). From System Goals to Software Architecture. In M. Bernardo, & P. Inverardi, *Formal methods for Software Architectures.* Louvain-la-Neuve: Springer- Verlag.

Van Strien, P. J. (1997). Towards a methodology of psychological practice. *Theory and Psychology*, 683-700.

Vezzetti, E., Violante, M. G., & Marcolin, F. (2014). A benchmarking framework for product lifecycle management maturity models. *The International Journal of Advanced Manufacturing Technology*, 899-918.

West, M. (2011). *Developing High Quality Data Models.* Tokyo: Morgan Kaufmann.

Xu, X., Chen, J. L.-Q., & Xie, S. Q. (2006). Framework of a product lifecycel costing system. *Journal of computing and infromation science in engineering*, 69-77.

Yang, X., Moore, P. R., Wong, C.-B., Pu, J.-S., & Chong, S. K. (2007). Product lifecycle information acquisition and management for consumer products. *Industrial Management and Data Systems*, 936-963.

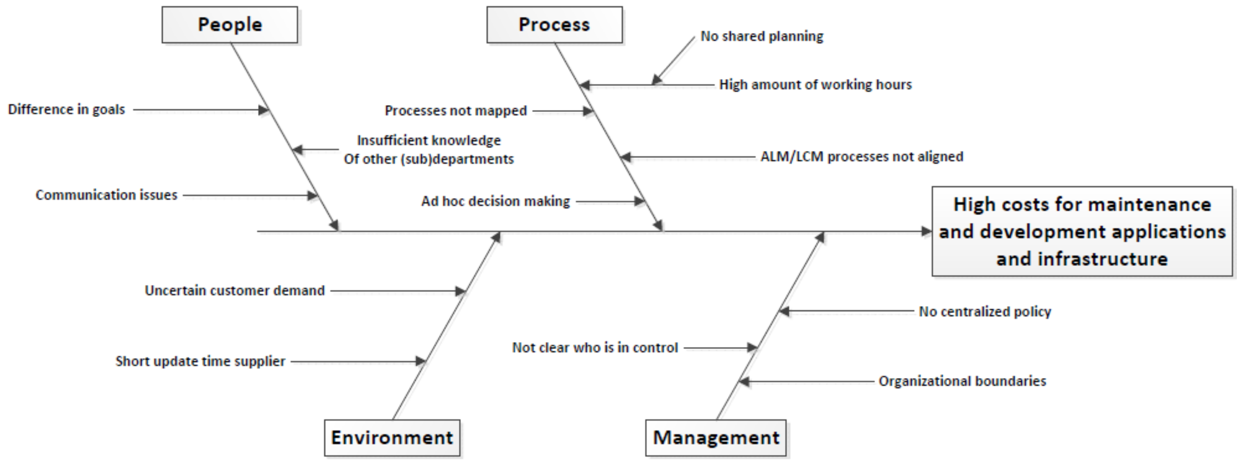
Appendix A – The taskforce

People included in the taskforce and how they are related to the project. NOG UPDATEN

Department	Employee role	ALM	LCM
Application Services	Technical Application Management Windows Applications (TAB WA)	x	
Application Services	Technical Application Management Windows Applications (TAB WA)	x	
Application Services	Technical Application Management Document Management (TAB DM)	x	
Application Services	Data Services – Database Architecture (DS-DBA)		x
Application Services	Data Services – Data warehouse (DS-DWH)		x
Application Services	Technical Application Management Back Office (TAB BO)	x	
Application Services	Technical Application Management Back Office (TAB BO)		
Application Services	Infra Engineer		
Application Services	Technical Application Management Front Office (TAB FO)	x	
User services	Work Place Services (WPS)		x
Information security and continuity management	DA IBCM		
Renewal and support	DA V&S		
Architecture	Architecture		
Architecture	Architecture		
Infrastructure services	Server & Storage infrastructure		x
Application services	System Information Management (SIM)		x

Appendix B - Cause-and-effect diagram

For every cause mentioned there is explained what it means and why/how it influences the high costs for maintenance and development of the applications and the infrastructure.



People

- *Communication issues:* Problems can arise when miscommunication occurs between the different (sub)-departments and/or employees. This happens for various reasons, and even a small misinterpretation of information could have a big impact on the costs related to the maintenance and development. If a small misinterpretations means that the processes takes longer this means it will be more expensive. For every hour more the project costs €100 more. Based on experience, 200 up to 400 hours per year can be saved, because now it takes a lot of time to tune the right ALM/LCM processes for everybody.
- *Difference in goals:* People from different sub-departments have different goals for doing their job. There are multiple perspectives that can be taken when comparing goals. This can be done for different sub-departments, as well as between different layers of the stack model. This can lead to inefficiency within processes that go through different sub-departments. With inefficiency often come higher costs. These higher costs are mostly related to more meetings and contact moments. It is also the case that I&O needs to take extra measurements because of more unsupported versions. Whereas I&O does not plan things during the sprints which means I&C needs to do more work.
- *Insufficient knowledge of other (sub)-departments:* A lack of knowledge about what tasks the other parties involved in the processes have, leads to insufficient situations. People think that something is done by another employee, which does not have to be the case. When steps are skipped, done multiple times or only partial this can lead to higher costs for the whole process. How many hours can be saved is hard to say, but a lot of frustration can be lowered when this point is tackled.

Process

- *No shared planning → high amount of working hours:* The different departments within ITC do not share their planning until it is finished. This leads to double updating of applications and/or the stack within a short time frame. And double work, means more hours need to be made for the same task, that could have been done at once. More hours means higher costs. For this point you can think about databases, operating systems, hardware and storage. There are multiple actions that need to be done repeatedly (test performance, test flexibility, measure downtime for application and systems) that take about 80 hours per platform for LCM. This means 4*80=320 hours of work can be saved. On top of that downtime can be reduced by 4-8 hours per application per combination because the application needs to be down less time. This downtime is very important, because now some applications need to be handled/updated multiple times a year. This means multiple times downtime. The total reduction of time is around 0,5 FTE.
- *ALM/LCM processes are not aligned:* The stack and the application layer are not connected. At SNS LCM processes are done on the stack and ALM processes are done on the applications. However the applications are dependent on the stack to be able to work. The non-optimal alignment of the two means that the maintenance and development processes can be optimal for one party, but disruptive for another. This inefficient way of combining the two means higher costs. There are about 50 applications that need to be handled per year and the average ALM handling costs 2 hours. This means that an average of 100 hours can be saved plus the downtime reduction that is comparable to the downtime reduction stated at *no shared planning.* So again a reduction of time of around 0,5 FTE.
- *Ad hoc decision making:* A lot of the processes are 'new' every time, because of the differences per update and what is included within each update. This means that ad hoc decisions need to be made about the process and the continuation of the process. Other sub-departments and employees therefore have less time to adapt their work to the process. This leads to an overfull planning and high work pressure for the employees that need to work on the involved processes. This can have high cost as an effect because of extra time that is needed, mistakes that can be made because of the pressure, etc.
- *Processes not mapped:* There is no general overview available of the processes related to maintaining and developing the infrastructure and applications. Therefore every sub-department made its own overview of what they have to do. Because nobody really knows the whole process, the different ways of working can interfere with having an overall smooth process. The possibility arises that different sub-departments give different input for the same sequel process steps. The cost reductions that can be made on this cause are a combination of the hour reduction of no shared planning and ALM/LCM processes not mapped. For every application or platform the combination of reduction possibilities will be different because of the different specifications of the system.

Environment

- *Uncertainty demand customer:* SNS has different users for the same application, which means that they have different requirements for what they want from the application. These different requirements make it hard to keep everybody happy. The fact that a user wants a certain feature now does not mean that this cannot change in the future.
- *Short update time supplier:* This means that a supplier of an application or part of the infrastructure gives a notice for update short before the update needs to be done. When it is unclear if and when an update is coming this makes it hard to find the time for working on it. Other projects now have to be delayed to be able to finish this update. This leads to higher costs, not only on this project, but on other projects as well.

Management

- *Not clear who is in control:* Because processes take effort from different sub-departments this means that different budgets are involved. I&O and I&C have different budgets, and because of this divide it happens that there is no clarity about who has to pay for which step of the process or for which hours.
- *No centralized policy*: As can be read in Chapter 2, every sub-department has its own processes and ways of working. There is no person or department responsible for keeping oversight on the whole process. The cost reductions that can be made on this cause are a combination of the hour reduction of no shared planning and ALM/LCM processes not mapped. For every application or platform the combination of reduction possibilities will be different because of the different specifications of the system. One combined LCM process will also reduce frustration between different parties involved.
- *Organizational boundaries:* Next to the previous mentioned causes, organizational barriers can also cause inadequate execution of processes which can lead to higher costs. The organizational structure can be a problem for this, as well as the unwillingness of different sub-departments to improve the aggregate performance. Management has an important role in this. The cost reductions that can be made on this cause are a combination of the hour reduction of no shared planning and ALM/LCM processes not mapped. For every application or platform the combination of reduction possibilities will be different because of the different specifications of the system. It is however important to mention that other programs currently starting at SNS, like I&O 2020, can help with this point.

Appendix C – Literature review

Introduction

The goal of this literature review is to gain information on the topic of application lifecycle management (ALM), to be able to conduct a research on this topic at SNS. This will be done by introducing the topic, which leads to the formulation of the research question. Finally, the structure of this literature review will be presented.

Goal literature review

To be able to graduate from the Eindhoven University of Technology the first part of the final thesis project conducted at SNS is a literature review. In this literature review an overview of the information available on the topic of ALM will be presented. Based on the found information a conclusion is drawn on where further research is useful. This is the starting point for a research proposal that will lead to the final thesis. Because the final thesis is conducted at SNS this will be taken into account for the scope of the articles used for this literature review. SNS is a financial company that is located in the Netherlands (SNS, 2016), that is interested in how they can better aligned their ALM processes with other processes at SNS.

The topic

Upon the 21st century the business environment has changed significantly. In today's competitive global market, companies are facing great on-going challenges. They are asked to produce a variety of innovative products to capture the customers' attention faster than ever before. On top of that they require to extend their operations beyond the traditional practices, because customers do not only consider price, but also after-sale services by the company (Xu, Chen, & Xie, 2006).

The software industry has been identified as one of the most important industries in the world (Colomo-Palacios, Fernandes, Sabbagh, & Amescua Seco, 2012). And regardless of industry and organization size, information technology (IT) is fundamental for improving productivity and development of knowledge-intensive products and services (Soto-Acosta, Martinez-Conesa, & Colomo-Palacios, 2010). When one realizes that the costs of an organization which main focus is IT can be accounted for 70% to 80% by its IT services (Orlov, 2005), this makes problems with IT services a relevant research topic.

Where IT departments previously focused on the production of software applications, this has started shifting towards more service focused operations (Marrone & Kolbe, 2011). More and more organizations are looking for more efficient and innovative technological services and solutions, therefore information technology service management (ITSM) is getting popular. ITSM refers to IT operations that are characterized by the emphasis on IT services, customers, service level agreements and IT function handling of the daily activities through processes (Marrone & Kolbe, 2011).

A typical company that wanted to launch a new product to the market went through the following process (Spark, 2015). First the market department decided which products were needed by the market, than the engineering department designed it. Afterwards the manufacturing department produced the product and after putting it on the market the after-sales department supported it. Sparks (2015) stated that this paradigm was agreed upon because the companies reasoned that specialists per department are the best equipped to carry out the activities and decisions of a certain function. However this led to problems because departments stopped working together and for instance had conflicted versions of the same data. Other problems that arose were environmental incompatibilities at department borders, waste, gaps, information silos, islands of automation, overlapping networks, ineffective fixes and product recalls. From this the need for a new paradigm arose. The new adopted paradigm was lifecycle management which can be divided into multiple directions, like products, applications and other fields (Spark, 2015).

Research question

The aim of this literature review is to collect information available on the topic of application lifecycles and how to manage those. To be able to do this in a structured way the following research question is answered in this literature review:

What is application lifecycle management and how can it be implemented?

As literature regarding application life cycle management is not very extended, this question is answered by first explaining the product lifecycle. Based on the differences and similarities of the product lifecycle and the application lifecycle the research question is answered.

Structure

The structure for this literature review is as follows. In Chapter 2 the methodology for this literature review is described, which means that the way the articles were found is explained here. After this, Chapter 3 gives more insight on the topic of lifecycle management. In this chapter product lifecycle management and application lifecycle management are explained. Chapter 4 includes a comparison of application lifecycle management and product lifecycle management. And finally a conclusion is drawn in Chapter 5. The last chapter also includes directions for future research.

Method

This chapter explains the methodology for the way this literature review is conducted. This is done using the stages of conducting a literature review according to Randolph (2009). The stages described are the problem formulation, literature collection, literature evaluation, analysis and interpretation and the public presentation (Randolph, 2009).

Problem formulation

During the problem formulation phase it is important to come up with criteria for inclusion or exclusion of certain topics and articles related to the topic. The starting point for this literature review is application lifecycles. To be able to find papers, different search terms are used in different search engines. The different search terms used can be found in Appendix C1.

By using inclusion and exclusion criteria some search words and terms lead to better articles than others. For this master thesis project, the main topic is how lifecycles can be implemented or used when talking about applications. For orientation purposes studies from different fields of work are checked, but projects or examples conducted at financial institutions will be seen as more relevant because the study is conducted at SNS. Some of the problems or recommendations given in these articles can be interesting to implement in the research design.

The decision has been made to divide this literature review into two topics, to be able to make a comparison between the different topics at the end. The first part explains lifecycle management from the point of view of product lifecycles and application lifecycles. In the second part the comparison between the two types of lifecycle is made.

Literature collection

By using the search terms, literature related to the research question is collected. In the first step the search terms were used in different search engines. In the second step the snowballing method is used to find more relevant articles.

By using different search engines, different articles popped up for comparable search terms. The used search engines are explained here.

Web of Science[1]: Web of Science is a journal database with citing's and cited relations. Three types of important kind of records can be found: science citation index, social sciences citation index and the arts and humanities citation index (Technische Universiteit Eindhoven, 2014).

Focus[2]: Focus is the search engine of the Eindhoven University of Technology (TU/e). Here, the complete library collection and items of the TU/e can be found, which include full-text articles, E-books, printed books and references to publications outside of the TU/e (Eindhoven University of Technology, 2016). Focus was only used when a link on Web of Science was not working.

---

[1] http://apps.webofknowledge.com.dianus.libr.tue.nl/WOS_GeneralSearch_input.do?product=WOS&search_mode=GeneralSearch&SID=R2xffgD478Vz5kzbPAe&preferencesSaved=

[2] http://tue.summon.serialssolutions.com/#!/

Google Scholar[3]: A database for scholarly literature and related works. Here, full articles can be found and also citations, authors and publications related to the topic are given. The information that can be found comes from academic publishers, professional societies, online repositories and universities (Google, 2016).

Google[4]: Google is a search engine that wants to give you the right information based on your search criteria. Information can be found on a lot of topics to gain a more general insight on the topic of the literature review.

The most used search engines are Web of Science and Focus, because they give insight into articles of a lot of journals, which means that no research is done on specific journals. All seemingly relevant articles are saved for further use based on their title and abstract. As a starting point my mentor handed over some relevant articles on the topics that are described in this literature review. These articles are the starting point for using the snowballing method. With backwards and forward snowballing articles can be found from citations or from sources that cited the original article.

All found articles are uploaded in Mendeley. Mendeley is a program to organize, read and annotate PDF documents (Mendeley, 2016). This made it easier to search for information in all articles at the same time.

During this phase it became clear that there is almost no information available on application lifecycle management. That is why the topic is broadened. Now application lifecycle management is explained in comparison with product lifecycle management, based on differences and similarities.

Literature evaluation

All the literature collected during the collection phase is evaluated to retrieve the useful information from all articles. The articles are read in more detail by reading the title, abstract, method and conclusion. Based on that information other parts that may be interesting are also read. Some collected literature is excluded from this literature review. This exclusion can happen based on one or more of the following criteria:

The first interpretation of the article was wrong and it does not mention the topics of application lifecycles or lifecycle management at all.

The data in the article is not relevant anymore because there are articles found that are more detailed or better supported.

The information in the article is comparable to other articles, but the quality of the journal is lower than for other articles. This can be determined by the number of citations, the journal it was published in, details on the topic and whether information was missing in other articles.

All used articles for this literature review can be found in the bibliography at the end of this literature review.

[3] https://scholar.google.nl/
[4] https://www.google.nl/?gws_rd=cr,ssl&ei=Yv3fVuuoEob4PumJrsAL

Analysis and interpretation

In the analysis and interpretation phase all literature that is left after the evaluation phase is categorized, determining which articles are useful for which part of the literature review. The most important articles are used for the basis of the review where others are used for insight or small adaptations or side notes. The importance of the articles is determined by the number of citations, publishing date, details on a certain topic, the journal it was published in and whether the information was missing in other articles.

For the interpretation part it is important that the discovered information helps in answering the research question. In this case the main research question is what application lifecycle management is and how can it be implemented. All interpretation of the articles is done based on this research question.

Presentation

Based on the articles and information collected after the analysis and interpretation phase, the information is divided per topic to create different chapters. As explained in the problem formulation this divide will create two main topics. Based on comparison of the different topics a conclusion is draw that leads to a possible gap in literature.

Lifecycle management

In this chapter the general concept of lifecycle management is explained. First the product lifecycle is explained after which product lifecycle management (PLM) is introduced. Then the application lifecycle is explained, after which application lifecycle management (ALM) is introduced. All information presented in this chapter is about manufacturing examples unless stated differently.

