Abstract

Knowledge management has become increasingly popular in the software industry. Knowledge is one of the software companies main asset, and large amounts of resources are being used to manage and re-use this knowledge. Management of architectural knowledge is also important, especially when dealing with software development. This is because a team with good architectural understanding will have a good chance at efficiently creating re-usable assets.

In this thesis I will describe how a Scandinavian software company deals with knowledge management. I have also analyzed the management of architectural knowledge. These subjects have been viewed from both the managers and employees point of view, and I have compared the intentions of the managers, with how the employees actually perform.

The research question: "How is domain- and architecture-knowledge managed in a Scandinavian software company?" is answered by describing and analyzing the data gathered by interviews in such a company. The thesis is concluded by summaries of the discussion and the analysis that has been done. My findings in the researched areas suggest that knowledge management practices are important but that they often are underestimated. The company wherein I have conducted my research does have a QA team and a re-use culture, this culture is described, however this thesis also points out areas in which the company can improve.

The case-study is based upon qualitative analysis of the results from eight interviews conducted among managers and developers in the company. In the thesis I discuss the findings and report upon issues such as company culture, routines and goals in the areas of knowledge management. My findings have been generalized, and hopefully other companies can make use of them to improve their own knowledge management processes and goals.

Acknowledgements

I wish to thank the following people:
Letizia Jaccheri for being my supervisor and assisting me when needed. Tor Erlend Fægri and Torgeir Dingsøyr who work for Sintef. They have helped me with creating the interview guides and the report, they have also spent much of their time reviewing my work. They supplied me with much relevant literature and have shared their knowledge in the domains of research and qualitative data. I wish to thank NTNU for sponsoring my trip to the company where I performed my interviews. I would also like to thank all my interview objects whose names shall remain in the shadows.

_____
Anders Person

Preface

Do you ever wonder how knowledge management is done in other small and medium sized companies? Would you like to compare your own KM initiative with others?

This thesis presents a description of knowledge management in a Scandinavian software company, it also contains research in a little covered area; architecture knowledge management. While writing this thesis I have had the pleasure of learning new theory and comparing it to real life. By interviewing people in a software company I have gained insight in the industry and how things are done. I believe this will enable me to influence my future work locations to employ knowledge management. See appendix 6 for further project evaluation.

Anders Person
Trondheim 23.06.05

Email: ooga40@hotmail.com

If a man empties his purse into his head, no man can take it away from him.
An investment in knowledge always pays the best interest.

-Benjamin Franknlin

Figure list

1 Introduction

1.1 Motivation

Knowledge management is nothing new, it has been used for hundreds of years as owners of family businesses have passed their knowledge to their children, and master craftsmen have taught their skills to their apprentices [Hansen et al, 2002]. Many industries adopted Knowledge Management in the 1990s in connection with new computer technologies. Since the 1990s KM implementation has increased rapidly: 80% of the largest global corporations now have KM projects [Lawton, 2001]. These initiatives had both wonderful and disastrous results and this marked the term "software knowledge management" as something tempting but a little scary.

Subjects like re-use and Software Product Lines (SPL) can give the impression that abstract things like knowledge and experiences can be stored and re-used, and thereby save resources. However there can be several difficulties with introducing such aspects. Software tools are often a requirement for these initiatives, people might have to alter their work-routines and it might be hard to measure and verify that it actually paid off. Aside from the technical difficulties and expenses, the employees have to be persuaded to apply the new methods and break from current routines.

This study is a relevant addition to the numerous theoretical works on software KM. There have been many studies of larger companies with extensive KM initiatives. However this thesis provides an insight in the challenges of managing knowledge in a medium sized software company in Scandinavia. In addition to describing practical challenges in the domain of KM, findings are also discussed and compared to relevant literature.

1.2 Problem definition

The tile of the thesis is: ” A study of practices for domain and architecture knowledge management in a Scandinavian software company”. The study aims at answering and discussing the research question: "How is domain- and architecture-knowledge managed in a Scandinavian software company?" The thesis emphasizes on describing the company, analyzing the findings and comparing them to relevant literature.

1.3 Company description

The company in which i conducted my interview is a Scandinavian software and hardware producer. They have about 70 employees. The company is a leader in its domain which is hardware and software for earth observation ground-stations. They have customers around the world, and cooperate closely with several large international companies. They focus on KM and have a separate QA department that advocates this. The company has a widespread product line culture where

software and architectures are re-used to a large extent. Production processes also focuses on storing modules for later use.

1.4 Thesis overview

In chapter **two** and **three**, a literature study is performed, the chapters consist of summaries of relevant literature that I have studied in connection to this thesis. The summaries are supposed to introduce the reader to the domain of the thesis. Chapter **four** describes the process of planning and performing the data gathering. It also contains a more detailed problem definition and validity evaluation. In chapter **five**, data from the interviews are discussed and analyzed, they are also compared to relevant literature. Chapter **six** contains a summary of the analyses, valuation of the project and a further work chapter. Articles that have been read are listed in the bibliography. The **appendix** contains interview guides, container topics and statements, evaluated research questions, Gantt diagram, project context and project evaluation.

The thesis was organized in this fashion in order to present the data in a well arranged way. To some extent it also mirrors the order in which things were done.

2 Knowledge management
Introduction to knowledge management and summaries of read material. In this chapter I wish to introduce the reader to the domain of knowledge management, especially in regard to software.

3 Software architecture knowledge management
Introduction to managing architectural knowledge. Here the term architecture is defined and discussed, then related to knowledge and management.

4 Research focus and method
Description of the process of collecting data. This chapter aims to guide the reader through the different aspects of the research and method.

5 Analysis and discussion of results
Results are discussed and analyzed. The views from both relevant literature and the author are also presented in the discussion.

6 Conclusions
This chapter concludes the thesis. The reader is presented with a summary of my findings and analyses.

Bibliography

Appendix

2 Knowledge management

This prestudy was conducted in order for me to obtain some background knowledge about KM. The following chapter contains brief summaries from some of the literature I have read. I have focused on KM as I believed that it was the most relevant for the interviews and the research question.

2.1 KM definitions

Although the word management is fairly straight forward, the word knowledge is open for many interpretations. According to the website [ABC of KM], all knowledge assets can be put in one of the two categories; explicit and tacit. Included among the former are assets like patents, trademarks business plans marketing research and customer lists. As a general rule of thumb, explicit knowledge consists of anything that can be documented, archived and codified, often with help of IT. The concept of tacit knowledge is much harder to grasp. This is the knowledge in peoples heads. The challenge with tacit knowledge is figuring out how to recognize, generate, share and manage it. Although IT in the form of email, groupware, instant messaging and related technologies can help facilitate the dissemination of tacit knowledge, identifying tacit knowledge in the first place is a major hurdle for most organizations.

This definition of knowledge was found in Merriam-Webster's dictionary:
"2 a (1) : the fact or condition of knowing something with familiarity gained through experience or association (2) : acquaintance with or understanding of a science, art, or technique b (1) : the fact or condition of being aware of something (2) : the range of one's information or understanding <answered to the best of my knowledge> c : the circumstance or condition of apprehending truth or fact through reasoning" [Webster].

However, there seems to be a general understanding of what KM does, as many definitions have similarities. Words like process, organization and information often occur. Here are two definitions on KM:

"...KM is the process through which organizations generate value from their intellectual and knowledge-based assets. Most often, generating value from such assets involves sharing them among employees, departments and even with other companies in an effort to devise the best practices" [ABC of KM].

"The systematic process of finding, Selecting, organizing, distilling and presenting information in a way that improves an employee's comprehension in a specific area of interest. Knowledge management helps an organization to gain insight and understanding from its own experience" [KM FAQ].

According to these definitions, KM is a process where a company tries to gather information, refine it to knowledge and store it for later use in order to increase efficiency and decrease expenses.

2.2 KM and technology

[ABC of KM] claims: *"It's important to note that the definition (of KM) says nothing about technology; while KM is often facilitated by IT, technology itself is not KM"*.

However, IT seems to have an accelerating effect on the KM processes. Tools like Instant Messaging (IM) programs (i.e. MSN, ICQ), e-mail and chat programs (i.e. mIRC), are fast and effective channels for knowledge sharing among the attending participants. These examples are small programs used in addition to the normal work routines. Large enterprise systems (i.e. SAP) pervading the company is another alternative; these systems often have additional functions like coding, storing and sharing the accumulated knowledge. However these functions might be more expensive in terms of both licensing and time-consume. There are also smaller program packages that are intended for small and medium sized companies. On the web page [ABC of KM], it is claimed that KM tools often fall into one of the following categories: Knowledge repositories, expertise access tools, e-learning applications, discussion and chat technologies, synchronous interaction tools, and search and data mining tools.

2.3 Why is KM valuable

Not all information is valuable, different companies have different views on the value of information, data and knowledge. Therefore each company has to determine what information qualifies as intellectual and knowledge-based assets. KM can give several advantages to a company, both direct bottom-line savings and others that are more difficult to quantify. After all, when creating software it is probable that a large percentage of the input comes from the coders head. If this input, i.e. lines of code, was to be properly packaged and stored, it could reduce costs if it were to be used once a similar program was created. An article by P. Glasser [Glasser], shows an example of how knowledge is extracted by employees, after they are fired. This was done by asking departing employees to write down tacit knowledge that they thought was important. In order to make the ex-employees dedicate themselves to this task, a board of peers reviewed the document and awarded prizes from $1000 to $5000. This can be a powerful asset as the employees can express themselves without risk. Of course the prizes given to the exiting employees should be modified accordingly to the managers wishes and the KM budget.

2.4 A view on the current status on KM

On the web-pages of AOK, the Association Of Knowledgework [AOK], there is an interview with Jerry Ash. Here the KM guru elaborates his views on the current KM scene. He claims that the term 'knowledge management' has become a contentious issue and that many KM leaders have voiced their wariness of the expression itself. However he does not think that knowledge management is the latest invention of gurus. He says: "*it reflects the demand growing out of the explosion of the internet and other forms of personal communication*". Yet the KM model is not easily integrated in business. Management support is critical, still he fears "*We've scared the pants of stockholders and*

senior executives" and blames the "*somewhat reckless talk about bottom-up control and decision-making*". He claims that the biggest challenge to integrating KM in a company is the "*us and them*" mentality of both management and workers. However it might not be wise to paint a too rosy picture of things to come. There is a worry that KM might be hijacked by other disciplines, as people realize that IT is not the driver of KM but simply an enabler.

2.5 KM requirements

In order to successfully accomplish a successful KM initiative, there are several precautions that should be taken. Perry Glasser claims that: "To succeed at effectively managing knowledge, companies have to incorporate knowledge management completely into the cultural fabric, a shift *that requires changes in both organizational and personal philosophy"* [Glasser]. Hereby he rules out any small additional programs and emphasizes that a large investment is necessary in order to successfully implement KM. However there are numerous examples of where KM has gone wrong, and it is common knowledge that the more you spend on a KM initiative that fails, the more you can loose.

Culture is also an important factor as individualism and entrepreneurial competition are entrenched in human traits. The article by Glasser also refers to a statement from Ernst & Young; 56 percent of executives identify “changing people's behavior" as the greatest single obstacle to knowledge management initiatives. Tacit knowledge can be the one thing that makes an employee wanted, convincing him to share this knowledge can be difficult as it robs him of his unique skills. The need to sell the KM concept to employees shouldn't be underestimated.

2.6 Return on investment (ROI) on KM

[Glasser] states that figuring a return for knowledge management initiatives, is like trying to calculate payback from providing employees with telephones, paper, and pens. A problem when calculating ROI is to find a proper way to evaluate the information. Employee participation in KM is to be rewarded, but the difficulties in assessing the worth of information can make this job challenging. The article claims that a simple "hit count" on stored information use reveals little about its worth. It is imperative to find out who is using the information, for example if it is decision makers. If an employee uses stored knowledge to complete a task, he could be asked to estimate time saved by using KM. However these estimates would be very subjective.
In this study I describe some of the KM initiatives in a company, however I was unable to collect data on ROI on their KM project. Therefore this subject will not be covered to a great extent in the thesis.

3 Software architecture knowledge management

Software architecture is a common term in software producing companies. It is often used to describe the underlying frame, to which the software modules are to be connected. An architecture can also be a diagram showing modules and connections like the one below.

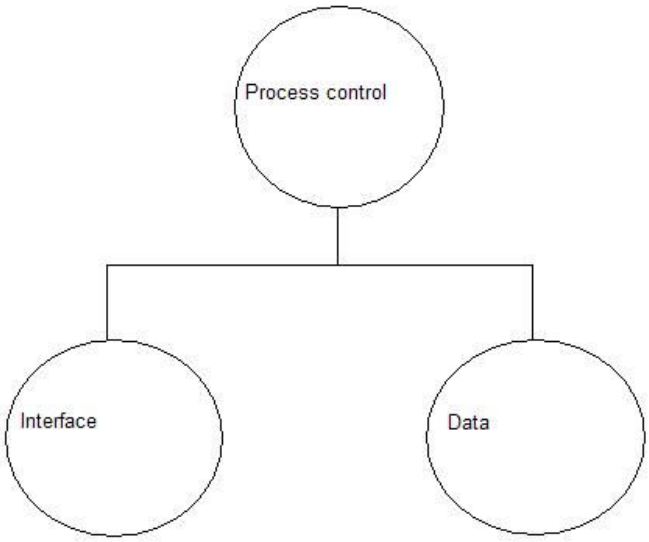

Figure 1: Example of an architecture

The figure shows a simple architecture, architectures can of course be very complicated. However architectures do not have to be modeled or drawn after certain templates. It is very possible to create a solid architecture by drawing on a whiteboard, photographing it with a digital camera, and print it out.

3.1 Definitions

In the book "Software Architecture in Practice", software architecture is defined as follows:

"*The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them*" [Bass et al, 2003].

This definition is discussed in the book "Empirical Studies of the Impact of Architectural Understanding on Software Evolution", where it is clamed that this definition does not try to list the concerns of a software architecture. That "components" and "relationships" are keywords. The definition does not define what a component is, nor which relationships

are considered. It is then concluded that the number of relationships between components is possibly unlimited.

"Therefore, this definition of software architecture cannot be used to distinguish what is not software architecture" [Bratthall, 2001].

For each architecture there are stakeholders, these are people or organizations that have interests or demands in the design of the architecture. Examples of stakeholders are customers, end users, developers and the project manager. Each stakeholder has individual concerns that might conflict with other stakeholders, and the architects have to take all views in regard and try to make a solution that everyone is reasonably happy with.

3.2 Architectures and knowledge

An architecture could be claimed to be knowledge, most likely explicit knowledge, as it can usually be printed and archived. However, as software architects become more experienced, I believe they attain tacit architectural knowledge. This is knowledge about architectures that might not easily be documented. Examples are patterns and essential parts of architectures that might occur in most architectures within a domain. Although some of this information could be stored by creating general architectural patterns, this might not be the smartest thing to do because of the costs. The granularity requirements of an architecture might differ on seemingly similar projects, but other factors like developer experience might have an influence. Therefore the experience and tacit knowledge of the architects must decide how detailed the architecture should be. I believe this knowledge is crucial in the design phase of the architecture and that it should not be underestimated.

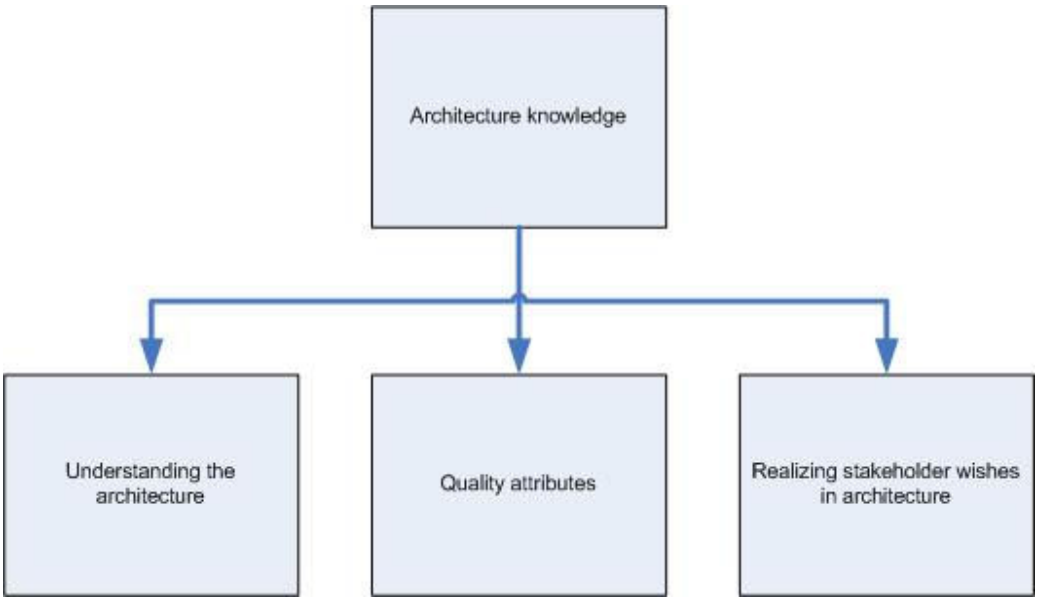

Figure 2: Architectural knowledge

The figure above shows how architectural knowledge can be broken down, thus transferring knowledge to the architects. This knowledge might be re-usable and could be lost if key employees leave or forget.