Lifecycle management provides a generic frame of reference for systems and methods that are necessary for managing all product related data during the product's lifecycle (Kaariainen & Valimaki, 2008). Lifecycle management can be used in various different situations, but for this research the focus is on product lifecycle management and application lifecycle management.

The product lifecycle

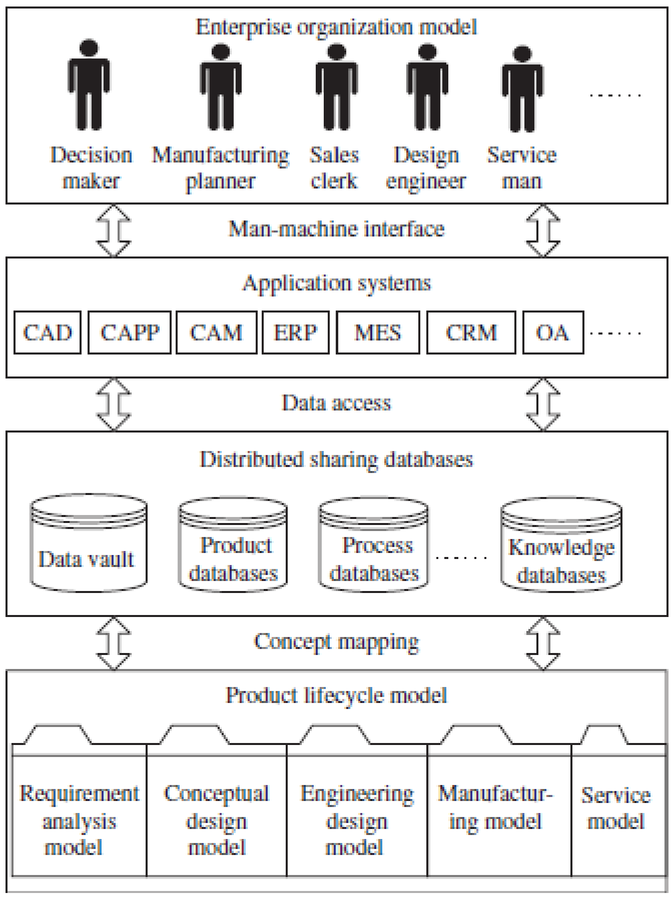
The product lifecycle (PLC) represents the unit sales curve for a product, extending from the time it is first placed on the market until it is removed (Rink & Swan, 1979). The product lifecycle portrays the evolution of product attributes and market characteristics through time, and the concept of PLC can be used in a prescriptive way in the selection of marketing actions and planning (Polli, 1968). When talking about a product this can be anything from a pen to software to a truck.

The bell-shaped PLC model (Figure 1) is adopted by the field and has a four-stage cycle-introduction that include introduction, growth, maturity and decline. However, there are more variations on the model suggested that can have up to six stages (Rink & Swan, 1979). For this literature review the bell-shaped PLC model is used, because it is the general model used in literature. The other shapes can be found in Appendix C2.

Figure 1

The phases of the classical (bell-shaped) product life cycle can be described as following: the first stage is the introduction stage, which is the startup phase. Here the company launches a product onto the market. This phase can be very expensive because of low sales versus high costs for marketing and research and development. In the growth stage the sales and profits go through a strong growth. The company begins to benefit from a larger scale of production and with the extra income more promotion can be done for the product. During the maturity stage the product is established and the aim is to maintain market share. For a company it is important to consider product updates or improvements to keep a competitive advantage. Finally, the decline stage means that the market for the product is shrinking. This can happen because customers switch to other products or the market gets saturated.

When you want to model the product lifecycle many elements have to be taken into account (Shu & Wang, 2007; Spark, 2015; Dos Santos Rocha & Fantinato, 2013). Four levels can be classified for elements that have to be taken into account, which are; organization elements, application elements, data elements and concept elements. How these levels are related to each other can be found in Figure 2.



The organization element is important, because employees of an enterprise have to generate, operate and access a large amount of product data. All system users have to be satisfied by the way they can access the information they need. The model should include a defined business structure so that every actor can perform their task according to the stated requirements. The way the different actors can access the data should be distinguished here, so that nobody has access information they are not allowed to have. The application system layer describes what data can be operated by which application system. It is also important that there is a data sharing policy that everybody has to work with. To make this sharing possible, Shu and Wang (2007) state that three levels of warehouses are needed; the personal workspace, the department warehouse and the enterprise warehouse. The department warehouse provides a space to share data for team members related to projects or phases of the product lifecycle. The enterprise warehouse where data is submitted into and can be retained for reuse. It is not permitted to make changes to the data once it is stored in the enterprise warehouse. For the model to work it is important that the mapping mechanism between each level is clarified before the model is used. In this case, concept mapping maps all physical data related to concepts of the product model. A data access mechanism decides the access strategy of enterprise employees and application systems and a man-machine interface defines how people operate the physical data.

Product lifecycle management

Product lifecycle management (PLM) can be defined as a strategic business approach for the effective management and use of corporate intellectual capital (Sudarsan, Fenves, Sriram, & Wang, 2005). In other words this means that PLM is the managing of business activities in the most effective way all across the lifecycle of the product. From the very first idea for a product all the way through until it is retired or disposed of (Spark, 2015). Furthermore, Sparks (2015) states that there are two important characteristics for PLM. The first is that when using PLM the activities that manage a company's product must be defined and documented in cross-functional business processes across the product lifecycle. Furthermore, cross-functional product data are managed by a system that manages the data across the product lifecycle.

Because of global changes, the managing of information in the lifecycle of a product is a major challenge. The benefits of using PLM for this are fast and easy exchange of documents and expertise, real-time control, improved communication and accessibility of product related information. PLM is also a collaborative platform that can improve information access and sharing inside the company, but also between a company and its stakeholders (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016). Felic et al. (2014) add more benefits like reduced time to market, a better collaboration and savings. However, they are careful, because for a lot of companies implementing PLM still means that they have to make heavy changes to the company structure. On top of that PLM solutions are based on an integrated model that stores product data that is shared with all contributors. Challenges arise when this information can only be interpreted by experts. PLM can also lead to communication overhead that leads to extra costs, extra product development time and therefore longer time to market (Felic, König-Ries, & Klein, 2014). It is important that during the product lifecycle a collaborative approach is used, because problems with using PLM can be categorized as product-centric, process-oriented or human-centric knowledge management (Felic, König-Ries, & Klein, 2014). To be able to manage all data when using PLM, different methods can be used that can be manual as well as software driven.

There is a lot of software available for implementing PLM, but the most common issue still is the lack of interconnectivity with other information systems. For small and middle sized companies another problem is that the available software is very expensive, which leads to limited access for them. (Vezzetti, Violante, & Marcolin, 2014). Savinirs (2012) created an overview of PLM including milestones, deliverables, roles and skills & knowledge, which is presented in Figure 3.

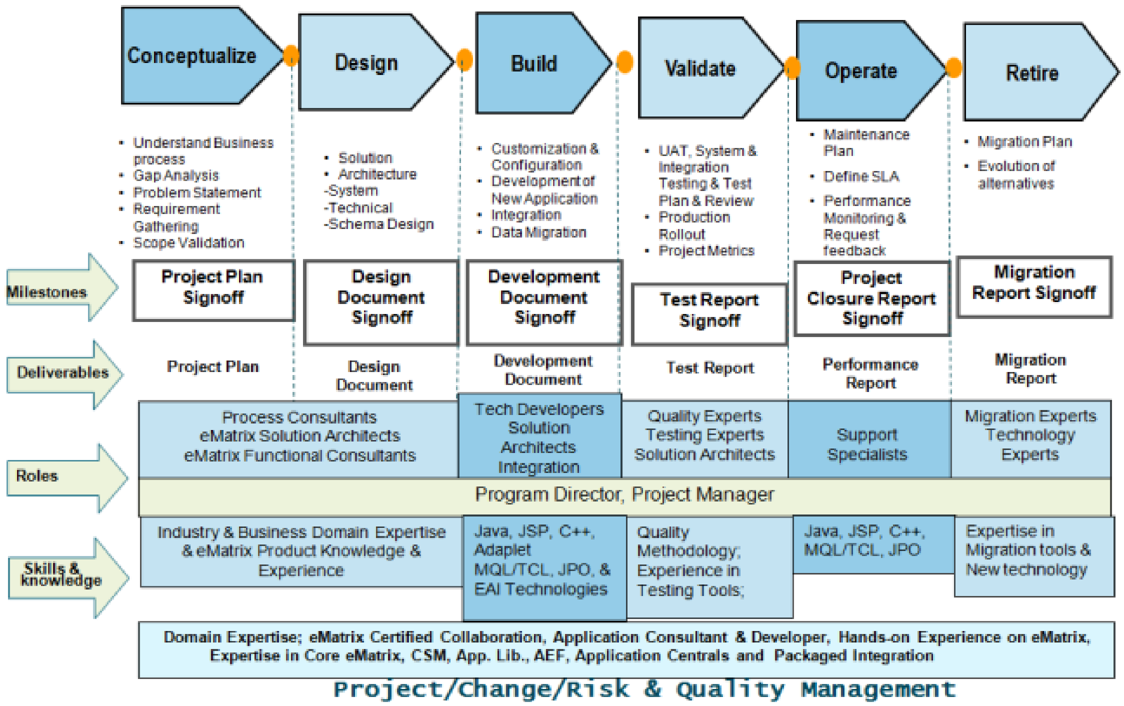


Figure 3

Information management

Product lifecycle information management (PLIM) is developed to address the need of systematically managing the acquired lifecycle data to close the information loop. The most important feature of this PLIM system is that it can receive and manage lifecycle data to provide information and knowledge for decision-making (Yang, Moore, Wong, Pu, & Chong, 2007). The PLIM system is implemented in such a way that it provides a business platform for manufacturers to interact with multiple stakeholders. It focusses both on business to business (B2B) and business to consumer (B2C). A reference architecture for PLIM has been developed by Yang et all (2007) for consumer products. It uses a three tier structure that can be found in Figure 4. The first tier is the external world, which consist of various information management actors. The second tier is the lifecycle data management layer that consists of different components. The function for this tier is to put all data in a neutral format, so that all data is represented in the same way. Another function for this tier is to make the communication and exchange of information more neutral. The third and final layer is the database management service (DBMS) tier. In the DBMS, all data is stored in different management components. It houses the product lifecycle data and information.

When choosing to implement PLIM it is important to keep some things in mind. First it is important to keep track of how a certain PLM system is chosen. This is important because failures can occur due to incompatibility between the selected software and the philosophy of the company. This can be avoided by extensively analyzing the business processes and procedures. Secondly the role of the employees is crucial for successfully implementing a PILM framework. Employees of all hierarchical levels have to be committed to the PILM framework, because this creates an atmosphere of acceptance and usage (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016). Yang et all. (2007) also state that a PILM framework can be commercial or self-developed, but that does not matter for the benefits. However, a self-developed PILM framework might be more adapted to the companies' needs and other systems, which means that the effectiveness will be higher.

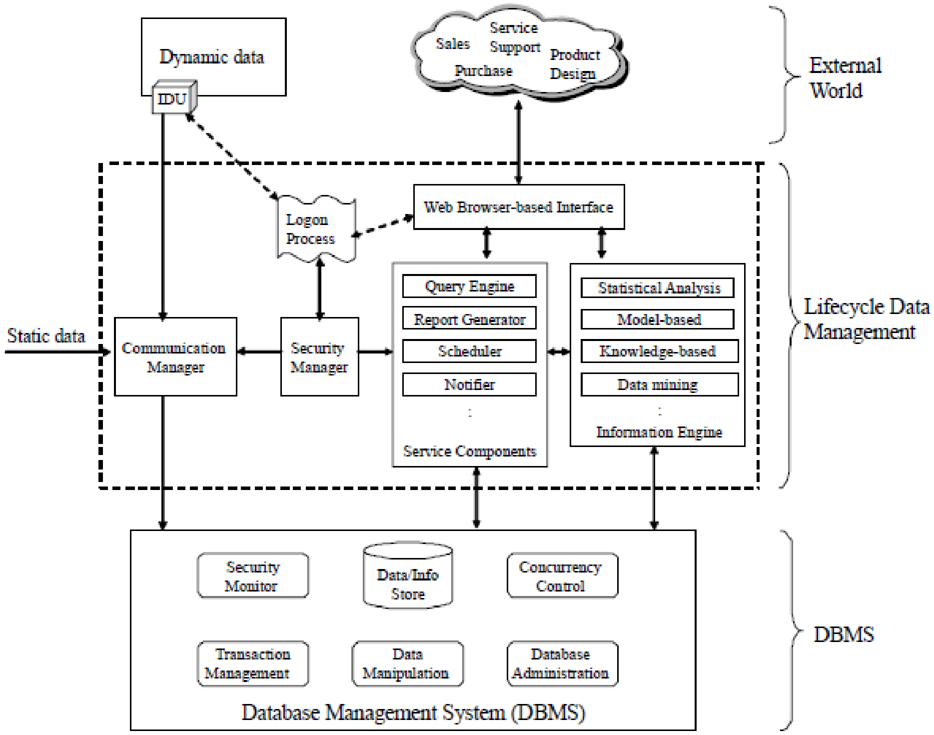
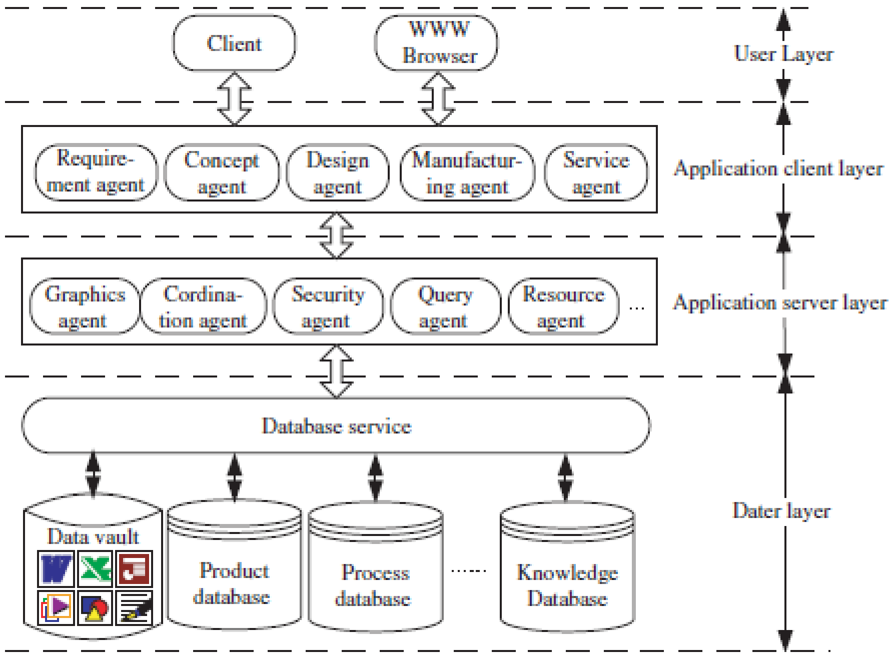


Figure 4

When talking about a web-based product the information sharing is done differently. A web-based framework is developed that consists of four layers (Shu & Wang, 2007) that are presented in Figure 5. The first layer is the data layer, that stores product-related data information in physical media as various formats in different locations. On top of this is the application server layer. This layer processes service requests from the application client layer and consists of mechanisms and software modules. The application client layer is the interface that users use to submit their service requests to the application server layer. Within this layer, five types of application agents related to PLM exist; requirement agent, concept agent, design agent, manufacturing agent and the service agent. The top layer is the user layer, which holds mechanisms and software modules for different end users that cover the entire product lifecycle.

The difference between the web-based and the non-web-based information sharing is that with a web-based option, webpages and other information can be downloaded from a browser to your personal computer. Opposite to that, with a non-web-based system the client has to download an application that can run on a personal computer.



Web-based PLIM model

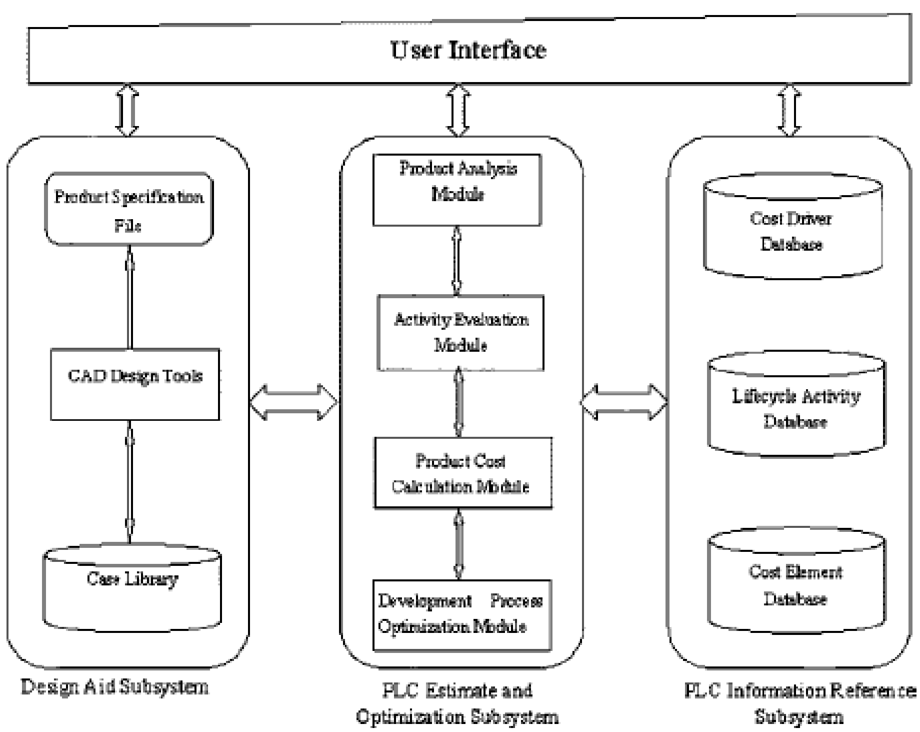
Cost model

When thinking about implementing PLM the cost for doing so have to be taken into account. The product lifecycle costs are an important measure for PLM implementation, because it can track and analyze the financial information of activities associated with each phase of a product's lifecycle (Xu, Chen, & Xie, 2006). Product lifecycle cost refers to all the costs that occur over the whole lifecycle of a single product (Artto, 1994), which means that visibilities across activities are increased, their performances improved and product lifecycle costs reduced. If properly applied, the techniques and

Product Lifecycle Stages	Cost Members
Design stage	Engineering design cost Drawing cost Computer processing cost Design modification cost Production preparation cost Management cost
Manufacturing stage	Material cost Facility cost Production cost
Marketing and after-sale stage	Marketing cost Distribution cost Maintenance cost
Disposal and recycling stage	Retrieval cost Disassembly cost Reprocessing cost Landfill cost

methods of cost analysis and cost estimates can make a significant contribution to new product development (Durairaj, Ong, Nee, & Tan, 2002). Different studies have proven that a significant portion of product lifecycle costs is affected by the decisions made in the design stage, which means that a proper cost estimate during the design stage is essential for making sensible decisions on the product. The most common costs per phase can be found in Figure 6 (Perera, Nagarur, & Tabucanon, 1999).