3.3 Software Product Lines (SPL) and re-use of software

Product Lines (PL) have been around for a very long time, almost as long as manufacturing itself. One example is Eli Whitney's interchangeable rifle parts in the early 1800s. He made it possible to build a product line of firearms that shared components [Whitney].

However, SPL is a relatively new idea. One definition is: "*A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way*" [Clemets, 2001, p. 5].

This definition has a lot in common with standard product line definitions. Yet it also puts constraints on how the systems in a software product line are developed. When the systems in a software product line are developed from a common set of assets in a prescribed way, substantial production economies can be achieved. This might not be the case when software is being developed separately, from scratch, or in an arbitrary fashion. The term "prescribed way" can be interpreted as an architecture, where all the modules fit and can communicate with each other.

As written in [Clements, 2001], many economic analyses of software productivity done by people like Barry Boehm, have identified software re-use as the biggest area of opportunity area for improving software productivity, quality and cycle time. This concept initially sounds very easy and attractive: just save all your components and reassemble them to make new applications.

However, this is far from easy. There are incredibly many ways that a software component can be incompatible with another. They can have different invocation conventions, units, dimensions, formats and so on. Still companies continue to assemble ad hoc collections of components in the vain hope that they will magically fit together. There might be solutions to this problem, one is to change the production process so that all components are designed to fit each other and use the same communication channels. In other words make an architecture and implement this in the different stages of the production process. This solution might not come without its own complications, as production times can go up when adding additional re-use functionality.

According to [Clements, 2001], a number of companies began using domain engineering and domain architecture, expecting to suddenly be able to use product lines efficiently. Instead they found a new obstacle, this one was a function of software economics. The book *Measuring Software Reuse* [Poulin, 1997] shows in a number of studies how reusable components cost about 50% to develop than one-off components. However the

company usually has a limited time to develop the software, if spending time making the software reusable, this can cause the deadline to be crossed. This is illustrated in the example below.

You may be pretty sure you can build a component in the six weeks allocated for developing it. If you're presented with a reusable component which requires some adaptation, you may assess it as having an 80 percent chance of needing only 2 weeks to adapt. But a 20 percent chance of needing 10 weeks to adapt. From an expected value standpoint, the expected time to adapt the component is (0.8)(2)+(0.2)(10)=3.6 weeks, which is a good deal better than the 6-week build-it-yourself potion. However, if the component is on the project's critical path, even a 20 percent chance of being responsible for a 4-week project slip will generally be enough to have you reject the reuse option. [Clements, 2001, p. xviii]

Although this might sound discouraging, there are ways to work around the challenges of SPL. Once a product line culture is in place the risk of overdue projects should decrease as the company becomes better in re-use and project resource estimates. As the component library continuously grows, more components will fit new projects and can be re-used. Below is an example of a company that successfully used software product lines and came out on top. This is interesting as it shows the power and possibilities that lie in a strong product line culture and mindset.

The company in question is CelsiusTech Systems AB from Sweden. They supply shipboard command and control systems to navies around the world. In 1985, CelsiusTech faced a monumental crisis: they had accepted to build two systems, larger than anything they had built before. And even on the smaller systems they had had problems meeting deadlines. They reasoned that the best solution would be to build one solution that would satisfy both customers as well as future customers. Surprisingly, CelsiusTech was able to deliver both systems and 50 subsequent contracts. By using software product lines CelsiusTech took years off delivery schedules, allowed a smaller staff to produce more systems, and sent their software reuse levels into the 90 percent range. [Clements, 2001]

In practice CelsiusTech managed to re-use code to a wide extent. On average, 70% to 80% of the elements were checked out of a library and inserted without code modification [Clements, 2001]. The company also had a very strong product line mindset, they considered the business to be the care, nurturing, and growth of its software product line, particularly on its core asset base.

4 Research focus and method

During two days in April 2005 I conducted eight interviews with employees in the respective company. Six software developers and two from management were interviewed. It was a descriptive and exploratory single case study. This chapter displays the focus and methods that were used to answer the research question: "How is domain- and architecture-knowledge managed in a Scandinavian software company?".

4.1 Research focus

The focus in this thesis is on practical management of domain and architectural knowledge. I have also studied how people work in comparison with the managers goals in these areas. In addition to general knowledge management I have studied how architectural knowledge is managed, and analyzed its role in a company with a strong re-use culture. My interviews also contain questions about the management of code and how this asset is treated when working on projects.

4.1.1 Detailed problem definition

This study has become quite descriptive and some of the focus changed from "community of practice" to "architecture knowledge management". I hoped to be able to observe relevant meetings, discussions and work processes during my visit to the company. However it became clear that I would have a small period of time to gather data, only two days. Therefore the data gathering process was limited to interviews only. I also started focusing on code management and how the management operated and their views on goals and communication.

Evaluated research questions can be found in appendix 3. Additional thoughts on choosing the research question are to be found in appendix 5.

Project title
A study of practices for domain and architecture knowledge management in a Scandinavian software company.

Research question
How is domain- and architecture-knowledge managed in a Scandinavian software company?

Sub questions
1. How does the company manage knowledge?
- How is new knowledge acquired?
- How is knowledge shared?
- How is knowledge stored?
- How is knowledge re-used?

I want to explore how the developers practice knowledge management on a daily basis and how this is done compared with the managers strategies.

2. What are the company's goals in knowledge management?

How do the managers relate to knowledge management?
How do the managers communicate and follow up their goals?

With this question I wish to map some of the management goals and how the management views them in action.

3. How is architecture knowledge treated in projects?

My interest in this topic is the knowledge and experiences that are created when dealing with architectures. As it is very common for software companies to use some form of design or architectures, I believe there are possibilities for saving time and resources by harvesting and re-using this knowledge.

4. How does the company manage code?

The purpose of this question is to describe how the company deals with one of its most important assets; code. I found this relevant, as data on this subject would help me describe the production culture in the company. Also there are often similarities between the management of code and explicit knowledge although the resources often don't seem to mirror this fact.

4.2 Data collection

4.2.1 Preparations

When visiting the company I did not have an interview schedule. After meeting some of the managers I was told who I could interview. I had already made interview guides with questions. I had the interview guide ready on my laptop when the interviews started and tried to write as much as possible down when the interviewee talked. In addition I had a voice recorder that I turned on before the interview started. The interviews usually lasted for about 20 minutes.

4.2.2 Interview design

Two types of interview guides (lists of questions to be asked) were created, one for the developers and one for the managers. This was done to be able to study the data from two different angles. The interview guides can be found in appendix 1.

The questions for the developers had two categories; knowledge management and architecture. The part about KM focused on acquiring new knowledge, storing

knowledge, sharing knowledge and re-use of knowledge. The intention of this focus was to gather information so the KM culture in the company could be described. There were also questions on how assets like code are managed. The other part of the interview asked questions about architecture. I asked about re-use of architecture, managing architecture knowledge and how people relate to the term architecture.

When interviewing managers I focused on company goals and communication as well as knowledge management. I wanted to capture the manager's view on KM and how it was realized in practice. The queries about the goals and documentation were done to enlighten and describe how the management works and what it intends. The communications with the employees was also queried about.

Some of the questions to both managers and developers were identical, this enabled me to directly compare the answers. Other questions were totally un-related, there were several reasons for this, one being the area of expertise of the interview object.

In addition to the questions described above, both interview guides shared the same warm-up questions, those are described below.

The warm-up questions were the following:

1. What is your name?
2. What is your title?
3. What project are you currently working on?
4. What is your job in that project?

There are several purposes with these questions, the first one acts as a natural opener in the conversation, it also gives a unique tag on that interview. This makes the later analyzing easier. The second question tags the interview object as a manager or developer. The last two questions give the interviewer background information and knowledge about any eventual bias that might appear in the rest of the answers.

The interview designs were re-evaluated after the first two interviews, the most important change was the use of the term 'knowledge'. I found out that using this term was very difficult because people have very different perceptions of what knowledge is, and many find it very hard to define in words. I ended up using a lot of time rephrasing questions and giving examples on what kind of answer I expected, only to realize that I might be leading the subject in a certain direction.

After discussing the word selection with my one of my supervisors, I chose to use the word experience instead of knowledge. Example: "storing experiences" instead of "storing knowledge". I realize that experience might not be a perfect synonym for knowledge but we felt that this change had to be made to improve the interview experience.