In general, product lifecycle costs are estimated by using one, or a combination, of the following two methods: the costs of a product are estimated in comparison to the cost of a similar product or aspect that was made in the past. Or the labor times and rates are estimated, material quantities and prices are calculated to determine the direct costs of a product. On top of this an allocation rate is used to allow for indirect costs (Shields & Young, 1991). Different types of cost estimation methods are developed over the years, but many of them have weaknesses. Xu et all. (2006) developed a framework that provides product cost information for decision making at all stages of a product lifecycle. Different types of methods are used, which include activity based costing, dynamic programming and object oriented modeling, that combined overcome the weaknesses of the existing cost estimate models. The framework they created is presented in Figure 7.

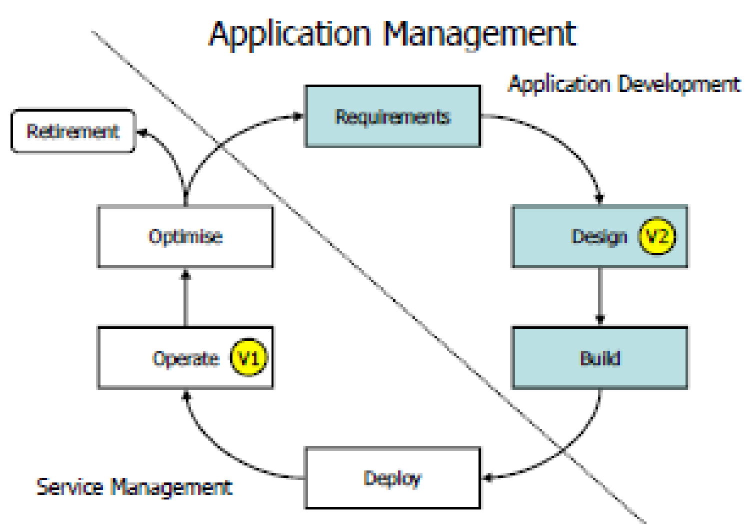


The design aid subsystem provides functions to help the product developers quickly build a model for a new product through searching a similar, existing product using case based reasoning (Rehman & Guenov, 1998). The PLC information reference subsystem consists of three reference databases that use different ways of determining different types of cost drivers. The final subsystem is the PLC estimation and optimization subsystem that is made up of four modules; the product analysis module, the activity evaluation module, the product cost calculation module and the development process optimization module. The interaction between the subsystems is really important, because it determines the system structure and it facilitates the information flow in-between the subsystems. Using the framework helps product designers to obtain lifecycle cost information in an early stage of the product development and because the framework is dynamic it can be updated when the process progresses through the product lifecycle stages (Xu, Chen, & Xie, 2006).

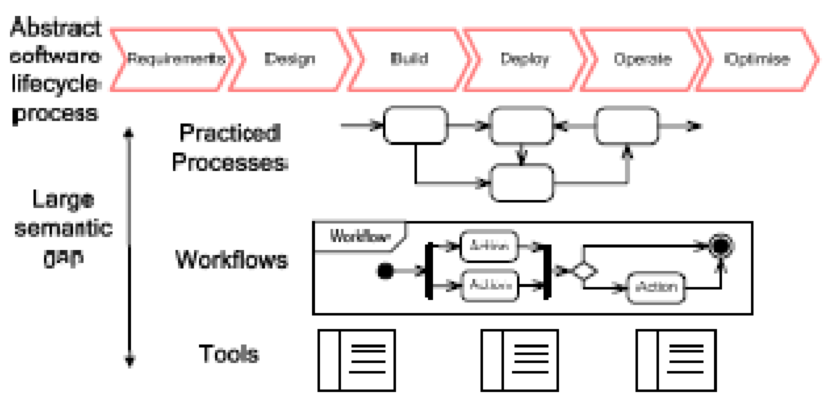
Even when using the product lifecycle stages, it does not necessarily mean that the data throughout the product lifecycle is used efficiently. Companies went from working with separate clusters to a more holistic view when implementing product lifecycle management. When implementing PLM, a lot of different factors have to be taken into account. First it is important to realize that the PLM has more phases than the product lifecycle that can be important during all phases of the product lifecycle. PLM starts from the moment the product idea is generated until the moment the product is no longer produced. PLM can be done in different ways, but the main reason for doing it is creating insight into the product flow and being able to collect and share data through the whole process.

The application lifecycle (software lifecycle)

The application lifecycle is often called the software lifecycle in literature, but it is the same type of process. For a long time, software development was seen separately from operational disciplines. But the interest for a more holistic view on the entire lifecycle is growing. The application lifecycle consists of application development as well as service management, as can be seen in Figure 8. By using the application lifecycle, a more broad view can be given than when only software development processes are taken into account. To make the lifecycle economically and efficiently, it is necessary that information flows of semantically annotated information is retrievable in a diverse operational infrastructure across organization boundaries (Oberhauser & Schmidt, 2007).



Oberhauser and Schmidt (2007) also identify two reasons why the implementation of an application lifecycle can rarely be fully exploited. The first point they mention is that there is a semantic gap between the abstract process descriptions and the executed processes (Figure 9). On top of this, best practices in other organizations cannot directly be reused for other instances. This means that executable processes have to be abstracted manually to get process descriptions that fit a specific company. However, executing these tasks manually makes this a very error-prone task. The second point is that often there are breaks in the information flow between software operations and software development. This means that the software developer gets little to no feedback about the created software. There are two types of information that could be interesting for the developer to get back. The first is the run-time data, which is data that can be observed by a machine that gathers this information automatically. This information can be important for debugging or updating the software. The second type of information is the information that cannot be gathered by a machine. This can be information about functional defects in the software. The software itself runs, but gives invalid results or exhibits incorrect behavior. This information is usually gathered by a service desk or helpdesk. In practice, it is very hard to decide which information can be used to diagnose which type of problem. An example of this could be that when there are performance problems, the server log and the application database can be analyzed, whereas with reliability problems the server can be analyzed together with the application log.



Application lifecycle management

Application lifecycle management (ALM) deals with the way a software system or application is conceived, planned, developed, maintained and decommissioned (Rajlich & Bennett, 2000). Typical activities that are included in the lifecycle are requirements development and management, project planning, solution development, deployment and issue tracking. Doyle (2007) states that ALM is a set of tools, processes and practices that enable a development organization to implement and deliver to software lifecycle approaches. This means that some kind of solution for ALM exists in every company (Doyle, 2007). The purpose of ALM is to provide integrated tools and practices that support project cooperation and communication through a project's lifecycle (Figure 10). For management it provides an objective mean to monitor project activities and generate real-time reports from project data.

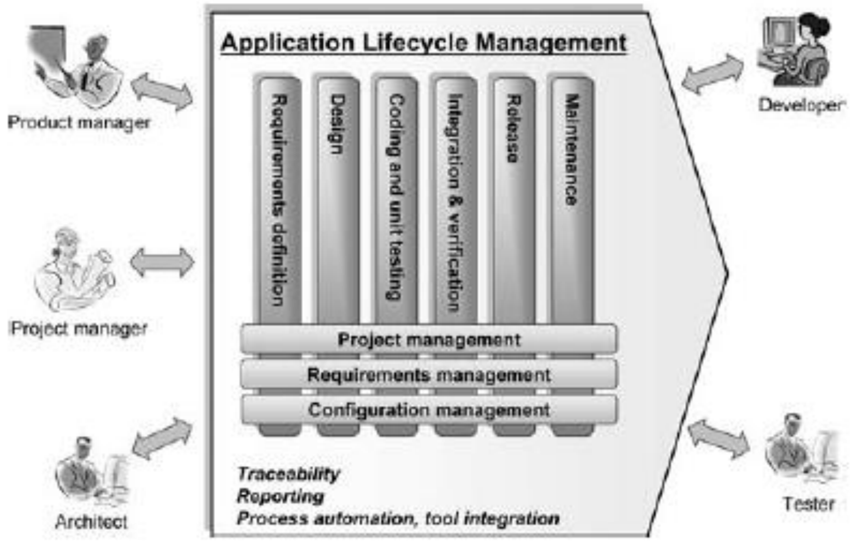
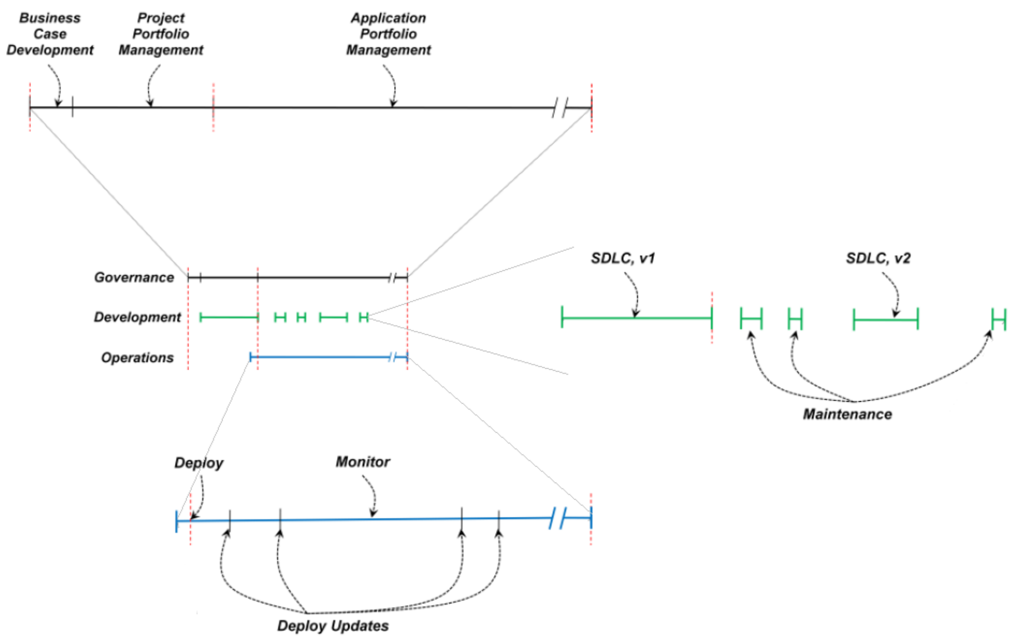


Figure 10 is created based on a literature review (Kääriäinen & Välimäki, 2008) performed on the topic of ALM. In this literature review three pillars of ALM are defined, which are traceability, process automation and reporting. *Traceability* includes all the cases that help to demonstrate that the software has delivered the functions the business wanted it to. Because of the increasing need to coordinate development across roles, location and organization traceability is more of a necessity than an ideal. *Process automation* includes the storing of all documentation associated with the tradeoff between different functions. It is very useful to have executable process descriptions that actually correspond to the processes, instead of a 'book of process' that is ignored. The final pillar is *reporting*, which includes the visibility of the progress on projects. Nowadays managers often only know the developments of their own part of the project, whereas if all information would be shared everybody would be more up to date. To achieve this, managers have to report about their own part of the project to the others working on it. Furthermore, the literature review showed that there are different viewpoints on what is the most important discipline of ALM. Doyle (2007) states that requirements management is one of the most critical disciplines, where others name traceability as most important. Configuration management is also a very important discipline, because to meet the changing needs of industry, it becomes important that a system could be merged into different types of infrastructures, so that it can support the entire software life cycle.

When using ALM it is important to understand it's true scope (Schwaber, 2006). Firstly, ALM is a discipline, as well as a product category. It is sometimes hard to remember that ALM can be implemented using a tool, but also without supporting tools. All three pillars of ALM can correspond to a manual process that could be made more efficient and effective through tool integration. Secondly ALM does not support specific life-cycle activities; it rather keeps them all in synchronization. ALM ensures that activities are coordinated, which keeps practitioners efforts directed at delivering applications that meet business needs. Finally, an ALM solution is the integration of life-cycle tools, not merely a collection thereof. A business already has tools implemented for different types of use. The connections between these tools, rather than the tools themselves, are the ALM solution.

The view on ALM given here can be used, but there is also another view that can be used as a reference. Chappell (2008) states that for ALM to be both accurate and useful, the view on it should be a broad one.

He defines three distinct areas when talking about ALM. The defined areas are governance, development and operations (Figure 11). The purpose of *governance* is to make sure that the application provides what the business needs, and consists of business case development, project portfolio management and application portfolio management. Governance is the only area that extends through the whole ALM timespan, and therefor is the most important one. If a company does not get this right, it will never come close to maximizing the applications business value. The second area is *development*, which starts after approving the business case. It involves the initial version of the application and the updates and maintenance that keeps the application up to date. The final area is *operations*, which includes monitoring and managing the application. It can entail multiple iterations and is closely related to de development line, because from the moment the application is deployed it has to be monitored throughout its lifetime (Chappell, 2008).



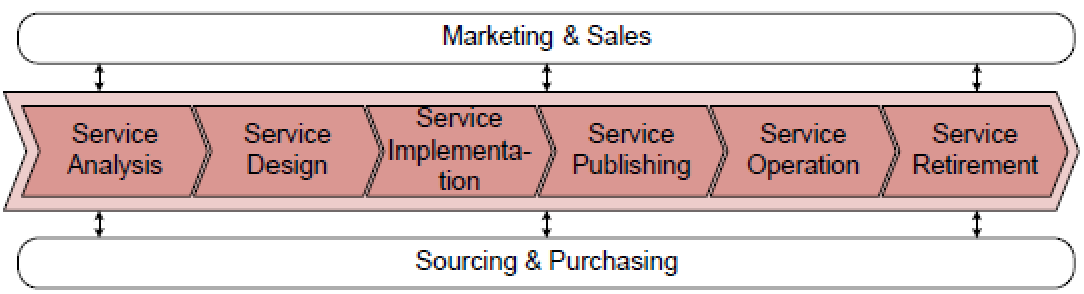
ALM platforms can be used to reduce the complexity of integrating development and management tools, but first the ALM activities for the project have to be correctly identified. To be able to identify these activities correctly, three aspects have to be taken into account (Jwo, Hsu, & Cheng, 2013). The first one is defining ALM activities, which includes the formal or semi-formal definition of the processes. This can be very challenging, because formulating these activities can be a project itself (Bennett & Rajlich, 2000). The second factor is implementing the definitions on an ALM platform. Here it is important that a platform or way of working is adopted that fits the business. This can be done by using a standard ALM platform or creating a tailor made solution based on whishes from the field. It is really important that this decision is made based on research, because a platform can offer too many unwanted features and can miss organization critical features (Shaw, 1990). The final point that has to be taken into account is enforcing ALM discipline. Even when the previous problems are dealt with, whether ALM activities are faithfully executed remains an area of doubt to many organizations (Rossberg, 2008). Shaw (1990) states that the most common problems are related to the coordination and cooperation among the developers, that view these actions as non-technical overhead. Everybody that works with PLM needs the right information at the right time in the right context.

A lot of the information about ALM is based on case studies or derivations of other projects. There is only little information available, and within the available information there are different methods that can be used for implementing ALM. However, the methods agree upon the point that an ALM solution has to be adapted to fit the needs a specific company. No ALM solution is standard, because it is a way of connecting instances within the business by using a platform. A common outcome of using ALM is that it reduces complexity within the company when it comes to application lifecycles. A final note that is agreed upon is that an important fact is that ALM requires coordination and cooperation among the people that have to use it and work with it.

All information given until now was mainly on the first phases of the lifecycle, however the service management part of the lifecycle is just as important. A service-oriented business level enables an organization to expose and offer operations as business services to business partners in order to facilitate on-demand collaborations (Kohlborn, Korthaus, & Rosemann, 2009). The service lifecycle as well as service lifecycle management will be introduced, which can be used during the operate and retire phases of the product lifecycle as well as the application lifecycle. The basis of the service lifecycle are Service oriented architectures (SOA). A SOA is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains (Kohlborn, Korthaus, & Rosemann, 2009). This is getting more and more important when talking about business models (Mueller, Viering, Legner, & Riempp, 2010), because when organizations also want to offer services they have to combine the different SOAs (e.g. SAP, Oracle) that are used within the company.

The Service Lifecycle

Kohlborn et al. (2009) conducted a literature review to create a generalized service lifecycle to represent the holistic view of SOA that includes business as well as software services (Figure 12).