For the same reasons, the term "domain knowledge" had to be changed into "your area of expertise", there seems to be a gap between the language used in literature and the language used in practice in companies.

These changes might partially have compromised the interviews, as they deviate from the original questions based on the research question. However we felt this was necessary to improve the overall quality of the interviews. Data from the interviews can be found in appendix 2.

4.2.3 Execution

I started the interview by introducing myself. Then I asked permission to use the voice recorder. I gave an estimate on how long the interview would last and started on my warm-up questions.

4.3 Data analysis

4.3.1 Context selection

The interviews were conducted in the offices of the interviewees. The interviews queried about both current attitudes and what the interview object had done in the past. According to [Wohlin et al. 2000], the study can be classified as follows:

- Off-line.
  I did not attend the process of knowledge management; I only made queries and described it.

- Professional.
  The company in question is a fully professional and competitive organization.

- Real problem.
  Knowledge management is a very current and interesting issue.

- Specific.
  The research has been conducted on a specific company.

I hope some generalization will be justifiable as many software companies have cultural and structural similarities. There will however be differences among the different company cultures.

4.3.2 Data set reduction

Little data set reduction did occur, some interview answers were incomplete or withdrawn and have been removed.

4.3.3 Validity evaluation

This chapter addresses the validity of the results. The evaluation was considered during the planning phase of the project. This was done to ensure valid results. Without validity

evaluation, one might end up with results that are not valid for the population from which the data is drawn.

The validity threats taken into account are listed below and discussed in relation to the thesis. They were found in the book "Applications of case study research" by Robert K, Yin [Yin, 1984].

Construct validity
"Construct validity deals with the use of instruments and measures that accurately operationalize the constructs of interest in a study. Because most instruments and measures are not necessarily as accurate as desired, a common strategy is to use multiple measures of the same construct as part of the same study. This same strategy works well in designing case studies and has been defined as the use of multiple sources of evidence" [Yin, 1984, p. 39-40].

The instruments used to gather the data were a laptop and a voice recorder. None of the instruments store data flawlessly, with this in mind I decided to record the same information with both instruments. This was done as a security measure. If the voice recorder should malfunction I might loose all data, if I hadn't written it down. I used the laptop to write down key words from the interview object's responses. My typing skills did not allow me to perfectly write down the entire conversation and I was forced to use shortcuts and extract the essence of long sentences. The voice recorder was recording during the entire interview, however some technical problems prevented it from recording everything that was said in the first two interviews. After the interviews the audio data had to be transcribed to a document. This also affected the data as I converted the incoherent speech to meaningful and correct sentences. The last construct issue was when I translated the written data to English; this also had an effect on the validity as one cannot be sure that no information is lost in translation.

However I expect the resulting data to be reasonably accurate and I have tried to create written data that mirrors the audio to the greatest extent possible. The fact that this project deals with statements and not numerical data also minimizes the damage done when validity is compromised. Because in this case the message was of the essence, not the medium.

Internal and external validity
"Both internal validity and external validity were touched on in the discussion on the role of theory. You can achieve internal validity through the specification of the units of analysis, the development of a priori rival theories, and the collection and analysis of data to test these rivals. Similarly, you can achieve external validity through the specification of theoretical relationships, from which generalizations can then be made" [Yin, 1984, p. 39-40].

Internal validity

Selection
The units of analysis were the people I interviewed. They were selected by a manager in the company. The selections were based on the fact that I wanted to speak with both

managers and employees. The manager picked the interview objects based on my request and their availability. The fact that the interview objects were picked by the manager could have some effect on the responses. Although I clearly stated that anonymization would be performed on the interview data, some of the interviewees seemed a little nervous after making bold statements.

Maturation
The data was gathered in a short period of time, two days. This might have an effect on the responses in the interviews as the interview objects were working on one specific project during the entire duration of the data collection.

Mortality
This should have little influence on the data as interviews were only performed once and without anyone refusing to be interviewed.

External validity

Interaction of selection and treatment
I was not able to find out to what extent the subjects in the study are representative for the population in other commercial organizations.

Interaction of setting and treatment
This is not a toy problem; I use the same tools as professionals and have experienced advisors. However, as there is little risk involved for the interviewees, there is always the possibility that some aspects have been taken lightly.

Interaction of history and treatment
The interviews were performed in a short period of time, therefore the data might be affected by current issues.

Reliability
"Finally, you can achieve reliability through the use of formal case study protocols and the development of a case study database. The protocols especially help to ensure that the same procedures are followed in multiple cases or in a study conducted by multiple investigators. The case study database is a way of differentiating the evidence from the case study manuscript. These two components are readily differentiated in other methods – for example, a survey database as distinct from the survey report – but are often confused in traditional uses of the case study" [Yin, 1984, p. 39-40].

According to this definition, the reliability of the research is threatened. This is because the case study was not modeled after a certain model, but mostly the time and resources that were available. The same goes for the analysis that might have similarities to the grounded theory but does in no way follow it strictly. Additionally, no case study database has been created and none has been investigated on beforehand. Some of these threats might be countered by the neutral descriptive layout of the research process and the fact that the author has little or no personal gains by influencing the data.

4.3.4 Chosen strategy

Doing a KM experiment was not an option because I was to research a company while they were working. Thus the option of gathering employees and making them perform an experiment was not viable. Therefore it was necessary to do an empirical experiment where I disturbed the work routines of the employees as little as possible. This limited my options and I decided on focusing on interviews as my primary source of data. There was also little time available so the data gathering had to be completed in a short amount of time.

At first the method of choice was a customized version of the grounded theory [Grounded]. This was done by searching the interviews for any interesting statements and writing them down on a note. Then the notes could be tagged and sorted in groups by different criteria. The groups of statements could then be compared against each other and discussed.

In practice, making about 200 small notes and carefully sorting them out on a large floor was a very cumbersome job. Another problem was the sheer number of notes and the compact text on them, this made it very difficult to keep an overview. The questions also differed between and among the interview groups. This made the art of grouping the notes more difficult, it also required a lot of "on the go" interpretation of where a note belonged. I realized that this would probably be a lot easier if done on a computer using a word processor. Sorting the statements physically and grouping them according to the interview object did allow for some observations. For example the number of interesting statements given from each interview group.

Figure 3: ~200 notes with relevant statements, grouped by interview subject and job description.

I figured that I would try to use the essence of this method on a computer instead in order to increase efficiency. This was done by creating a word document and listing "container-topics" that were relevant to the collected data and research question. Then I read through all the transcribed material and selected interesting statements. These statements were then placed below the most fitting "container topic". This created groups of statements that were sorted by their contents. This method allowed for easy comparison of statements about the same subject. By tagging the statements with their origin, for example "Developer 1" or "Manager 2" I was able to investigate and compare statements by different criteria, one being the relationship between managers goals and developers daily work routines. This is in consensus with the overall research question.

There are several methods that can be used to analyze the qualitative data. When there are large quantities of data, it is common to feed the data to a computer and use a software package to deal with it. According to the book "Real world research" by Colin Robson [Robson, 2002], a word-processing package can do much to reduce the sheer tedium of qualitative analysis. I chose to not use a computer program to analyze the statements. The reason for this was that I didn't think it would save a lot of time. I also wanted to stay close to the data and not have to spend time learning how to use a new program.

5 Analysis and discussion of results

The following chapter contains analysis and discussion of the results, the discussion is based on the data from the interviews and angled towards the research questions created for this thesis.

5.1 Knowledge management

5.1.1 Acquirement of new knowledge

Gathering new knowledge is usually described as a very important task in companies that produce software. Although important, it might be so to a lesser degree in companies that have strong product line and re-use cultures. This is because new work methods and processes can interfere with the reusability of older assets. An example of this can be the usability of old code if a new programming language is to be used in upcoming projects.

When I asked questions about how people acquired new knowledge, 66% stated that they did so by working in projects. All the managers I interviewed said that one of the main sources of new knowledge should be working in projects. These numbers seem to show that the management has successfully forwarded some of the KM strategies to the employees. My impression on how the company gains knowledge from working on projects is illustrated in figure 4 on page 26. Overall I believe it is reasonable to say that the company in which I performed the interviews has a good culture of learning by working in projects. The strong product line culture ensures that older knowledge is refreshed and new knowledge is gained as the company uses its architectures on new projects.

According to the book "People-focused knowledge management":
"*A broad range of knowledge transfer activities are pursued to ascertain that valuable knowledge is captured, organized and structured, deployed widely, and used and leveraged*" [Wiig, 2004, p. 225].