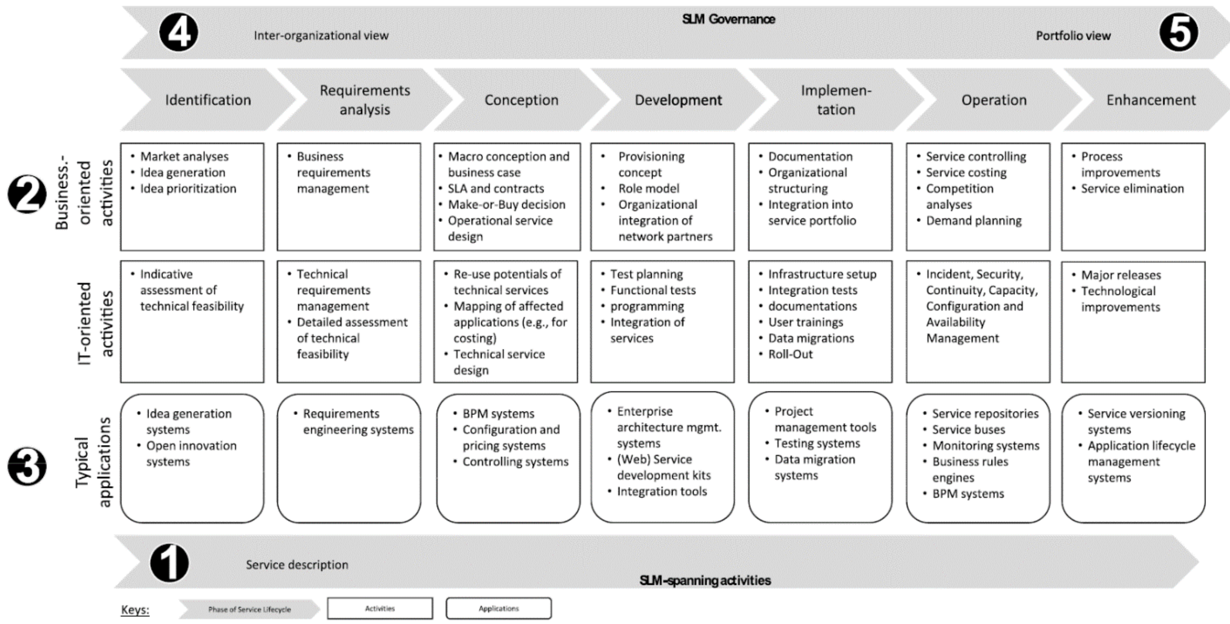


Before this lifecycle starts, some preparations need to be done, which are done at a more strategic level. It is important that motivations for SOA and services get documented and business and IT imperatives that need to be resolved need to be mapped (Marks & Bell, 2006). During the *service analysis* phase all activities required for identifying and contextualizing a service are captured. This can be done from different starting points and leads to the scope of the project. Next the *service design* phase starts. During this phase the conceptual service design is translated into a more detailed model of the service that can act as an appropriate specification for the actual development and reuse of the service. The goal of this phase is to get to an idea that is refined enough that the service can be implemented afterwards. Based on the created service design the *service implementation* phase follows. During this phase the actual service is built, which can be a piece of software with technical service characteristics or a marketable and fully executable non-technical service. It is important that a decision is made about the hosting environment of the application and the programming language. The service needs to be tested, to verify that requirements have been met and deliverables are of the accepted quality and according to standards (Papazoglou, 2008). Once the service is build and tested it can be published during the *service publishing* phase. During this phase access rights, costs, pricing models and sanctions in case service level agreements are not met are determined. When all this is done the service can be published and the *service operations* phase can start. The service is up and running and related metrics are saved for monitoring, billing and compliance purposes (Papazoglou, 2008). The service provider has to regularly update customers about maintenance activities and updates that will lead to new capabilities, prices and/or contractual attributes. At one point in time the service might not be good enough anymore or updating and maintenance gets to expensive that the *service retirement* phase starts. The service has reached the end of its economic or technical competitiveness and gets taken out of service.

Service lifecycle management

Software providers no longer offer their solutions solely as complete packages, but rather allow customers to use them in parts or as a whole on a pay-per-use basis (Fischbach, Puschmann, & Alt, 2013). This leads to higher complexity, due to heterogeneous service specifications, service development processes, service implementation and operating models (Puschmann & Alt, 2011). Different suppliers have heterogeneous platforms, which means dedicated management of services along the lifecycle (service lifecycle management - SLM) is needed. Two types of approaches can be used when implementing SLM (Fischbach, Puschmann, & Alt, 2013), the IT-oriented approach or the business-oriented approach, both will be explained. The *IT-oriented approach* or 'SOA Management' can be described as the management and monitoring of applications, services, processes, middleware, infrastructure and software in accordance with the business goals (Behara & Inaganti, 2007). The best known representatives are the IT Infrastructure Library (ITIL) and the Control Objectives for Information Related Technology (COBIT). ITIL is a collection of established common practices describing a possible implementation of service management. Where ITIL focusses on the management of IT services, COBIT aims at connecting IT specific (e.g. ITIL) to companywide frameworks (Fischbach, Puschmann, & Alt, 2013). The *business-oriented approach* aims at transferring approaches from industrial product development and product management to the service area, mostly by means of process-based models, because services often have business-oriented aspects that go beyond technical elements. Most of the business-oriented approaches can be attributed to certain scientific disciplines like marketing, product management, finance and engineering (Fischbach, Puschmann, & Alt, 2013).

Both of the above mentioned approaches are integrated in the service lifecycle (Bardhan, Demirkan, Kannan, Kauffman, & Sougstad, 2010). Based on this combination the integrated SLM is created that can be found in Figure 13. The service lifecycle consists of seven phases; identification, requirements analysis, conception, development, implementation, operation and enhancement. The service lifecycle enables a process-oriented, integrated view on the two approaches, where five requirements must be considered (Fischbach, Puschmann, & Alt, 2013).



Requirement 1 are the *service descriptions*. The description of a service constitutes the basis of many management related activities along the service lifecycle. Number two represents the value orientation, which shows the bridge between a value oriented corporate management and a process-oriented corporate organization. This point comes from the unavailability of approaches that address cost- as well as revenue-aspects for the management of services (Fischbach, Puschmann, & Alt, 2013), which means that determining the costs of a piece of hardware is done, but not the service-related costs of installing and maintaining the hardware. Number three indicates the *system support*. Because of the complexity of SLM a consistent IT support is needed to be able to formalize the processes. No single solution can cover the whole lifecycle, which leads to the prevailing of clusters that focus on certain aspects of the service lifecycle. The fourth number shows the *inter-organizational view*. To be able to implement an inter-organizational orientation of SLM, cooperating partners have to agree on basic rules. This means an inter-organizational governance has to be created as well as role models and inter-organizational interfaces. Number five indicates the *portfolio view*. The description and design of a single service is the basis for SLM, however in the inter-organizational context the use of SOAs can lead to high complexity.

Implementing an integrated SLM solution can be a challenge, because implementing SLM requies extensive standardization with respect to the governance, processes, applications and service descriptions. If a company is able to do this, it can reduce costs for the organization and lead to time benefits. Finally, by creating standardized unified service level agreements the quality of service definition, provision and enhancement can be increased (Fischbach, Puschmann, & Alt, 2013). However, Fischbach et al. (2013) state that there has not been done a lot of research in these areas, so future research is important on this matter.

PLM versus ALM

The previous chapter explains product lifecycle management and application lifecycle management. However, based on a lack of information on ALM, a comparison between ALM and PLM can lead to further knowledge on ALM. Table 1 shows all benefits and pitfalls of ALM and PLM, of which the most important ones will be used for making the comparison.

PLM is the adopted paradigm when it comes to the production of products (Spark, 2015). When talking about a product this can be anything from a pen to software to a truck. From this point of view, applications are also part of PLM. However, based on new insights on the lifecycle of applications, ALM has been introduced. The basis of ALM still is PLM, but with adaptations to the application lifecycle.

PLM is defined as managing the business activities in the most effective way all across the lifecycle of the product (Spark, 2015), where ALM is defined as providing integrated tools and practices that support project cooperation and communication through a projects lifecycle (Doyle, 2007). Based on these statements the conclusion can be drawn that there is a lot of overlap between the two definitions. Both PLM and ALM take into account the whole lifecycle and tools which can be used for support in implementing either of the two.

When looking at the benefits of ALM there are only a few that can be found in scientific literature. Because some form of ALM can be implemented in every company (Doyle, 2007) it is a very flexible concept that can have a lot of benefits, but also has a lot of pitfalls. A benefit both ALM as well as PLM have is that it gives real-time insight into project data and therefore provides a company with the possibility for real time control (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016). The point that ALM has to be implemented over organizations' boundaries makes it a challenge to implement it (Oberhauser & Schmidt, 2007). This has also been a problem when PLM got implemented, but because all parties involved understood that it would help the business it became accepted (Spark, 2015). Based on this, the assumption could be made that there is a chance that ALM will also be accepted, but that it needs more time and more proof needs to be generated to show that ALM really works in different types of situations.

A very important benefit of PLM is the improved communication (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016). If the communication would improve when ALM is introduced, this could mean that the breaks in the information flow between operations and development could be closed. Better communication could tackle more problems that are now foreseen when implementing ALM. The gap between the abstract process description and the executed processes (Oberhauser & Schmidt, 2007) could be closed when departments and teams are working together and have insight in the processes of others in the ALM process.

Some other benefits of PLM mentioned are reduced time to market and savings (Felic, König-Ries, & Klein, 2014). To say something about reduced time to market when implementing ALM is hard, because the lifecycle of a product is significantly different from the lifecycle of an application. It could be possible that when ALM is implemented in a good way for a specific company this leads to savings for the company. However, this is company specific and depends on a lot of different factors that all have to be taken into account.

When looking at the information management of the data in the product lifecycle the PLIM could also be implemented for applications. The reference architecture given in Figure 4 is a general one that includes the external world, the system that does the data management and the databases. These factors are also important for the ALM processes, only the data that has to be saved might look different. The other factor that is clear when talking about PLM is where different costs within the product lifecycle can be related back to. This would be a valuable step to take for ALM and the application lifecycle, because when everybody knows where costs are made it is easier to see where costs can be reduced. Because the service lifecycle, as well as service lifecycle management, can be implemented in both ALM and PLM it is not relevant to include this in the comparison. Making predictions about how and if PLM best practices can be used to say something about ALM processes is difficult, because all ALM implementations are different and specific based on the company where it is implemented.

	PLM	ALM
Benefits	Takes into account the whole lifecycle (Spark, 2015)	Takes into account the whole lifecycle (Doyle, 2007)
	Tools can be used for support in implementation (Xu, Chen, & Xie, 2006)	Tools can be used for support in implementation (Schwaber, 2006)
	Real time insight into project data and control (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016)	Real time insight into project data and control (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016)
	Improved communication (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016)	Very flexible concept that can be implemented in every company (Doyle, 2007)
	Reduced time to market (Felic, König-Ries, & Klein, 2014)	It is a product category as well as a discipline (Schwaber, 2006)
	Savings (Felic, König-Ries, & Klein, 2014)	
	Overview of where different costs are made within the product lifecycle (Savinirs, 2012)	
	Can improve information access and sharing inside the company as well as between the company and its stakeholders (Soto Acosto, Placer Maruri, & Perez Gonzalez, 2016)	
	Self-developed PILM frameworks work the best, but take more time to develop (Yang, Moore, Wong, Pu, & Chong, 2007)	
Pitfalls	Heavy changes need to be made to the companies structure when implementing PLM (Felic, König-Ries, & Klein, 2014)	Has to be implemented over organizations boundaries (Oberhauser & Schmidt, 2007)

	Product information is shared with all contributors of the product which is a problem if information can only be interpreted by experts (Felic, König-Ries, & Klein, 2014)	There is a gap between the abstract process description and the executed processes (Oberhauser & Schmidt, 2007)
	Communication overhead (Felic, König-Ries, & Klein, 2014)	No generalizable model or framework available for implementation or reuse (Oberhauser & Schmidt, 2007)
	Lack of interconnectivity with other information systems (Vezzetti, Violante, & Marcolin, 2014)	There can be breaks in the information flow between software operations and software development (Oberhauser & Schmidt, 2007)
	Available software is very expensive (Vezzetti, Violante, & Marcolin, 2014)	

Conclusion and further research

First a short conclusion is given based on the literature described in this review which leads to the description of a gap that was found in the literature. This gap can be researched in future research.

Conclusion

The research question of this literature review, stated in Section 1.3 was given as: *What is application lifecycle management and how can it be implemented?*

Based on this question, literature was searched and with this information the review was written. The first part of the question is about what application lifecycle management is. This can be answered by using the definition of Doyle (2007) who states that 'ALM is a set of tools, processes and practices that enable a development organization to implement and deliver to software lifecycle approaches'.

The second part of the research question was harder to answer. The answer to the question how application lifecycle management can be implemented is still not clear. Because of the fact that ALM has no standard format for every company to implement, it is not possible to make a general claim on how to implement ALM. As can be read in this review a lot of different factors have to be taken into account when implementing ALM, but how a specific company can implement ALM cannot be answered with a general statement.

Future research

Based on this conclusion and the rest of the report, some directions for further research are discovered. The first is an investigation of how ALM can be implemented in different companies. This research can be done using companies with the same characteristics or different ones, so that you can compare or make a more generalized framework for that specific company type.

Another approach is to see whether ALM can be implemented in combination with other processes a company already uses. This could be lifecycle management, lean, six sigma, etcetera. It would then be really important to see whether ALM can merge all systems using another way of working as a basis. This is where the services also become relevant, because to be able to deliver a good service this must be able to be integrated in the processes already running. How services are related to ALM is therefore an interesting field for future research.

Furthermore, one could also state that any research that leads to a literary document is useful at this time, because of the lack of grounded research available until now. If more research becomes available, better comparisons can be made between situations and therefore better insight can be given into the concept of ALM.

Appendix C1. Search terms

A short overview of the search method used for this literature review is given in this appendix. In all databases (Web of Science, Focus and Scholar) the articles are searched based on relevance. At Web of Science this had to be selected, for the other two this is the standard. No time period has been selected for the initial research. Table 2 gives the main search terms with their synonyms. Below the research method for the search terms will be further explained.

	Search term	**Synonyms**
1.	Lifecycle management	Life cycle, life cycle management, lifecycle, cycle management
2.	Product lifecycle management	Product lifecycle, product life cycle, product development cycle, product development, product management, product management cycle, PLM, product stages, stages PLM, PLM cost
3.	Application lifecycle management	Application lifecycle, application life cycle, application management, software lifecycle, ALM, software lifecycle management, application cycle, software development cycle
4.	Literature review	Wright literature, dissertation, review articles

Phase 1:
During this phase general information is searched about conducting a literature review. This is done because of the lack of knowledge and experience on the topic.

	Search term	**Web of Science**	**Focus.tue.nl**	**Scholar.google.com**
1.	Literature review	311,822	4,095,318	3,000,000
2.	Dissertation	6,288	766,468	3,800,000

Based on this search and some conversations with other students the article by Randolph (2009) was selected as a base for this literature review. Based on the structure Randolph (2009) explains in the article, articles on the topic of ALM are searched and selected.

Phase 2:

	Search term	Web of Science	Focus.tue.nl	Scholar.google.com
1.	Lifecycle management	3,302	280,452	239,000
2.	Cycle management	41,770	2,664,531	4,900,000
3.	Product lifecycle	3,030	266,378	167,000
4.	Product lifecycle management	1,589	217,391	137,000
5.	Application lifecycle	2,013	193,558	200,000
6.	Application lifecycle management	871	161,652	168,000
7.	Lifecycle management framework	1	64,656	323,000
8.	Application lifecycle management framework	1	51,771	253,000
9.	Product lifecycle management framework	1	51,521	235,000

During Phase 2 the articles were searched for this literature review. When the search terms stated in Table 4 where used a lot of hits appeared on the different search engines. The basis used was always Web of Science. When there were little to no articles available that seemed interesting for this review Focus was used. Scholar was only used when an article seemed interesting, but the link did not work in any of the other two search engines. Now, most of the times the article was available to read and therefore taken into account.

Still there are a lot of hits stated in Table 4, way too many to all be taken into account. However a lot of these hits turned out to not be on the searched topic. Based on the sometimes general terms like ‘product’ a lot of articles popped up on the topic of product selling or articles on a specific product. The term ‘application’ also showed a lot of articles on how different methods could be applied in different situations. That is why in all of the cases almost 2/3 of the articles where on a different topic than relevant for this literature review.

Based on the abstracts of the different articles, some were selected. An article is selected when from the abstract it seemed that the article was on the topic of ALM or PLM. Based on the criteria stated in Chapter 2.3 articles were left out or included. Than the remaining articles were read in greater detail and the most important once were selected. These are the shown in orange in Table 6.

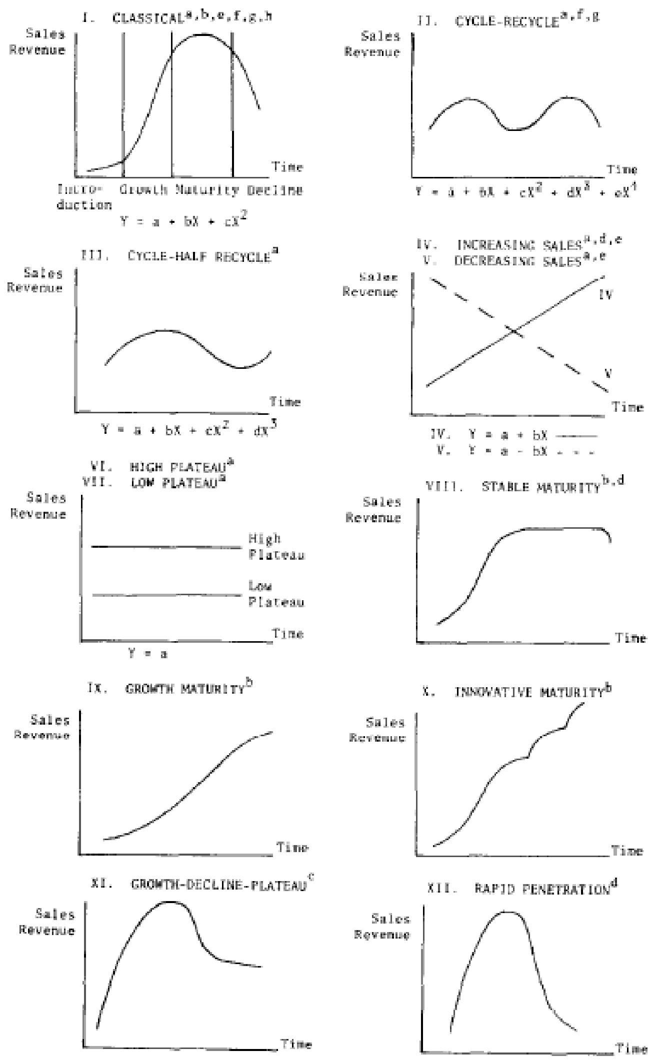
When a good article was found snowballing was used to search for related articles on the same or related topics. The articles found through snowballing were evaluated based on their title and abstract, when this was interesting the rest of the article was evaluated.

Phase 3:

After the first version of this literature review was handed in, the feedback mentioned that information on the topic of maintenance and service was missing. Based on this advice extra articles were searched

using the same method described in Chapter 2. Two articles were handed to me by my supervisor for a good starting point, these are shown in orange in Table 5.
Search terms used are service lifecycle, service lifecycle management, service management, service life cycle management. But mainly the snowballing method is used based upon the received articles.

Appendix C2. Different PLC graphs



Appendix D – Interviews

In this appendix the different interviews conducted are explained and the asked questions are presented. As explained, semi-structured interviews were used to be able to adapt the interview based on the comments the interviewee makes. The structure is the same for all conducted interviews.

D1 – Interview protocol general insight

Date, time
Interviewee: Name (department)

1. Introduction

Personal introdcution of interviewer and interviewee.
General explanation of the research project.

2. Problem scoping

General questions related to ALM and LCM and issues arround it.

- What is ALM and LCM? Explain what they mean.
- What are the main isseus related to ALM/LCM you encounter during your work?
- What are the biggest general problems related to ALM/LCM?
- Are there problems between the cooperation between I&O and I&C? If yes, explain.
- How are these problems currently handled?

3. Future view

Questions about whether the interviewee has idea's for the future to examine whether his or her knowledge can be usable in a later phase of the project.

- Do you have an idea of a better solution to handle some of the mentioned problems?
- Who do you think should be responcible for implementing the changes?
- Additional questions to clarify specific changes mentioned.

4. Other

- Do you have additional information that can be usefull for gaining more insight into these problems?
- Does the interviewee want to help with the next phases of the project?
- Explain what will be done with the outcome of these questions.

D2 – Interview protocol selected members of taskforce

During this interview specific questions were asked to gain information on the aspects goals, processes and data.

Date, time
Intervieuwee: Name (department) and specialism (ALM/LCM)

1. Introdruction

Explain what the outcome of the first round of interviews was and why this interviewee was selected for the this round of interviews.

Give an update of the research project.

2. Questions on aspects

- What are your goals related to the work you do?
- Can you explain your work process?
  This is done using a whiteboard where the actual process is drawn by the interviewee.
- What are the triggers for your process?
- Per step of the process, can you explain what data you need or use?
- Per step of the process, how do you save/store this data?
- Additional questions to gain more insight into the different aspects

3. Additional questions

- What are the biggest problems you encounter during your work (related to ALM/LCM)?
- Do you think harmonization of the processes will help?
- Additional questions to gain more insight into specifics mentioned that were not clear.

4. Other

- Do you have additional information that can be usefull for gaining more insight into these problems?
- Does the interviewee want to help with the next phases of the project (explain next steps)?
- Explain what will be done with the outcome of these questions.

Appendix E – Goals

Balanced scorecard SNS

Dimension	*Object*	*Name*	*Description*	*Measure*	*Target (2020)*
Customer perspective *(People-oriented banking)*	C1	Work values	Employees apply work values; like know your customer and be assertive	Assessment cycle	All employees score at least a 'good' on work values
	C2	People oriented	Employees know how to act in a people oriented way	Employee survey	Grade equal or higher than 8
	C3	User-friendly client processes	The client processes are client oriented and easy to understand	Customer survey	Grade equal or higher than 8 for top 10 client processes
Social perspective	S1	Environmentally conscious services	The services are environmentally conscious and add usefulness for customers	Customer survey	Grade equal or higher than 8
	S2	Simple organization model	Value is created through using a simple organization model	Return and Cost	ROE=COE[5] C/I[6] < 50%
	S3	Financial stability	Actively help people to increase the financial stability	Financial check	100% of customers did the financial check-up
Environmental perspective	E1	Sustainability	Customers sustain the living environment	Energy label	Energy label B or higher for all mortgage customers
	E2a	Climate neutral	The business is climate neutral	CO2 emission	Emission is 50% of the emission in 2014
	E2b	Climate neutral	The balance sheet exposures are climate neutral	CO2 benefits/CO2 costs	>30%

[5] ROE = Return on Equity and COE = Cost of Equity
[6] C/I = Cost Income Ratio

As can be derived from the balanced scorecard of SNS the goals are so high level that they do not specifically say anything about processes or harmonization of them.

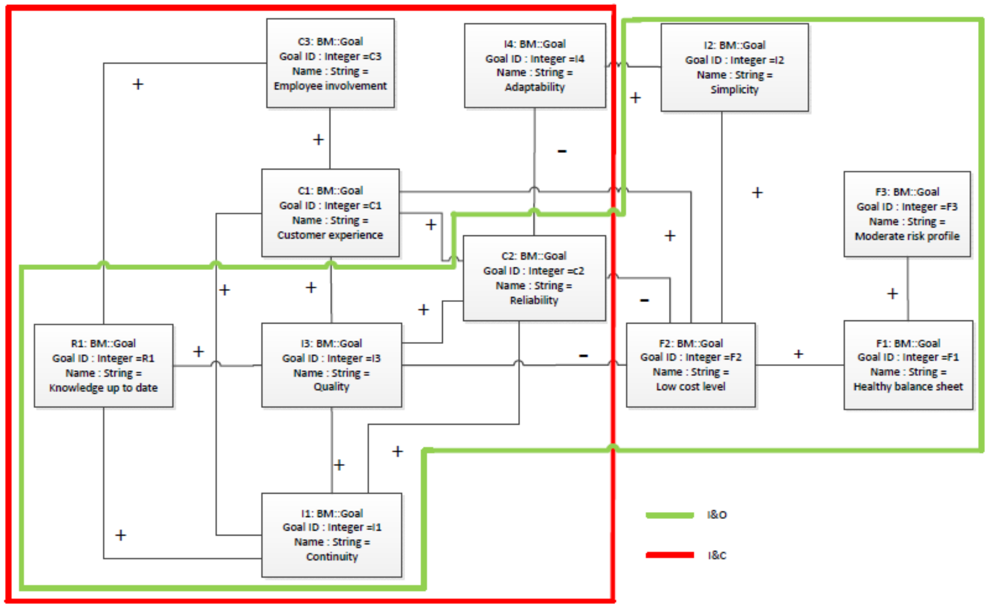
To make it more clear what the relationship between the goals of ITC and SNS are. This overview is created by the researcher in close collaboration with the company supervisors. These relations were not identified before. An overview of the relations can be found below.

ITC goal	Related SNS goal
Healthy balance sheet	S2, S3, E2b
Low cost level	S2
Moderate risk profile	S2
Customer experience	C3, S1, S3
Reliability	E1, C1, S2
Employee involvement	C1, C2
Continuity	S1, S3, E1
Simplicity	C3, S2
Quality	S2, C1, C2, C3
Maneuverability	C1, C2, S2
Knowledge up to date	C3, S1, S2, E1

The balanced scorecard of ITC created in the main document leads to the goal network. For the division between I&O and I&C there was also a user goal model created.

User	*User goal*	*Goal ITC*
I&O	O1: Implement changes with a short time to market	I1
	O2: Facilitate automated business processes	I1, I2
	O3: Create reliable, safe, simple and affordable IT solutions	I3, C2, F2, I2
	O4: Support employees to keep their knowledge up to date	R1, I3
	O5: Keep exploitation costs equal or lower than they are now	F2, F1
	O6: Create better insight of costs and revenue	F1, F3
I&C	C1: Create possibilities for digital client contact	C1
	C2: Use a way of working that is relevant, takes into account privacy and transparency	C2, I3, C3
	C3: Focus on R&D for new technology possibilities	R1, I1, I4
	C4: Support initiatives from teams and customers	C3, C1
	C5: Create an as high as possible availability for the customer	C1, I1
	C6: Maintain a continues and reliable service	C1, C2, I1

Based on this table there is another goal network created where the user goal model with the distinction between I&O and I&C is depicted.

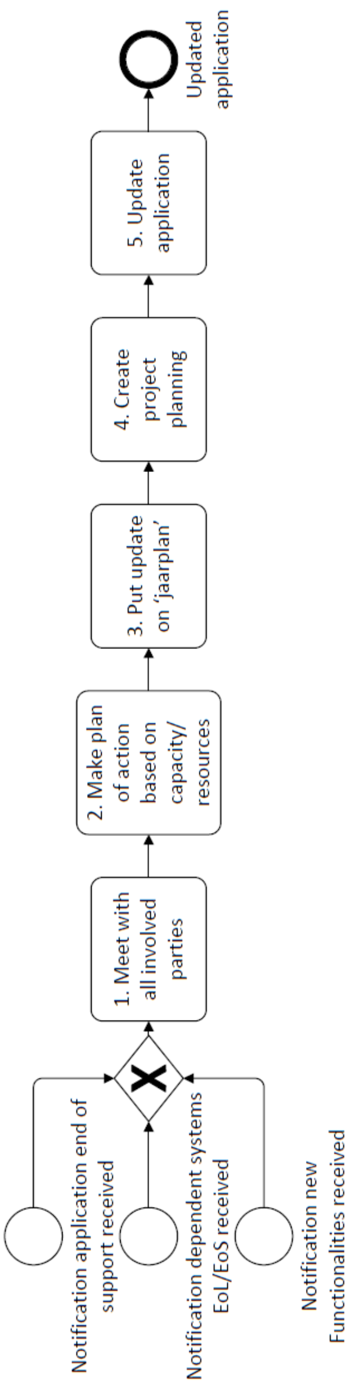


As can be derived from the goal network (colors), the goals of I&O are more related to the ITC goals like simplicity and costs, whereas the goals of I&C are more related to customer experience and maneuverability. Both departments have reliability and quality as a common goal for multiple of their own goals.

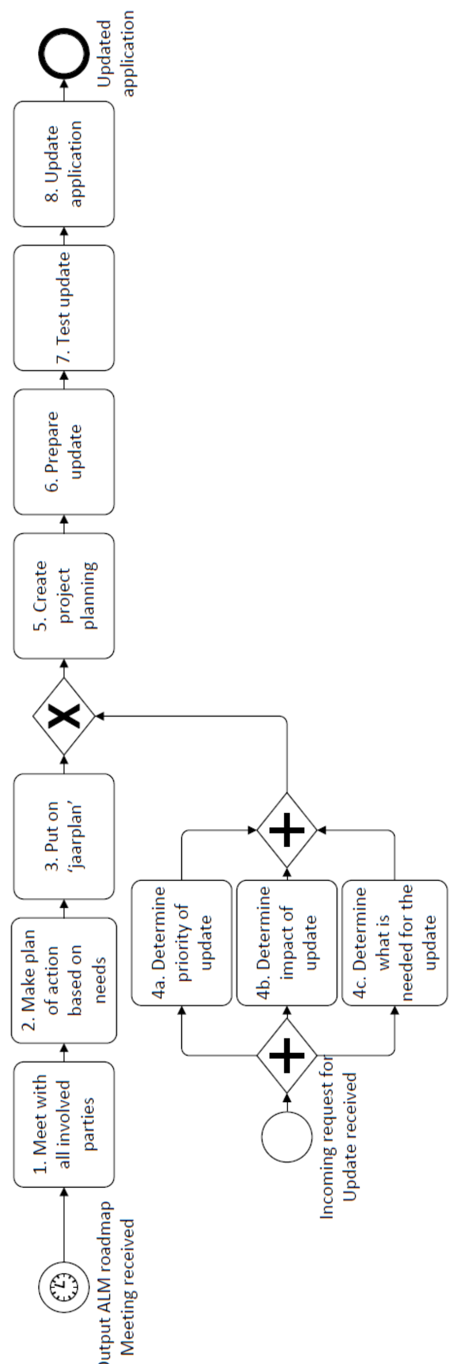
Appendix F - Processes

Processes created during the process step

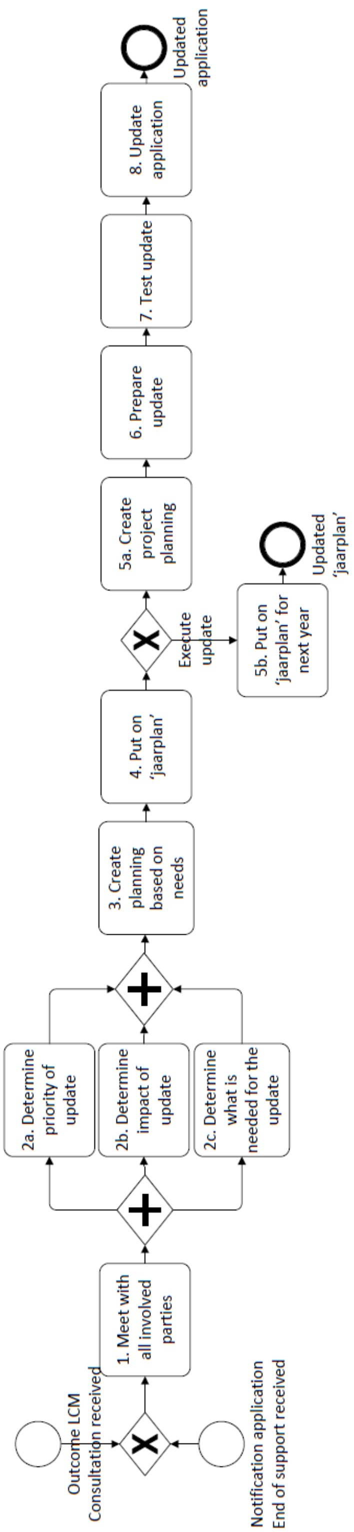
TAB document managment (ALM)



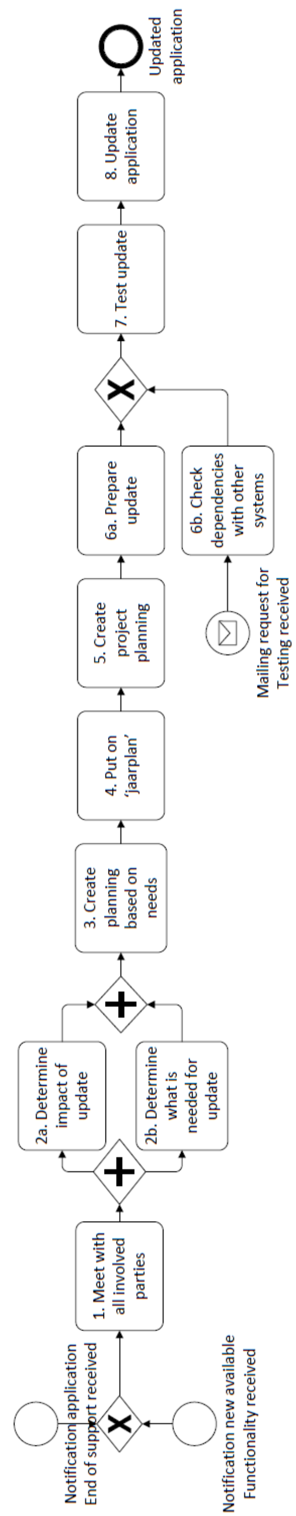
TAB back office (ALM)



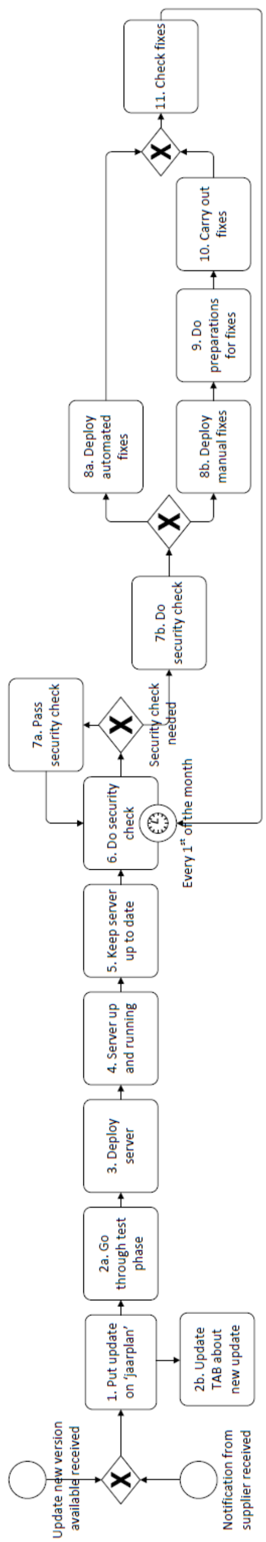
TAB front office (ALM)