These "knowledge transfer activities" occur frequently when employees participate in new projects. Not only are they brought up to date in their domain, they also get to repeat and "over-learn" what they already know. Additionally, new employees are "thrown right into it" and get to learn by doing, while surrounded by experienced people. However there are also pitfalls with this strategy, [Wiig, 2004] claims that when learning on the job, one mainly builds knowledge about "that is the way it is" and have limited understanding of the underlying mechanisms. This makes employees vulnerable when confronted with novel challenges.

Both managers and employees agree that little resources are spent on courses. This is done because for financial reasons. A manager also claimed that they had had some bad experiences with courses. People often forgot what they learned unless the knowledge was used immediately. In addition to the course expenses, the courses also seize manpower that could be used on commercial projects. However, one manager did say

that prior to each project the need for courses was evaluated. This indicates that the company hasn't totally discarded the option of sending employees on courses, it just happens less frequently. The employees are also responsible for keeping themselves updated on domain knowledge, although they aren't in any way explicitly rewarded for this. An employee stated that there were more courses previously, but they were removed because of the economy. I got the impression that acquiring new knowledge was a bit chaotic since employees are expected to stay updated on domain knowledge but haven't been allocated time to explore new knowledge. One person stated that "There is no time to sit down and study, those days are gone".

Yet, an article from IEEE Software [Rus, 2002] claims that organizations must quickly acquire knowledge about new technologies and master them. I am not sure whether it is enough to only rely on new projects as a source of new information. Especially in companies with a strong product line culture, because they tend to have a limited domain. Therefore the knowledge gained in projects is often "old news" as most of the processes and knowledge are re-used.

5.1.2 Sharing knowledge

66% of the interviewees answered that they usually ask colleagues when they need work-related help. It seems like this kind of information sharing works well in small and medium sized companies. By doing this, new employees learn from more experienced colleagues through informal meetings. The problem with relying on this strategy is that not everyone will have access to all the information they need. It also requires everyone to know what everyone else knows in order to ask the right person for assistance.

This is rarely the case unless the management has committed to create a list of who knows what, such a list will also come with its own challenges like updating and availability. Asking colleagues also becomes increasingly difficult as the number of employees grows because many colleagues might have some insight in the problem without being an expert. Therefore the help might not come from the one that knows the topic best. Physical distances in large companies also complicate ad-hoc sharing of knowledge, employees might have to spend a good amount of time to search for the expert. However, the hindrance of physical distances can be somewhat overcome by using phones and the Internet, but it might not yield the same result as a private conversation. Communication by web and phones are not covered in this thesis.

The company wherein I performed my interviews can be classified as a medium sized company with about 70 employees. A manager told me that they had: "put an effort into grouping employees offices based on their job descriptions". This was done to encourage and ease ad-hoc meetings and job-related conversations. One employee said that people weren't good at sharing what they did, so although the management has partially committed to improve the sharing of knowledge, I believe that the communication could be made more efficient by introducing new initiatives. Some of the interviewees also mentioned a thing called "Educational Friday", this was an initiative where one employee would have a 10 minute presentation of a recent project or something new, and those who were interested would go and listen. However these presentations had faded away lately

and several employees stated that they missed it. I believe this is a good initiative, but that it might need support from the management in order to survive.

In large or medium sized companies, formal knowledge sharing ought to be effective, perhaps in parallel with small-scale ad-hoc knowledge sharing. The article by [Rus, 2002] claims that "… formal knowledge capturing and sharing ensures that all employees can access it". Such formal knowledge sharing can be in many forms like in-house courses, presentations and papers. Although formal knowledge sharing can be expensive, it enables everyone who attends to receive the knowledge shared.

In order to advocate knowledge sharing, it is possible to create a reward system. The following quote has been taken from an IEEE paper by Iona Rus and Mikael Lindvall, it shows an example how employees can be rewarded for contributing in KM.

"Xerox recommends creating a "hall of fame" for those people whose contributions have solved real business problems. Xerox rewards staff that regularly share useful information and identifies them as key contributors to the program.
Bruce Karney, evangelist of a Hewlet-Packard KM initiative, gave out free Lotus Notes licenses and free airline miles to prospective users"[Rus, 2002, p. 34].

The article also mentions KM currency that can be converted into cash. Although these are examples of large projects in large companies, the principle is still interesting. Small and medium sized companies can also benefit from using some kind of rewards to encourage knowledge sharing. Possible rewards can be to go home from work early once in a while, or a bottle of wine. A prerequisite for this system will be a board consisting of managers or QA leaders that evaluate the knowledge sharing proposals that are presented.

Informal knowledge sharing can be effective in medium sized companies, however I believe that some effort should be taken in order to simplify the information sharing. One way to do this is to have an updated webpage that shows the domain of all the employees. Such a site will simplify the task of finding out who to ask when employees meet challenges. The trick will be to keep it updated with relevant information without having to spend too much resources on it. One suggestion can be to implement updates to this site in the project process. At some stage where the employees know what they will do in the current project they could be asked to update their profile. Simple KM tools might be created by anyone that has some knowledge in html and making web sites. It is also possible to buy competence management systems, for example SkillScape [SkillScape] and SkillView [SkillView].

Creating such a site will definitely need some resources. Keeping it up-to-date will also require an effort by the employees. Each company will have to weigh the pros against the cons, and I believe that the number of employees will be of significance when choosing whether or not to create such a solution. Small companies might not find it very useful, but as the company increases in size, such a system might be a welcome addition as it can promote and ease knowledge sharing and employee interaction.

5.1.3 Storing knowledge

Both of the managers that I interviewed agreed that the project leader was responsible for storing knowledge and experiences. (One also added that it was everyone's responsibility.) The company does have a project web where experiences and knowledge are stored. However 66% of the interview objects stated that the routines for storing knowledge should be improved.

Quote from one of the interviews: "*My experience is that it is especially important to store knowledge when there is a lot to do at work*". The interview objects back this statement up with several arguments. Like the fact that experiences are easy forgotten when there is much stress and focus on deadlines. Yet, especially in busy situations, storing knowledge is often not prioritized. This is justified by the fact that it doesn't immediately pay off. On the other hand, ignoring KM and focusing on getting the job done often seems a lot more sensible in the short run. However, in retrospect, mangers did admit that reserving time for KM does pay off. One employee told me that storing knowledge was "Dependent on employee initiative" and therefore it was often overlooked.

I believe that a possible solution to this issue can be to focus more on storing experiences, and to integrate KM-procedures in the project process. It might be claimed that this is already done, in meeting logs and FAQs (Frequently Asked Questions). However these documents have a habit of drowning the stored knowledge and make searches more difficult. Adding a dedicated experience and knowledge section in the project reports might store data in a fashion that is more accessible later.

5.1.4 Re-use of knowledge

Re-use of knowledge is often an unconscious act. Therefore it is often overlooked while resources are given to storing knowledge and managing stored knowledge. However, stored knowledge is of little use if it isn't re-used. When asking about re-use of knowledge I got the impression that this wasn't a much discussed topic. Statements like: "There is no conscious re-use politic" and "It is ad-hoc" were contradicted by statements like "There are procedures that are supposed to catch these experiences".
Several interview object stated that they did re-use knowledge and experiences when working on new projects, especially in the start phase. This was usually done by inspecting previous project documentation. However there was some concern that knowledge and experiences could be difficult to find and that it often drowned in other information. One employee stated that "What we need (and there might be one) is a project forum". This statement tells us that the re-use of knowledge could be easier, more importantly it displays the lack of communication in this area as the interviewee does not know if such a forum exists.

In my opinion re-use of knowledge can be divided into two categories; re-use of own knowledge and re-use of other peoples knowledge. Re-using own knowledge can be done by simply remembering things and perhaps writing them down. In order to re-use other peoples knowledge, there are many demands that must be fulfilled. The knowledge must be stored in appropriate places, coded in appropriate ways and be available when needed.

While these demands might require hardware and money to fulfill, the advantages of re-use has to be implemented in the employees heads and the company culture. The software company wherein I performed the interviews

5.2 Company goals in KM

The second half of my manager questionnaire dealt with the KM goals in the company. A QA manager told me that there was an "ongoing project where we work on a knowledge database and how to use this in the process of creating new projects".

A good knowledge strategy is of little use unless it's followed. Hence it is important to make sure that it is communicated throughout the entire organization. Both managers affirmed that the management agrees on the strategies, then they are forwarded to project leaders on meetings. Project leaders share the strategies and management directives with the rest of the employees. Changes and requests also come from below as shown in figure 4 on page 26. This strategy enables changes to be made in a slow and controlled manner.

I was told that "managers have a responsibility to ensure that strategies are followed". Measures are taken to ensure that the strategies are followed. A QA manager inspects the blast-off process to ensure that the project leader has used the encouraged strategy and experience data. This happens during the planning of the project. However I was told that there might be difficulties for a QA manager to get an overview, especially if his background differed from the domain of the project. This inspection allows the QA team to verify that the project follows the architecture and the guidelines set by the management.