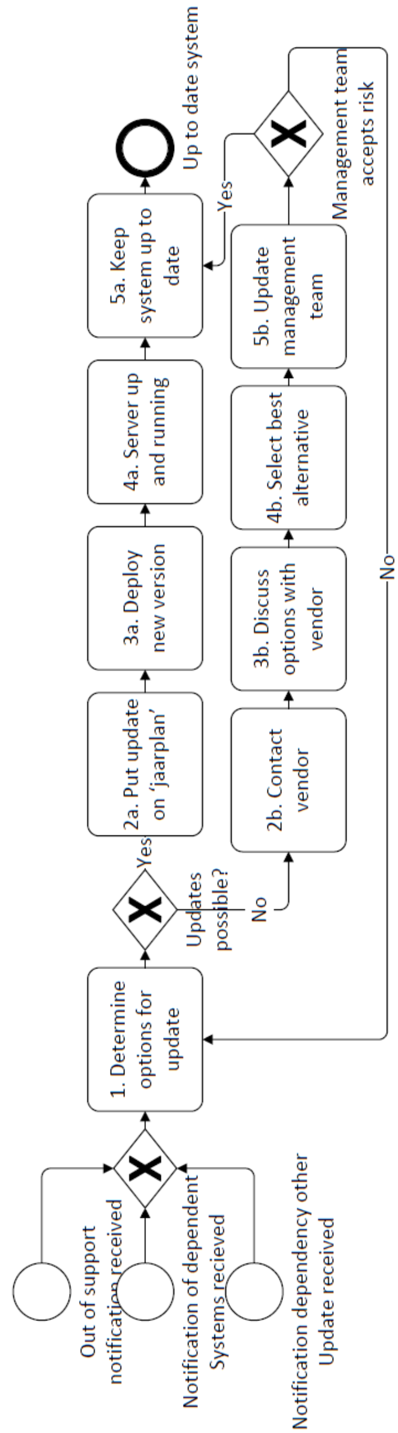
TAB windows applications (ALM)



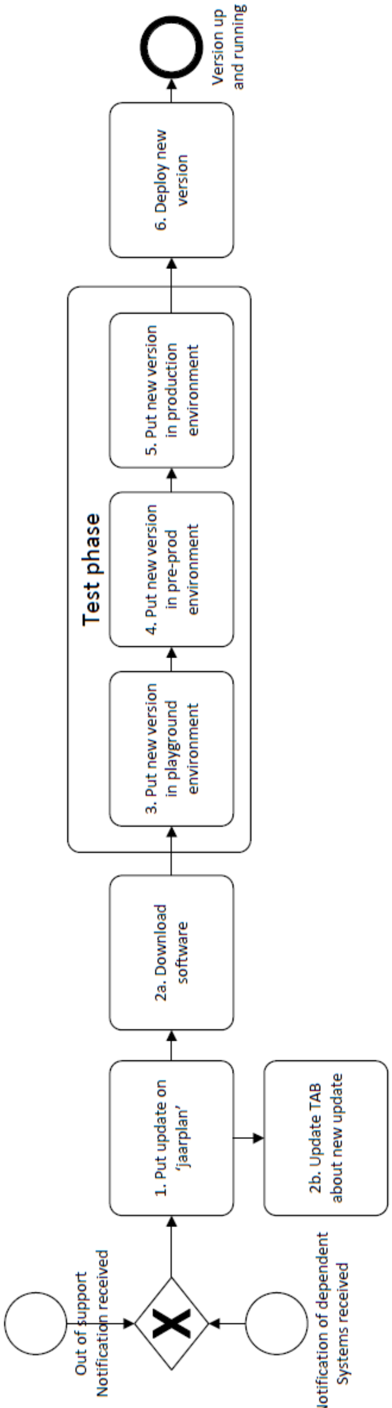
Infrastructure services (LCM)



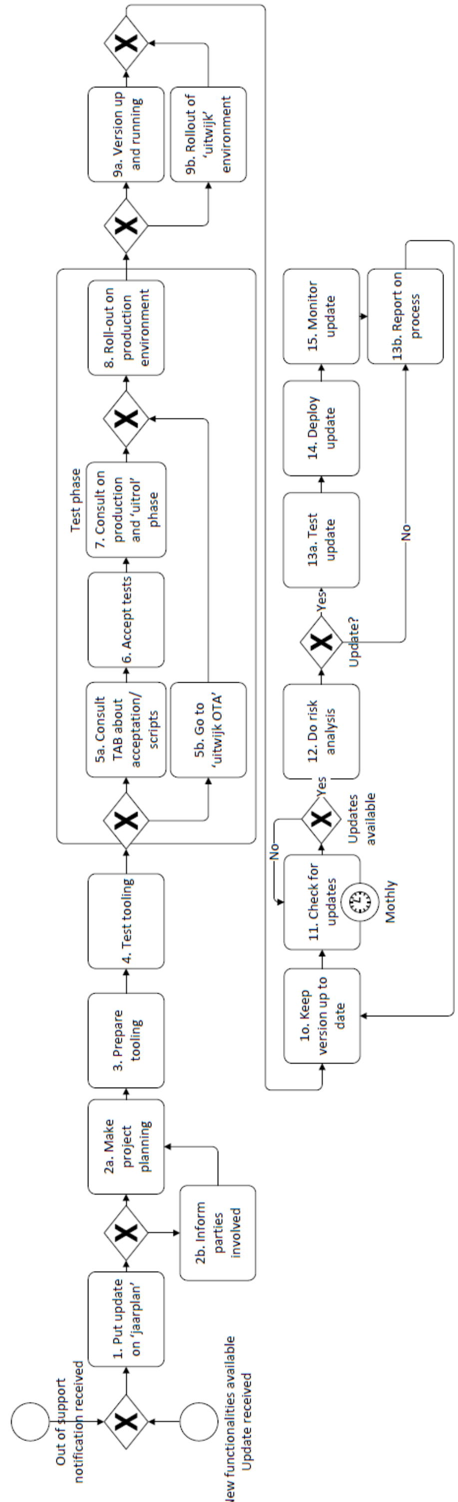
Data warehouse (LCM)



Database architecture (LCM)



System information management (LCM)



Appendix G – RACI matrices

A RACI matrix gives an overview of the roles and responsibilities of all actors or actor groups involved in a process (Haughey). The RACI model has four categories; responsible, accountable, consulted and informed. *Responsible* means that the actor is the one that does the work on the task or has the responsibility for the decisions made. The category *accountable* is the actor that is accountable for the correct and thorough completion of the task. This often is the person that is a project executive or a sponsor. The *consulted* party provides information for the project using two-way communication. This is often related to subject matters where an expert is needed. The final category is the *informed* party. This final category holds people that are kept up to date about the progress of the process. It is often one-way communication and they do not influence the outcome of that specific task or activity.

For this project the selected actors are the own sub-department, technical application management (TAB), functional application management (FAB), suppliers and the management/team leader. These actor categories are selected based on the interviews. An actor is not necessarily the same person, but refer to a person with a certain function. Dependent on the system that needs updating the contact person changes. This is why more general actor categories are selected. When talking about ALM all these actors are taken into account, whereas for LCM, FAB is not taken into account. LCM processes only relate to the infrastructure and have no dependencies with a FAB employee, this contact goes through the TAB.

ALM

Process input technical application management – document management

Activity	*Own department*	*TAB*	*FAB*	*Supplier*	*Management/ Team leader*
Meet with all involved parties	R	C	C	C	I
Make plan of action based on capacity/resources	R,A				
Put update on 'jaarplan'	R	I	I		I
Create project planning	R	I	I		A
Update application	R				

Process input technical application management - back office

Activity	*Own department*	*TAB*	*FAB*	*Supplier*	*Management/ Team leader*
Meet with all involved parties	R	C	C	C	I
Make plan of action based on needs	R,A				
Put on 'jaaplan'	R	I	I		
Determine priority of update	R	C	C	C	
Determine impact of update	R	C	C	C	
Determine what is needed for the update	R	C	C	C	
Create project planning	R				A
Prepare update	R	I	I		
Test update	R	I	I		I

Update application	R

Process input technical application management – front office

Activity	*Own department*	*TAB*	*FAB*	*Supplier*	*Management/ Team leader*
Meet with all involved parties	R	C	C	C	I
Determine priority of update	R	C	C	C	
Determine impact of update	R	C	C	C	
Determine what is needed for update	R	C	C	C	
Create planning based on needs	R				
Put on 'jaarplan'	R				A
Put on 'jaarplan' next year	R				A
Create project planning	R				A
Prepare update	R				
Test update	R	I	I		I
Update application	R				

Process input technical application management – windows applications

Activity	*Own department*	*TAB*	*FAB*	*Supplier*	*Management/ Team leader*
Meet with all involved parties	R,A	C	C	C	I
Determine impact of update	R	C	C	C	
Determine what is needed for update	R	C	C	C	
Create planning based on needs	R				A
Put on 'jaarplan'	R				A
Create project planning	R				
Prepare update	R				
Check dependencies with other systems	R	C,I	C,I		
Test update	R	I	I		
Update application	R				

LCM

Process input infrastructure services

Activity	*Own department*	*TAB*	*Supplier*	*Management/ Team leader*
Put update on 'jaarplan'	R,A	C	C	I
Update TAB about new update	R	I	I	
Go through test phase	R,A	I	I	I
Deploy server	R			
Server up and running	R			
Keep server up to date	R,A			
Do security check	R	C,I	C	I
Deploy automated fixes	A			
Deploy manual fixes	R			

Do preparations for fixes	R	C,I		
Carry out fixes	R			
Check fixes	R,A			

Process input data services – data warehouse

Activity	*Own department*	*TAB*	*Supplier*	*Management/ Team leader*
Determine options for update	R	C	C	I
Put update on 'jaarplan'	R,A	I	I	
Deploy new version	R,A			
Server up and running	A			
Keep system up to date	R			A
Contact vendor	R		C,I	
Discuss options with vendor	R		C	A
Select best alternative	R		I	A
Update management team	R			A

Process input data services – database administration

Activity	*Own department*	*TAB*	*Supplier*	*Management/ Team leader*
Put update on 'jaarplan'	R	I	I	A
Update TAB about new update	R	I		
Download software	R,A			
Put new version in playground environment	R			
Put new version in pre-prod environment	R			
Put new version in production environment	R	I	I	
Deploy new version	R,A			

Process input system information management

Activity	*Own department*	*TAB*	*Supplier*	*Management/ Team leader*
Put update on 'jaarplan'	R	I	I	A
Inform parties involved	R,A	I	I	
Make project planning	R			
Prepare tooling	R	I		
Test tooling	R	C		
Consult TAB about acceptation/scripts	R	C		
Accept tests	R			A
Go to 'uitwijk OTA'	R			

Consult on production and 'uitrol' phase	R	C	C	
Roll-out on production environment	R			
Version up and running	R			
Rollout of 'uitwijk' environment	R			
Keep version up to date	R			A
Check for updates	R		C	
Do risk analysis	R	C	C	
Test update	R	I,C		
Deploy update	R,A			
Monitor update	R			
Report on process	R			I

For the processes of ALM as well as for LCM, the own department states that they are responsible for all steps. There was also stated that they did not think other departments could get the responsibility over the process, because they are not aware of what needs to happen. By keeping your own way of working to yourself, this can lead to problems when people are leaving the company or when different processes get combined. When a new process gets created it will be important that people are more aware of what the other parties tasks are and how their own processes influence others.

Appendix H - Explanation process steps

The bold title is the activity that needs to happen. After this an explanation is given about what happens during the activity and what information flows into the step and is output of the activity. If needed, the key controls are explained and the departments involved with the step are stated.

ALM

Process TAB document management

Meet with all involved parties	
What happens	Meetings are set with all involved parties about the major updates that are planned for the upcoming year. The big releases and updates are discussed and decisions are made on when the update should take place. All other departments can give input on their own updates and releases and dependencies are discussed.
Inflow/outflow	Inflow: End of support or End of life date Outflow: timeframe/ period update will happen (Q1 –Q4)
Key controls	All End of support/End of life dates are discussed and planned
Department	Document management, TAB, FAB and the supplier

Make plan of action based on needs	
What happens	During this step internal decisions are made on the update that is planned . Here decisions are made on what steps need to be taken to be able to make the update happen. There is also checked how many hours are needed, who will be placed on this task and what further resources are needed to finish the update.
Inflow/outflow	Inflow: timeframe planned update, employee planning, available resources Outflow: detailed plan of action, employee coupled to task, needed resources
Key controls	Resources and planning stay within the available resources and planning opportunities
Department	Document management

Put update on 'jaarplan'	
What happens	The update that needs to happen is put on the 'jaarplan' of the department and other dependent departments are informed
Inflow/outflow	Inflow: planning upcoming year Outflow: updated planning upcoming year
Department	Document management

Create project planning	
What happens	The actual planning of the report is created. So what steps are done when and dependent departments are informed of the dates changes will occur.
Inflow/outflow	Inflow: timeframe planned update, plan of action

	Outflow: project planning with specified dates
Department	Document management

Update application	
What happens	After the test phase and all ’ok’ on the tests are received the application is updated. From this point everybody starts working with the updated application.
Inflow/outflow	Inflow: project planning, new application update Outflow: updated application
Key controls	The update works the way it is supposed to work
Department	Document management

Process TAB back office

Meet with all involved parties	
What happens	Meetings are set with all involved parties about the major updates that are planned for the upcoming year. The big releases and updates are discussed and decisions are made on when the update should take place. All other departments can give input on their own updates and releases and dependencies are discussed.
Inflow/outflow	Inflow: End of support or End of life date Outflow: timeframe/ period update will happen (Q1 –Q4)
Key controls	All End of support/End of life dates are discussed and planned
Department	Back office, TAB, FAB and the supplier

Make plan of action based on needs	
What happens	During this step internal decisions are made on the update that is planned . Here decisions are made on what steps need to be taken to be able to make the update happen. There is also checked how many hours are needed, who will be placed on this task and what further resources are needed to finish the update.
Inflow/outflow	Inflow: timeframe planned update, employee planning, available resources Outflow: detailed plan of action, employee coupled to task, needed resources
Key controls	Resources and planning stay within the available resources and planning opportunities
Department	Back office

Put update on ‘jaarplan’	
What happens	The update that needs to happen is put on the ‘jaarplan’ of the department and other dependent departments are informed
Inflow/outflow	Inflow: planning upcoming year Outflow: updated planning upcoming year
Department	Back office

Determine priority of update	
What happens	During this step there is checked how important the update is and thus, whether it should get priority over other projects that are already planned. This is done by checking what the trigger of the update is. For instance if this is a safety concern this makes the priority very high, however if the trigger is a yearly update with almost no changes, the priority becomes low and other projects or updates get priority over this one.
Inflow/outflow	Inflow: triggers for the update Outflow: priority
Department	Back office, TAB, FAB and supplier

Determine what is needed for the update	
What happens	During this step there is checked what resources are needed to be able to execute the update. This can be working hours, external input and so on. A list is created with the resources that are needed for doing the update.
Inflow/outflow	Inflow: type of update Outflow: list of resources needed for the update
Key controls	All resources need to stay within the resource capacities
Department	Back office

Determine impact of update	
What happens	During this step the impact of the update is determined. This is done using the type of update
Inflow/outflow	Inflow: type of update Outflow: impact
Department	Back office

Create project planning	
What happens	The actual planning of the report is created. So what steps are done when and dependent departments are informed of the dates changes will occur.
Inflow/outflow	Inflow: timeframe planned update, plan of action Outflow: project planning with specified dates
Department	Back office

Prepare update	
What happens	The new update is downloaded or created and all dependent parties are informed. This is done so that everybody that needs to use the new system is aware of the upcoming test phase. It is important that the right version of the new application or software is ready for the test phase.
Inflow/outflow	Inflow: new update of application

	Outflow: prepped update for use within the company
Department	Back office

Test update	
What happens	During the test phase the new application is tested multiple times. This is done by the users that are going to use the actual system. They can test the new application on a safe testing environment and if there are changes that need to be made to the new application this can be done and tested again.
Inflow/outflow	Inflow: date test environment can be used Outflow: totally tested application
Key controls	Ok form all testers
Department	Back office

Update application	
What happens	After the test phase and all ’ok’ on the tests are received the application is updated. From this point everybody starts working with the updated application.
Inflow/outflow	Inflow: project planning, new application update Outflow: updated application
Key controls	The update works the way it is supposed to work
Department	Back office

Process TAB front office

Meet with all involved parties	
What happens	Meetings are set with all involved parties about the major updates that are planned for the upcoming year. The big releases and updates are discussed and decisions are made on when the update should take place. All other departments can give input on their own updates and releases and dependencies are discussed.
Inflow/outflow	Inflow: End of support or End of life date Outflow: timeframe/ period update will happen (Q1 –Q4)
Key controls	All End of support/End of life dates are discussed and planned
Department	Front office, TAB, FAB and the supplier

Determine priority of update	
What happens	During this step there is checked how important the update is and thus, whether it should get priority over other projects that are already planned. This is done by checking what the trigger of the update is. For instance if this is a safety concern this makes the priority very high, however if the trigger is a yearly update with almost no changes, the priority becomes low and other projects or updates get priority over this one.
Inflow/outflow	Inflow: triggers for the update

	Outflow: priority
Key controls	
Department	Front office, TAB and FAB

Determine impact of update	
What happens	During this step the impact of the update is determined. This is done using the type of update
Inflow/outflow	Inflow: type of update Outflow: impact
Key controls	
Department	Front office