Post mortem analyses are conducted after projects and the management also encourages employees to share suggestions. At the moment, the QA team has focused on negative experiences. These experiences have been systematized and eventual actions to prevent them from happening again have taken. By reviewing negative experiences from a previous project, one might avoid them in a future project. Utilizing positive experiences is also possible, for example by documenting a problem that can be solved in several ways, and then documenting a solution that was successful. One of the managers told me that this was an are they were working on.

I also asked about managing the knowledge of employees that quit. The company did not have any exit interviews. However the managers were open for formalizing a process where leaving employees can have an interview without any restraints. I believe exit-interviews are a cheap and easy method for gathering knowledge. Then the employee can speak from the gut without worrying too much about what the boss thinks. If the employee decides to have such an interview, he might do it because he wants to be polite or that he has so much respect for his ex-boss that he doesn't want to say no. In any case there might be little that motivates the ex-employee. Therefore some kind of reward might be fitting.

5.3 Architecture knowledge management

5.3.1 Defining architecture

When asking questions about architecture I was often asked what I meant by the term architecture. So after some interviews I started asking the interviewees about how they defined the word architecture. As expected, none of the answers were remotely similar. Architecture might be one of the most abstract words in computer science. Here are some of the definitions:

- Architecture is programming language.
- Architecture is how busses and chips are connected.
- I define architecture as a rough sketch, a functional picture of things, how functions are connected, not necessarily how they are realized. A group of building blocks.

The book "Software Architecture in Practice" defines architecture as:
"*The software architecture of a program or computing system is the structure or structures of the system, which comprise of several elements, the externally visible properties of those elements, and the relationships among them*" [Bass et al, p. 3, 2003].

One might think that communication must be hampered when "everyone" has their own definition of such an important word. However I don't see an easy solution to this issue, I believe the term architecture should have different meanings when used by different people and in different domains. Although there is the possibility of creating a standard where the word architecture has certain meanings in certain situations, I do not know whether this would ease communication or just make people use other words like design and layout. I did not get the impression that this was a problem in the company and that they had learned to live with the fact that the word architecture was used in many different contexts.

5.3.2 Architecture knowledge

There might be many similarities between managing knowledge and managing architecture knowledge. However I did get some unique answers when querying about architecture knowledge. The strong product line culture in the company almost makes it seem like new architectural knowledge is more or less redundant. A manager stated that "90% of the systems are based on the same architecture". Therefore new architectural knowledge is not given much priority. As people work on previously created architectures, much of the knowledge sharing happens by working on projects. Working on architectures usually involves modifying a previous architecture.

This process enables experiences and knowledge from previous architectures to be shared and re-used. I was also told that both negative and positive experiences are stored when working with architectures. These descriptions of more or less successful architectural implementations provide data on both what to do, and what not to do. Employees search

among previous projects to find similar architectures and solutions. However there did not seem to be a separate architectural experience library, and one employee did request a forum with experiences and knowledge to simplify searches for architectural knowledge. The basis for this statement was that there was “Often very much to look through in order to find re-usable assets”. When relating these statements to the research sub-question "How is architecture knowledge treated in projects?" it seems to me that that in the company where I performed the interviews, there is little formal focus on this part of knowledge management. I believe an improvement in the availability and re-usability of the architecture knowledge could make the production process more effective. A possible improvement could be to focus on storing architects tacit knowledge for later use, either in a separate architecture knowledge repository, or in the project documentation.

5.4 Asset management, code

Re-use of code is very important in every software company. It has several advantages and can save time and resources. The company in which I conducted my interviews is very interested in re-use of code. They have strict guidelines on how to create re-usable modules. Creating re-usable modules has gone from being a “smart thing” to a strong culture that pervades the entire company. All my interview objects claimed that they store and re-use code. A manager said that there has to be a very good reason for not making a module re-usable. As the company has a strong product line culture, it uses a standard frame and adds modules where necessary. They compare it to Lego [Lego], where modules are based on a common standard and easily can interact with each other. I was also told that a part of the company’s goal was to sell additional functionality. The company sells upgrades in forms of “easily” installable modules. This is a win-win situation, as the customers are “tied” to the company. On the other hand the customers are able to buy small modules based on need, instead of having to buy an entire new system.

The company has a library of re-usable code modules. Once a new project starts, the library is inspected for fitting modules. These might be usable with some or none modifications. This is the essence of the production process. Previously created modules are modified to fit the current project requirements.

When asking about the content of the code library I received the following response: *“Much* (code) *lies fallow* (in the library)*, the information is correct but not relevant any more”.*

Very little code is deleted from the library. This is because one never knows what code will be needed in the future. However this does cause some issues, the library will constantly have to be enlarged as no code is removed while more is added. Searches will take longer as there is more to search through. There is no formal evaluation of reusable components, see chapter “5.1.2 Sharing knowledge” for my thoughts about evaluating/rewarding KM achievements. One manager stated that “Project leader is first and foremost interested in finishing the project, not create re-usable assets”, and that this was an area where they can improve.

5.5 Interpretation of KM data-flow

The model on the following page was created to give an example of how a company might handle the data-flow related to KM. It also helped me to understand the underlying framework in the company. It is not an official model from the company wherein I performed the research. The model is based on my interpretations on the statements from both managers and employees.

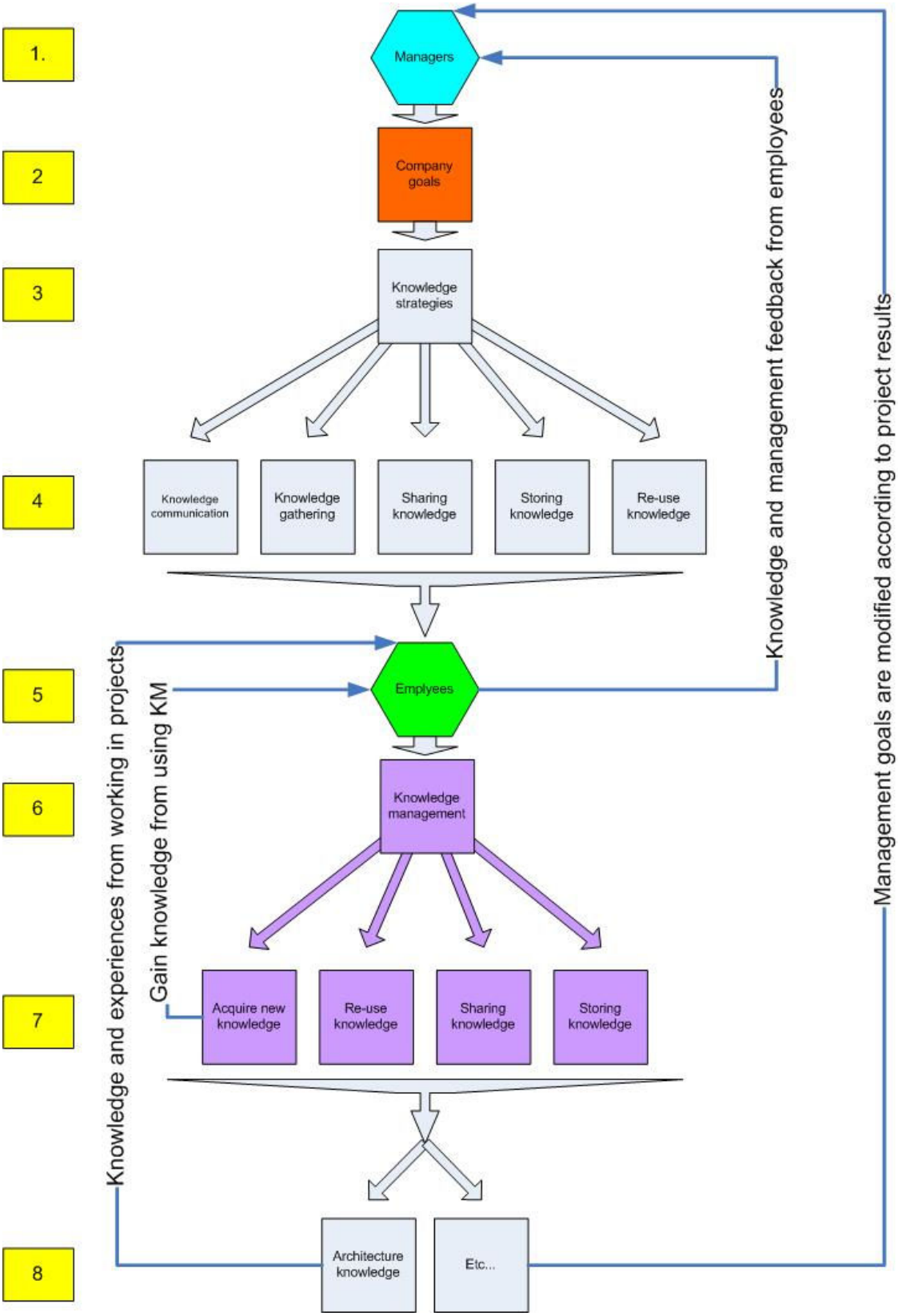

Figure 4: My impression of the flow of knowledge

1. "Managers" decide the direction of the company.