Determine what is needed for the update	
What happens	During this step there is checked what resources are needed to be able to execute the update. This can be working hours, external input and so on. A list is created with the resources that are needed for doing the update.
Inflow/outflow	Inflow: type of update Outflow: list of resources needed for the update
Key controls	All resources need to stay within the resource capacities
Department	Front office, TAB and FAB

Create planning based on needs	
What happens	Based on the input from the previous steps a planning is created for the update. This planning includes how many hours are needed for the project, when the exact update will take place (within the determined timeframe) and what can go wrong.
Inflow/outflow	Inflow: priority, impact and resources needed Outflow: specified planning
Department	Front office

Put on ‘jaarplan’	
What happens	The update that needs to happen is put on the ‘jaarplan’ of the department and other dependent departments are informed
Inflow/outflow	Inflow: planning upcoming year Outflow: updated planning upcoming year
Department	Front office

Put on ‘jaarplan’ next year	
What happens	If an update cannot be done this year for whatever reason, but it is still important, it will be put on the ‘jaarplan’ of next year. It will be again discusses during the meeting with all parties, but it will definitely be on the planning
Inflow/outflow	Inflow: Update cannot be updated during current year

	Outflow: updated ‘jaarplan’ upcoming year plus one
Key control	Update is placed on planning
Department	Front office

Create project planning	
What happens	The actual planning of the report is created. So what steps are done when and dependent departments are informed of the dates changes will occur.
Inflow/outflow	Inflow: timeframe planned update, plan of action Outflow: project planning with specified dates
Department	Front office

Prepare update	
What happens	The new update is downloaded or created and all dependent parties are informed. This is done so that everybody that needs to use the new system is aware of the upcoming test phase. It is important that the right version of the new application or software is ready for the test phase.
Inflow/outflow	Inflow: new update of application Outflow: prepped update for use within the company
Department	Front office

Test update	
What happens	During the test phase the new application is tested multiple times. This is done by the users that are going to use the actual system. They can test the new application on a safe testing environment and if there are changes that need to be made to the new application this can be done and tested again.
Inflow/outflow	Inflow: date test environment can be used Outflow: totally tested application
Key controls	Ok form all testers
Department	Front office

Update application	
What happens	After the test phase and all ’ok’ on the tests are received the application is updated. From this point everybody starts working with the updated application.
Inflow/outflow	Inflow: project planning, new application update Outflow: updated application
Key controls	The update works the way it is supposed to work
Department	Front office

Process TAB windows applications

Meet with all involved parties

What happens	Meetings are set with all involved parties about the major updates that are planned for the upcoming year. The big releases and updates are discussed and decisions are made on when the update should take place. All other departments can give input on their own updates and releases and dependencies are discussed.
Inflow/outflow	Inflow: End of support or End of life date Outflow: timeframe/ period update will happen (Q1 –Q4)
Key controls	All End of support/End of life dates are discussed and planned
Department	Windows applications, TAB, FAB and the supplier

Determine impact of update	
What happens	During this step the impact of the update is determined. This is done using the type of update
Inflow/outflow	Inflow: type of update Outflow: impact
Department	Windows applications

Determine what is needed for the update	
What happens	During this step there is checked what resources are needed to be able to execute the update. This can be working hours, external input and so on. A list is created with the resources that are needed for doing the update.
Inflow/outflow	Inflow: type of update Outflow: list of resources needed for the update
Key controls	All resources need to stay within the resource capacities
Department	Windows applications, TAB and FAB

Create planning based on needs	
What happens	Based on the input from the previous steps a planning is created for the update. This planning includes how many hours are needed for the project, when the exact update will take place (within the determined timeframe) and what can go wrong.
Inflow/outflow	Inflow: impact and resources needed Outflow: specified planning
Department	Windows applications

Put on ‘jaarplan’	
What happens	The update that needs to happen is put on the ‘jaarplan’ of the department and other dependent departments are informed
Inflow/outflow	Inflow: planning upcoming year Outflow: updated planning upcoming year
Department	Windows applications

Create project planning	
What happens	The actual planning of the report is created. So what steps are done when and dependent departments are informed of the dates changes will occur.
Inflow/outflow	Inflow: timeframe planned update, plan of action Outflow: project planning with specified dates
Department	Windows applications

Prepare update	
What happens	The new update is downloaded or created and all dependent parties are informed. This is done so that everybody that needs to use the new system is aware of the upcoming test phase. It is important that the right version of the new application or software is ready for the test phase.
Inflow/outflow	Inflow: new update of application Outflow: prepped update for use within the company
Department	Windows applications

Check dependencies with other systems	
What happens	If an ad hoc change in another department influences the processes of windows applications they have to test the updates for the dependent systems. When this is done an ‘ok’ is given to the sender of the request to test.
Inflow/outflow	Inflow: request to test application Outflow: ‘ok’ on all tests
Department	Windows applications

Test update	
What happens	During the test phase the new application is tested multiple times. This is done by the users that are going to use the actual system. They can test the new application on a safe testing environment and if there are changes that need to be made to the new application this can be done and tested again.
Inflow/outflow	Inflow: date test environment can be used Outflow: totally tested application
Key controls	‘Ok’ form all testers
Department	Windows applications

Update application	
What happens	After the test phase and all ’ok’ on the tests are received the application is updated. From this point everybody starts working with the updated application.
Inflow/outflow	Inflow: project planning, new application update Outflow: updated application
Key controls	The update works the way it is supposed to work
Department	Windows applications

LCM

Process infrastructure services

Put update on ‘jaarplan’	
What happens	During this step the updates that need to happen are put on the ‘jaarplan’ for the upcoming year. This is done by planning it into a period of the year (so Q1-Q4).
Inflow/outflow	Inflow: notification end of life or end of support system Outflow: updated ‘jaarplan’ upcoming year
Key controls	All updates with an end of life or end of support date are planned
Department	Infrastructure services

Update TAB about new update	
What happens	The TAB department that deploys dependent applications is notified about when certain updates are planned.
Inflow/outflow	Inflow: updated ‘jaarplan’ Outflow: updated TAB
Department	Infrastructure services

Go through test phase	
What happens	During this phase the new update is tested. There are several steps that need to be taken when testing the new update, because there are multiple test moments and environments. For all tests there is a check that has to be checked to continue.
Inflow/outflow	Inflow: update that needs to be tested Outflow: ‘ok’ on all tests
Key controls	All tests are done and got an ‘ok’
Department	Infrastructure services

Deploy server	
What happens	All tests have received an ‘ok’ and the update can be deployed.
Inflow/outflow	Inflow: ‘ok’ on all tests Outflow: deployed update
Department	Infrastructure services

Server up and running	
What happens	The update is deployed and now the new update is working. Then it should all work without bugs and problems.
Inflow/outflow	Inflow: deployed update Outflow: working server
Key controls	Server is working predetermined % of time
Department	Infrastructure services

Keep server up to date	
What happens	When the server is up and running it is important that it is checked regularly for problems that could occur or output that gets generated that is not correct.
Inflow/outflow	Inflow: working server Outflow: annotation of problems if they occurred
Key controls	For all problems an annotation is created
Department	Infrastructure services

Do security check	
What happens	There is a monthly checkup to see whether an update is available that needs to be implemented and whether it is necessary to do fixes.
Inflow/outflow	Inflow: monthly checkup moment Outflow: does fix need to happen
Key controls	Is the security pass passed for the update
Department	Infrastructure services

Deploy automated fixes	
What happens	When an update or fix needs to happen this can be done automated. Automated updates only need to be started and then the fix is completed after some time.
Inflow/outflow	Inflow: fix needs to happen Outflow: notification automated fix is done
Department	Infrastructure services

Deploy manual fixes	
What happens	When an update or fix needs to happen this can be done manually. Manual updating means that the different steps that need to happen to carry out the fix need to be started or done manually. During the deploy step the fix needs to be downloaded.
Inflow/outflow	Inflow: fix needs to happen Outflow: downloaded fix
Key controls	Is it the correct version of the fix
Department	Infrastructure services

Do preparations for fixes	
What happens	The downloaded fixes need to be manually adapted to the needs of the company. This can mean that they have to be able to link to other programs or applications. Therefore the TAB employee of the linked departments are consulted and informed about the made choices
Inflow/outflow	Inflow: downloaded fix Outflow: prepared fix
Key controls	All features the fix should have are there
Department	Infrastructure services and TAB

Carry out fixes	
What happens	During this step the fixes are implemented and carried out in the system. From this point all employees start working with the system including the new fix.
Inflow/outflow	Inflow: prepared fix Outflow: installed fix
Department	Infrastructure services

Check fixes	
What happens	The fixes are up and working and now the system including the fix needs to be checked regularly.
Inflow/outflow	Inflow: system with installed fix Outflow:
Key controls	Stay above minimal availability system
Department	Infrastructure services

Process data warehouse

Determine options for update	
What happens	The first step in the process for data warehouse is the selection and determination of different update options. All options are considered and all related actors are consulted.
Inflow/outflow	Inflow: trigger update is needed Outflow: best option for updating
Department	Data warehouse

Put update on 'jaarplan'	
What happens	During this step the updates that need to happen are put on the 'jaarplan' for the upcoming year. This is done by planning it into a period of the year (so Q1-Q4).
Inflow/outflow	Inflow: notification end of life or end of support system Outflow: updated 'jaarplan' upcoming year
Key controls	All updates with an end of life or end of support date are planned
Department	Data warehouse

Deploy server	
What happens	All tests have received an 'ok' and the update can be deployed.
Inflow/outflow	Inflow: 'ok' on all tests Outflow: deployed update
Department	Data warehouse

Server up and running

What happens	The update is deployed and now the new update is working. Then it should all work without bugs and problems.
Inflow/outflow	Inflow: deployed update Outflow: working server
Key controls	Server is working predetermined % of time
Department	Data warehouse

Keep system up to date	
What happens	When the system is up and running it is important that it is checked regularly for problems that could occur or output that gets generated that is not correct.
Inflow/outflow	Inflow: working system Outflow: annotation of problems if they occurred
Key controls	For all problems an annotation is created
Department	Data warehouse

Contact vendor	
What happens	When during the determination of the option for updating the scenario appears that an update is not possible or the wished for update cannot be realized the vendor is contacted/supplier is consulted.
Inflow/outflow	Inflow: the wished for update is not possible Outflow: the vendor/supplier is contacted
Department	Data warehouse

Discuss options with vendor	
What happens	The vendor/supplier is asked what the options are, because they have to be able to support the system after the update. The vendor comes with options that can be selected.
Inflow/outflow	Inflow: contacted vendor/supplier Outflow: options vendor/supplier
Department	Data warehouse

Select best alternative	
What happens	From the alternatives the vendor/supplier gave the best solution has to be selected. This usually means that the management gets accountable for the choice that is made. During this step the best option is selected.
Inflow/outflow	Inflow: options from vendor/supplier Outflow: selected option
Key controls	All parties agree to the selected option
Department	Data warehouse

Update management team	
What happens	The selected step is not explained to the management and they can choose to either accept the option with all its risks, or decide to not take the risks. If the risk is taken the system will be updated accordingly.
Inflow/outflow	Inflow: selected option Outflow: management takes risks or not
Department	Data warehouse

Process database architecture

Put update on ‘jaarplan’	
What happens	During this step the updates that need to happen are put on the ‘jaarplan’ for the upcoming year. This is done by planning it into a period of the year (so Q1-Q4).
Inflow/outflow	Inflow: notification end of life or end of support system Outflow: updated ‘jaarplan’ upcoming year
Key controls	All updates with an end of life or end of support date are planned
Department	Database architecture

Update TAB about new update	
What happens	The TAB department that deploys dependent applications is notified about when certain updates are planned.
Inflow/outflow	Inflow: updated ‘jaarplan’ Outflow: TAB updated about planning
Department	Database architecture

Download software	
What happens	Just before the update is planned the new software needs to be downloaded and prepared.
Inflow/outflow	Inflow: start date update reached Outflow: prepared software update
Key control	The right software version is downloaded and prepared
Department	Database architecture

Put new version in playground environment	
What happens	The prepared software first has to be tested in the playground environment. Here the new version is put in an environment where the basic features are tested and it is not a problem if things are going wrong, because this environment is totally separated from the work environment of SNS.
Inflow/outflow	Inflow: prepared software Outflow: new version tested in playground environment
Key control	‘ok’ on all checks that need to happen during this phase of testing
Department	Database architecture

Put new version in pre-prod environment	
What happens	The version that is already tested in the playground environment is now placed in the pre-prod environment. Here the new version is tested in an environment that exactly resembles the actual working environment of SNS. Again a lot of tests are done and final changes are made to the version.
Inflow/outflow	Inflow: new version tested in playground environment Outflow: new version tested in pre-prod environment
Key control	‘ok’ on all checks that need to happen during this phase of testing
Department	Database architecture

Put new version in production environment	
What happens	The version tested in the pre-prod environment is now taken in production. This means that it is taken into daily use. If faults or errors occur this has an actual influence on the business.
Inflow/outflow	Inflow: new version tested in pre-prod environment Outflow: new version in use
Key control	‘ok’ on all checks that need to happen during this phase of testing
Department	Database architecture

Deploy new version	
What happens	The version is kept in use and changes are made if this is necessary based on incidents that might occur. If this does not happen this means that the new update will work (mostly) on the background.
Inflow/outflow	Inflow: new version in use Outflow: working system
Department	Database architecture

Process system information management

Put update on ‘jaarplan’	
What happens	During this step the updates that need to happen are put on the ‘jaarplan’ for the upcoming year. This is done by planning it into a period of the year (so Q1-Q4).
Inflow/outflow	Inflow: notification end of life or end of support system Outflow: updated ‘jaarplan’ upcoming year
Key controls	All updates with an end of life or end of support date are planned
Department	System information management

Inform parties involved	
What happens	The dependent departments that handle dependent applications are notified about when certain updates are planned.
Inflow/outflow	Inflow: updated ‘jaarplan’ Outflow: updated departments with dependencies
Department	System information management

Make project planning	
What happens	The actual planning of the project is created. So what steps are done when and dependent departments are informed of the dates changes will occur.
Inflow/outflow	Inflow: timeframe planned update Outflow: project planning with specified dates
Department	System information management

Prepare tooling	
What happens	The new tooling is downloaded or created and all dependent parties are informed. This is done so that everybody that needs to use the new system is aware of the upcoming test phase. It is important that the right version of the new application or software is ready for the test phase.
Inflow/outflow	Inflow: new update of infrastructure Outflow: prepped tooling for testing
Department	System information management

Test tooling	
What happens	During the test phase the new tooling is tested multiple times. This is done by the users that are going to use the actual system. They can test the new tooling on a safe testing environment and if there are changes that need to be made to the new tooling this can be done and tested again.
Inflow/outflow	Inflow: date test environment can be used Outflow: totally tested tooling
Key controls	Ok form all testers
Department	System information management

Consult TAB about acceptation/scripts	
What happens	The tests that are done using the previous steps are checked by the department in collaboration with TAB if there are concerns about the tests. If the tests did not go the way they needed to go the tests will not be seen as sufficient for taking the tooling in use.
Inflow/outflow	Inflow: totally tested tooling Outflow: ‘ok’ on all tests from TAB and the department
Key controls	‘ok’ of all actors
Department	System information management

Accept tests	
What happens	All actors gave an ‘ok’ on the tests so the tests become accepted and the new tooling version continues.
Inflow/outflow	Inflow: ‘ok’ of all actors Outflow: accepted tests of new tooling

Key controls	All ‘ok’ that need to be there are there
Department	System information management

Go to uitwijk OTA	
What happens	If there are no detected problems with the test phase the uitwijk OTA is created. This is needed for the case that something goes wrong in a later stadium and the system needs to stay active. Than the uitwijk OTA can be used as a buffer.
Inflow/outflow	Inflow: accepted tests Outflow: completed uitwijk OTA
Key controls	Uitwijk OTA is working
Department	System information management

Consult on production and ‘uitrol’ phase	
What happens	for the accepted tests now the production and uitrol environments needs to be created. How this is done is consulted at TAB. Together they create a plan for this.
Inflow/outflow	Inflow: accepted tests new tooling Outflow: plan for rollout production environment
Department	System information management

Rollout on production environment	
What happens	The accepted new tooling is rolled out on the production environment and taken into daily use. . If faults or errors occur this has an actual influence on the business.
Inflow/outflow	Inflow: accepted tooling Outflow: new version tooling in use
Key controls	‘ok’ on all checks that need to happen during this phase of testing
Department	System information management

Version up and running	
What happens	The update is deployed and now the new update is working. Then it should all work without bugs and problems.
Inflow/outflow	Inflow: deployed update Outflow: working version
Key controls	Version is working predetermined % of time
Department	System information management

Keep version up to date	
What happens	When the server is up and running it is important that it is checked regularly for problems that could occur or output that gets generated that is not correct.
Inflow/outflow	Inflow: working server Outflow: annotation of problems if they occurred

Key controls	For all problems an annotation is created
Department	System information management

Check for updates	
What happens	There is a monthly checkup to see whether an update is available that needs to be implemented and whether it is necessary to do updates.
Inflow/outflow	Inflow: monthly checkup moment Outflow: does update need to happen
Key controls	Is the security pass passed for the update
Department	System information management

Do risk analysis	
What happens	If an update is available a risk analysis is done to see if it is good for the company to do the update. If the risk is not to high the update will happen. If the risks are too high the update will not happen.
Inflow/outflow	Inflow: update available? Outflow: is update happening
Key controls	Is the risk level lower that the risk that can be taken
Department	System information management