2. This results in "Company goals".

3. "Knowledge strategies" are created based on "Company goals".

4. "Knowledge strategies" contains strategies on several areas.

5. The strategies are forwarded to the "Employees".

6. "Employees" perform "Knowledge management".

7. "Knowledge management is performed on several stages of several processes.

8. "Knowledge management" is performed in many areas, for example "Architecture knowledge".

The model shows how the managers create knowledge strategies and forward them to the employees. It also illustrates how employees implement the strategies by performing knowledge management. This is done on several objects, architecture knowledge being one of them. Although there are similarities between knowledge management and system architecture management, there are also differences. The model illustrates this by having architectural knowledge management below knowledge management. This model mainly relates to the research sub-question "How does the company manage knowledge?".

6 Conclusions

This thesis set out to answer the research question: "How is domain- and architecture-knowledge managed in a Scandinavian software company?" This has been done by gathering data, processing the data and discussing the findings. Based on the data I have gathered, I believe that the company wherein I have performed my research could improve in several aspects of KM, most noticeably arranging for better communication, and management of knowledge and experiences. However I was told that this was an area they were working on. This chapter concludes the thesis, each section summarizes my findings on the research sub-questions. As this is a descriptive thesis, I figured that a conclusion consisting of summaries of my data, research and analysis would be proper.

The research question was divided into the following questions:
How does the company manage knowledge?
What are the company's goals in knowledge management?
How is architecture knowledge treated in projects?
How does the company manage code?

6.1 Knowledge management

The company in which I performed my research has taken several measures to ensure a good KM culture. My research data indicates that there is a clear connection between the managers intentions on how employees should acquire new knowledge, and how this is done in practice. This shows that the management has successfully forwarded the KM strategies to the employees. The company also seems content with the fact that little resources are used on courses. Although this works well at the moment, there are numerous articles that warn against not staying up to date on technology.

When queried about knowledge sharing, most of the interviewees answered that they preferred to ask colleagues when they needed help. The fact that this is done indicates that this is an effective way of quickly overcoming problems. There was uttered some concern about this routine as it required everyone to know what everyone else knew in order to ask questions. Apart from an outdated CV there where no sources of information about the employees backgrounds and knowledge. This might not be a necessity in a medium sized company, however it might improve communication and encourage employee interaction.

Several employees stated that there used to be organized knowledge-sharing sessions where an employee had a presentation. However this had not happened lately and many interviewees missed it. I believe a reason for this might be the fact that little encouragement was given from the management and perhaps some kind of reward system might be appropriate.

Although some actions were taken to store knowledge in the project web, several of the interview object acknowledged that this was an area where the company could improve. There were concerns about knowledge getting lost when working in projects. Several

employees stated that storing knowledge was not a priority when working on projects. This was especially true when they were busy. This can be expensive as valuable experiences might not be stored for later use, but forgotten.

Re-use of knowledge is very much used in software companies in general and the one wherein I performed my interviews. However there seems to be little focus on the actual process of re-using knowledge and former experiences. I did not recognize a formally described process where former projects were studied when starting on a new project. Although there seems to be a widespread re-use culture in the company, I believe there are several improvement possibilities, for example an experience forum, and dedicated experience-posts in the process guides.

6.2 Company goals with KM

The QA team did have KM strategies, however they didn't appear to be very explicit. None of the employees presented a list of KM guidelines. I was told that the management decided on KM strategies and forwarded them down the ranks. However several employees did claim that the company encouraged them to manage knowledge but this seems to have been communicated informally as some employees knew very little about KM guidelines. The QA team also performed post-mortem analyses and inspections in projects to ensure that they followed the KM guidelines. The company had no exit interviews, yet both managers that I interviewed expressed interest in this subject and acknowledged that it might yield valuable results.

6.3 Architecture knowledge management

When talking with the interview objects the word architecture quickly arose as a source of misunderstandings. There were several different interpretations of the term, and I suspected that this might cause issues in the communication. However I found no evidence that this was a big issue that needed addressing.

As both managers and employees told me that they did study former project reports prior to starting a new project, I get the impression that the company does recognize architectural knowledge as a real asset. However some improvement in this area might ease the flow of knowledge and increase efficiency and re-usability. This could be done by focusing more on formal store, capture and re-use processes integrated in the production process.

"As many as 90% of the systems are based on the same architecture" a manager estimated. This did raise some interesting issues as one might expect new architectural knowledge to be of little interest. This does contradict the belief that gathering new knowledge is a prerequisite to success[Rus, 2002]. However I suspect that this claim might not be as important for all companies. While software companies that work with software and hardware might need a stable development platform, companies that for

example limit their domain to web development are far more adaptable and open to make use of new standards and technologies.

6.4 Asset management, code

There is much focus on managing code. Every code module that is written has to follow standards and be re-usable. There are guidelines that help making modules that can communicate with previously created modules. A manager told me that making re-usable code segments does take more time than making disposable segments. However their experience was that having a library of re-usable modules was very efficient, especially with regard to the product line culture in the company. Yet this storing of data does not come for free, in addition to the initial increase in development time when creating a re-usable module, the code library has to be maintained and enlarged from time to time. But the managers believed that the benefits from having a number of reusable modules outweighed the costs by far. Also the overall development time was reduces when a new project could use several modules that had been created earlier and only needed modifications. A manager also told me that creating re-usable assets while working on projects was a subject they could improve in.

6.5 Valuation of project

My hope is that this thesis will be read by managers and employees in companies that are interested in KM and wish to learn about how it is done in other companies. Not only have I described crucial aspects of real life KM, but have made neutral comments and referred to relevant literature.

While KM is very well documented and an important issue in today's software industry, the management of architectural knowledge does not seem to have been given much attention. This thesis describes this area and debates its role in the larger KM initiatives.

6.6 Further work

A formal inspection of current KM initiatives in the company
I believe a thorough inspection of the KM initiatives in the company would yield interesting results. By mapping all the initiatives based on info from both managers and workers I believe the company would be made more attentive to the KM initiatives, it should also be possible to compare the initiatives and measure their success against each other. This would also give all the employees an overview of all the KM initiatives and they would be put into perspective. I believe this would increase the overall KM spirit.

Integrating KM in the work process
Most software companies have intricate documented processes that are to be followed when working on projects. By re-designing this process it should be possible to extract more knowledge and experiences from the projects. By altering the process and

documents to allow for separate knowledge and experience chapters, it will also be easier to search and extract knowledge from previous projects.

Architectural knowledge management
Architectural knowledge management does come with its own challenges and issues, I believe this area deserves a more thorough investigation to reveal the possibilities for saving resources in software companies.

Bibliography

Referenced books and articles in alphabetical order

[Bass et al, 2003] Len Bass, Paul Clements, Rick Kazman (2003) Software Architecture in Practice, Second Edition, Addison-Weseley, Pearson Education, Inc, Boston

[Clements, 2001] Paul Clements, Linda Northrop, Software product lines, Practices and patterns, Addison- Wesley, 2001

[Bratthall, 2001] Lars Bratthall (2001), Empirical Studies of the Impact of Architectural Understanding on Software Evolution, Facutly of Mathematics and Natural Sciences, University of Oslo

[Hansen et al, 2002] What's your strategy for managingknowledge, Morten T. Hansen, Nitin Nohria, Thomas Tierney, Harvard Business Review, March-April 1999

[Lawton, 2001] G. Lawton, "Knowledge Management: Ready for Prime Time?" Computer, vol 34, no. 2, Feb. 2001, p 12-14

[Poulin, 1997] Jeff Poulin, Measuring software reuse, Addison-Wesley, 1997

[Robson, 2002] Colin Robson (2002) Real world research, Second edition, Blackwell Publishing.

[Rus, 2002] Knowledge Management in Software Engineering, Iona rush and Mikael Lindvall, IEEE SOFTWARE May/June 2002, Frauenhofer Center for Experimental Software Engineering, Maryland

[Wiig, 2004] Karl Wiig (2004) People-focused knowledge management, Elsevier Butterworth-Heinemann, 200 Wheeler Road, Burlington, MA 01803 USA

[Yin, 1984] Robert K. Yin (1984) Applications of case study research, Volume 34, Sage publications, International Edcuational and Professional Publisher, Newbury Park London New Delhi.