Test update	
What happens	During this phase the new update is tested. There are several steps that need to be taken when testing the new update, because there are multiple test moments and environments. For all tests there is a check that has to be checked to continue.
Inflow/outflow	Inflow: update that needs to be tested Outflow: ‘ok’ on all tests
Key controls	All tests are done and got an ‘ok’
Department	System information management

Deploy update	
What happens	The version is kept in use and changes are made if this is necessary based on incidents that might occur. If this does not happen this means that the new update will work (mostly) on the background.
Inflow/outflow	Inflow: Outflow: new version in use
Department	System information management

Monitor update	
What happens	When the update is in use the system always stays monitored for bugs or attacks.
Inflow/outflow	Inflow: working update

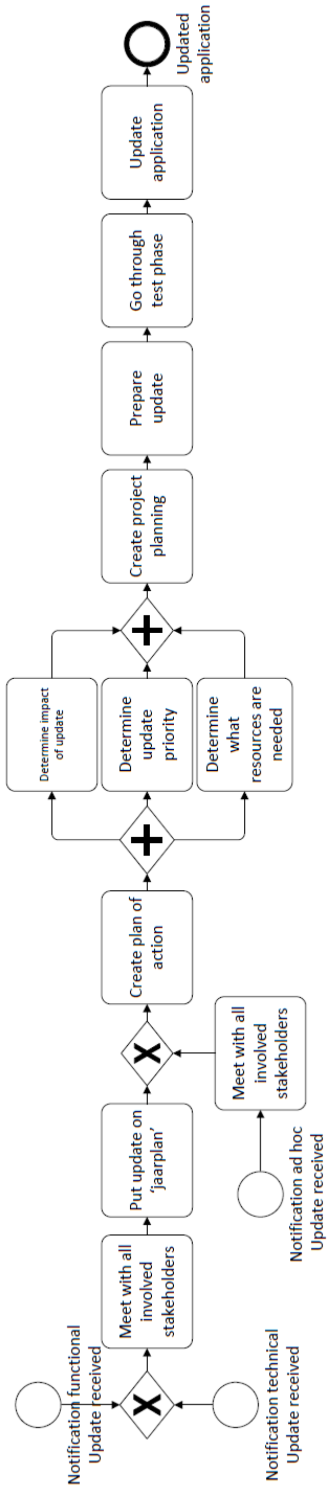
	Outflow: signal if there is a problem
Key controls	System needs to work a pre discussed amount of time
Department	System information management

Report on process	
What happens	All choices and steps taken during the process are noted and checked. This can be a document that is already in use and gets adapted or a newly created document.
Inflow/outflow	Inflow: steps whole process Outflow: document with decisions and taken steps
Key controls	For all updates a document needs to be created/updated
Department	System information management

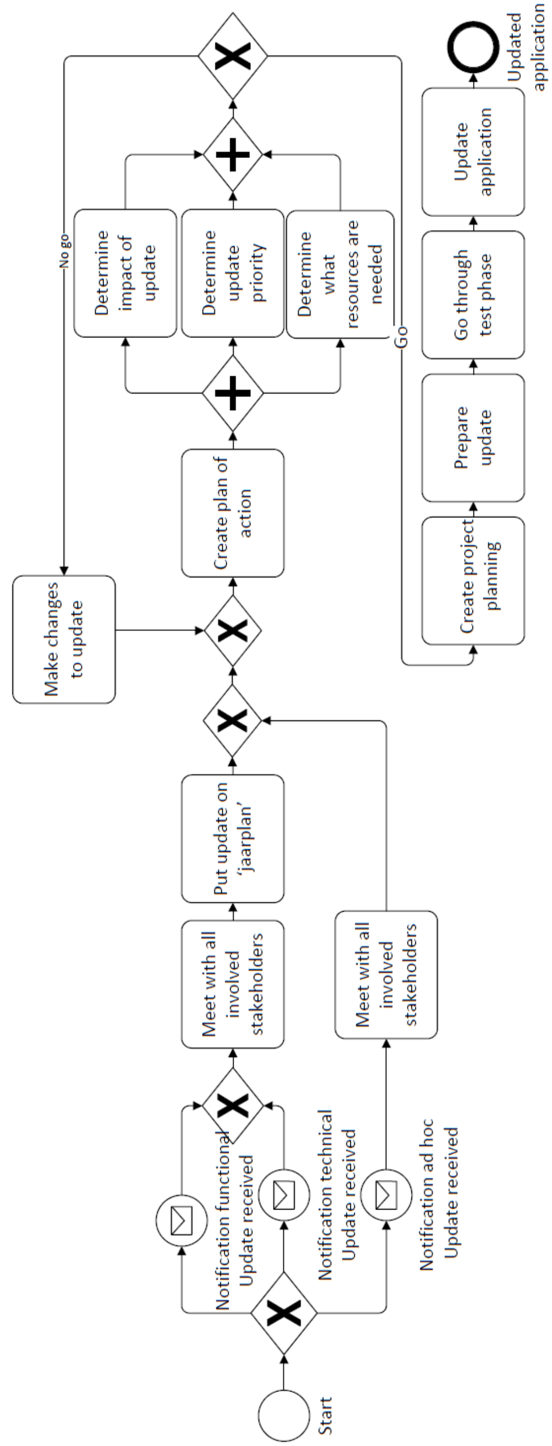
Appendix I – Processes design step 2

ALM

First version

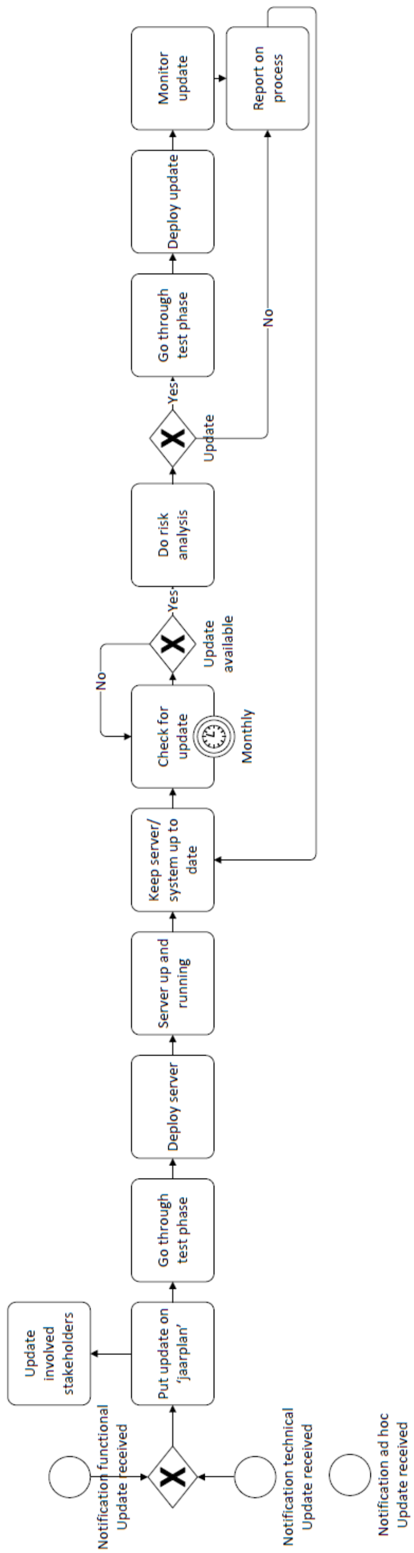


Final

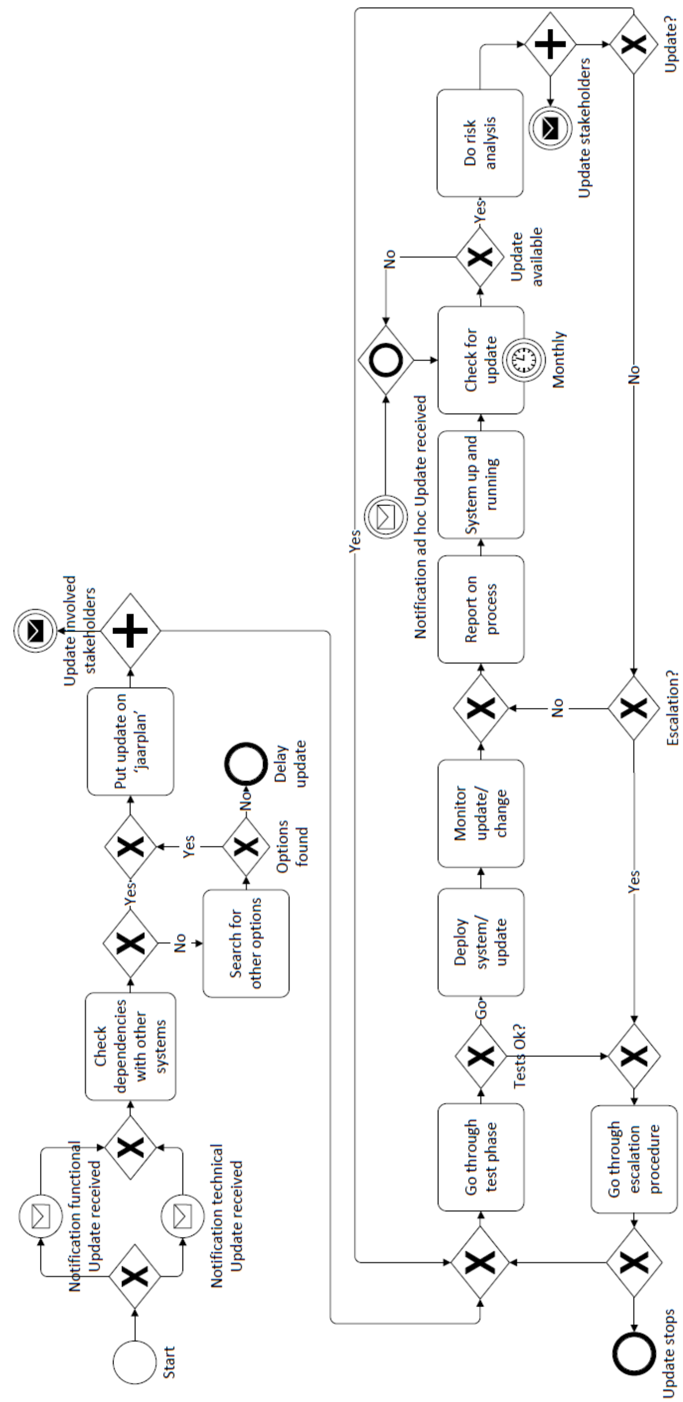


LCM

First version



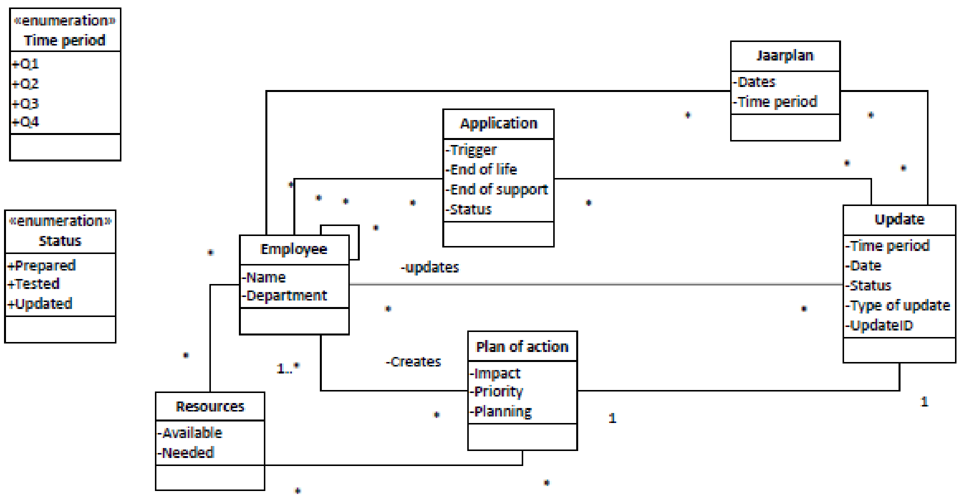
Final



Appendix J - Changes in links

ALM

The test phase needs to be added to the data model, because this was determined during the preliminary steps. The test entity was not included before, because it was only mentioned twice during the analysis phase.



The next section will describe the influence of changing the process on the link between the different aspects. The link between the goals and the processes will not change, because the outcome of the process did not change. The main goal of the process still is to update the application. The link between the processes and the data does change, because both aspects were adapted during the design. Because there is only one process, there is only one CRUD matrix that can be used by all employees working with the new process. This would represent the optimal situation, in a real setting it could be the case that different departments use different CRUD matrices for the same process. This could also vary over the type of update or other differences that can be found between the updates. For this research the optimal solution is used.

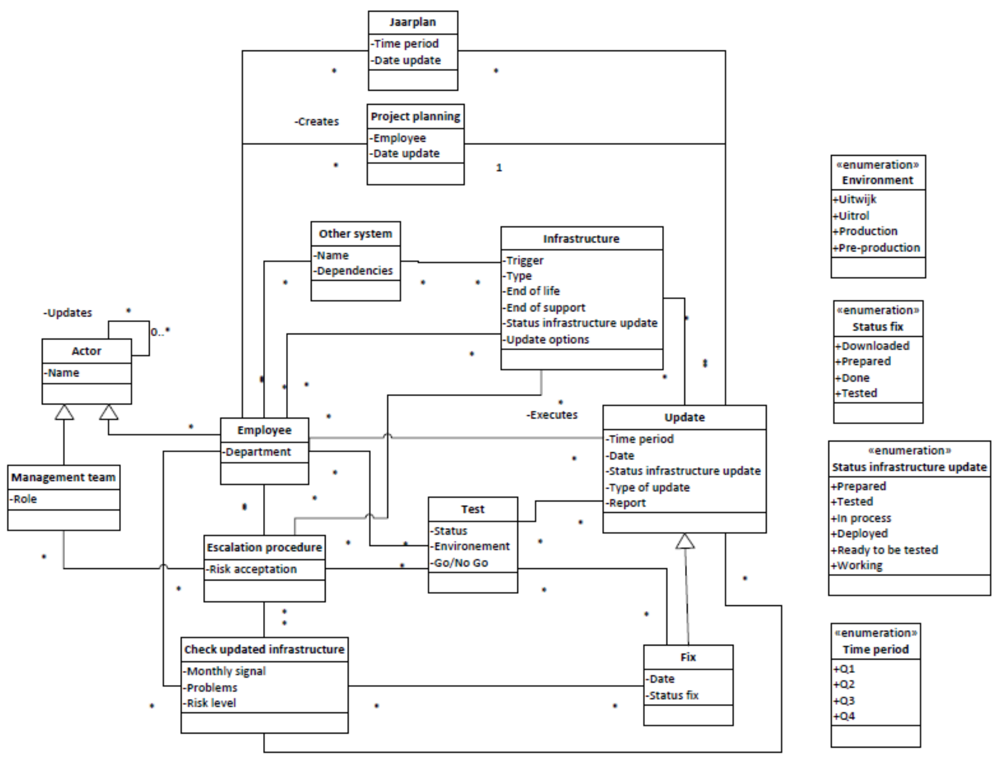
Activity	Entity						
	Employee	Resources	Plan of action	Application	Year planning	update	Test
Put update on 'jaarplan'				R	U		
Create plan of action			R,U		R		
Make changes to update		R	R	R		U	
Determine impact of update			U				
Determine update priority			U				

Determine what resources are needed		R	U				
Create project planning		R,U	C				
Prepare update		R	R			R,U	
Go through test phase				R		U	R
Update application				U		U	

The final link is the link between the goals and the data. Because the organization does not change overnight, there is no change in which goals are measured or not. This was not taken into account when creating the new data model.

LCM

Because the data models are directly related to the processes, the changes in the data model will be discussed first. When taking into account the final process model, the data model would be the same for all departments working on LCM. This data model was created using the old one as the starting point.



The most important aspect to mention is that of saving the data. When everybody works with the same process, the basis of the saved data should be the same. However, this can only be the case if the data is actually stored and visible for all parties involved. For this new data model the assumption is made that it is normal for employees to store the data in a shared location. As can be derived from the new data model, the basis is the same as the old data model. This can be explained, because the old data model already was a merge of the data used during the four explained LCM processes. However, there are some changes. The first change is that the vendor and the vendor options are gone. This path was only mentioned by one member of the taskforce, which meant that it is not included in the new model. The second change is related to the adding of the escalation procedure. This was added in the process, which means that it also needs to be added in the data model. As can be derived from the process, the escalation can happen during various phases of the process. This is why this entity is quite vague, because it can be executed in different ways during the process. Besides the infrastructure the update is about, the dependencies with other systems also need to be monitored. This led to the new entity ‘other systems’. Finally the test phase now has to send out a go/no go sign to be able to move on in the process.

Now the changes of the link between the processes and the data is explained. The link between the processes and the data does change, because both aspects were adapted during step 2 of the design. Because there is only one process, there is only one CRUD matrix that can be used by all employees working with the new process. This would represent the optimal situation, in a real setting it could be the case that different departments use different CRUD matrices for the same process. This could also vary over the type of update or other differences that can be found between the updates. For this research the optimal solution is used.

Activity LCM	**Entity**											
	Employee	Actor	Other system	Management team	Escalation procedure	Jaarplan	Project planning	Infrastructure	Test	Check update infrastructure	Update	Fix
Check dependencies with other systems			R									
Search for other options								R				
Put update on ‘jaarplan’	U	U				U	C	R				
Go through test phase							R	R	R		U	
System up and running								U				
Check for update								R		R		
Do risk analysis										R		
Go through test phase (fix)								R	R			U
Deploy system												U
Monitor update								R		R		
Report on process								R	R	R	C	R
Go through escalation procedure				R	R			R,U				

Appendix K – Harmonized processes