Additional readings

Torgeir Dingsøyr, Nils B. Moe, Tore Dybå and Reidar Conradi, A workshop-oriented approach for defining electronic process guides

Judit Bell, 1999 Doing your research project, A guide to first-time researchers in education and social science, Third edition, Open University Press, Buckingham, Philadelphia.

Web pages in alphabetical order

[ABC of KM] http://www.cio.com/research/knowledge/edit/kmabcs.html
Accessed Feb 2005

[AOK] http://www.kwork.org/
Accessed Feb 2005

[Glasser] http://www.cio.com/archive/010199_know.html
Accessed Feb 2005

[Grounded] http://www.scu.edu.au/schools/gcm/ar/arp/grounded.html#a_gt_over
Accessed June 2005

[KM FAQ] http://www.mccombs.utexas.edu/kman/answers.htm
Accessed Feb 2005

[Lego] http://www.lego.dk
Accessed May 2005

[SkillScape] http://www.hrhub.com/storefronts/skillscape.html
Accessed June 2005

[SkillView] http://www.skillview.com/
Accessed June 2005

[Webster] http://www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=knowledge&x=0&y=0
Accessed Feb 2005

[Whitney] http://www.eliwhitney.org/
Accessed Feb 2005

Appendix

A1 Interview guides (Norwegian)

Intervjuguide developer

Oppvarmingsspørsmål

1. Hva er navnet ditt?
2. Hva er din tittel i bedriften?
3. Hvilket prosjekt jobber du med nå?
4. Hva er dine oppgaver der?

Utviklere/arkitekter

Spørsmål om kunnskapsforvaltning:

1. Hvordan anskaffer du deg ny kunnskap innen ditt fagfelt??
2. Hva synes du om lagring av komponenter og erfaringer for senere bruk?
3. Hvordan skjer lagring av erfaringer i praksis?
4. Hva er bedriftens syn på lagring av erfaringer? DVS fremgangsmåter og resultater. Har bedriften ønsker på områdete?
5. Synes du bedriften oppfordrer deg til lagring av kunnskap og erfaringer?Evt hvordan? Kunne bedriften gjort det mer attraktivt å ta initsiatitv til slik arbeid?
6. Hvilke faktorer har en innvirkning på hvor mye tid du bruker på lagring av kunnskap? FEKS: om man har det travelt? dersom du mener det er veldig viktig ? Hvor mye tid bruker du i praksis på dette?
7. Skjer dialoger om ditt fagfelt kun i prosjekter, eller finnes det andre arenaer?
8. Hvordan evalueres kunnskap og erfaringer for gjenbruk?
9. Når velger du å lagre en erfaring eller kunnskap for senere bruk?
10. I hvilke faser av et prosjekt evalueres erfaringer for gjenbruk?
11. eksempel på at dette har skjedd i praksis?

Spørsmål om arkitektur og KF:

1. Bruker du begrepet arkitektur?
2. Hvordan definerer du arkitektur?
3. Hvordan brukes tidligere kunnskap når nye arkitekturer lages?
4. Brukes det egene databaser for lagring av kunnskap og erfaringer?
5. Hvordan leter du etter tidligere erfaringer i arkitektur-design fasen? EKSEMPLER?
6. Hva synes du om gjenbruk av erfaringer i arkitekturen? Eksempler? Hva oppnår man med gjenbruk?
7. Hvordan lagres arkitektur-kunnskap når arkitekturer designes? Finnes det egne arkitektur dokumenter?
8. Hvordan kommuniseres arkitektur-kunnskapen internt?
9. På hvilke arenaer anskaffer du deg ny kunnskap om arkitektur?
10. Hva gjør du når du oppdager utdaterte komponenter som du er ansvarlig for?

Intervjuguide Manager

Oppvarmingsspørsmål

1. Hva er navnet ditt?
2. Hva er din tittel i bedriften?
3. Hvilket prosjekt jobber du med nå?
4. Hva er dine oppgaver der?

Kvalitetsavdeling/managers

Spørsmål om forvaltning av kunnskap:

1. Hvordan ønsker bedriften at ansatte skal tilegne seg ny kunnskap?
2. Hvordan blir ressurser til kursing for ny kunnskap prioritert?
3. Hvordan oppfordres ansatte til å lagre erfaringer og komponenter?
4. Hvordan lagres komponenter og hvilke krav er det til pakking?
5. I hvilken grad lagres erfaringer og kunnskap når man arbeider med prosjekter?
6. I hvilke perioder av prosjektene blir erfaringer og kunnskap lagret?
7. Hvem har ansvar for lagring av kunnskap og erfaringer?
8. Hvordan evalueres ansattes bidrag til gjenbruksdatabasen?
9. I hvilken grad forekommer gjenbruk i arkitektur-design fasen?
10. I hvilke faser av et prosjekt letes det etter gjenbrukbare arkitektur-erfaringer?
11. Kan du beskrive hvordan du leter etter gjenbrukbare erfaringer når du jobber med prosjekter?

Spørsmål om kommunikasjon og strategi:

1. Hvordan er bedriftens strategi for forvaltning av kunnskap og arkitektur?
2. Hvordan kommuniseres strategien til de ansatte?
3. Hvordan sjekkes det at strategien blir fulgt?
4. Hvordan legges det opp til at ansatte skal dele kunnskap og erfaringer?
5. Hva mener du om lagring av kunnskap til ansatte som slutter?
6. Hvordan er fokuset på gjenbruk av arkitektur og kunnskap i ditt nåværende prosjekt?

A2 Container topics and selected statements

Knowledge Management:

Acquire new knowledge
Learn from projects and web.
Multi-layered qa-function, everyone's responsibility.
On web, talk to colleagues.
Courses if needed.
Earlier we had allocated time to this, but has been removed lately cause of economy.
By working on projects, read and conversations.
By working on projects.
Not time to sit down and study things, those days are gone.
I'd might have gotten courses if I asked for it.
Largely internet, some in books.
Ask colleagues, learn while working, see what others have done in similar projects.
By working on new projects.
Method courses to prepare employees for specific tasks, often within project frames.
Ideally it might have been planned better.
Training and courses that functioned independent from projects, not possible today because of economy and much project work.
Today resources for courses are not prioritized.

Sharing knowledge
Free flow of knowledge. Often ask colleagues.
Need to know who knows what
Dialogues in project leader forums, information shared across projects.
People are not good at sharing what they do.
Educational Friday: Still exists, but happens very seldom.
Very interesting to learn what other people do.
Hardly know anything about what other colleagues do.
Depends on whether the presenter has time.
I miss it.
Much talking in the corridors.
Educational Friday, happens sometimes.
Interesting, usually participate.
3-4 times per year.
Would like it to happen more often. Then one could choose if one wanted to participate.
Atm one feels like one has to.
Send out information if I have done something new.
Kind of many unofficial places one learns things.
Have meetings but not much happens there.

Storing knowledge
Debrief, methods in projects catch experiences.
Happens after projects.
We do not have good routines for this.
Dependent on employee initiative.

The company urges employees to document experiences… I think.
No reward at all for storing/sharing knowledge.
Happens when I believe someone else can benefit from it.
Happens continuously in projects.
We try to store everything that is produced.
Company wants us to store and share experiences.
When an error has been made enough times.
Company makes storing knowledge attractive because employees learn from it and can work faster.
Workload is a deciding factor if I want to store knowledge.
My experience is that it is wise to store knowledge when workload is high though, cause then it's even easier to forget things.
Structure is important, good documenting routines.
Storing knowledge routines could have been better.
Company is interested in it. But when it comes to money…
The will to store knowledge could have been better.
Too much talk about costs, feels like one should already have harvested the fruits of new knowledge before learning it on courses.
Storing knowledge is based on personal commitment, not company encouragement.
It might have been possible to spend some time on storing knowledge when the workload isn't so high.
When in projects, no time is given to courses. Then there are time-limits and we must focus on getting it done.
Storing experiences is very smart, almost a must
Store knowledge by making html pages, put them on intranet.
The company thinks storing knowledge is important but does not have a very good system that does it. Many things are stored and disappear.
I guess the company encourages me to store knowledge. Sorry that there is no good system for it. The most important thing is that it is stored.
The most important factor on how much time I use to store knowledge is how easy the tool is to use.
I choose to store knowledge and experiences when I see things that should be stored for later use.
Info is stored in my head. The best is usually to talk to the right person.
Ask colleagues, learn who knows what.
I don't know what the company view on knowledge management is.
The company does not encourage me to store knowledge.
Don't know of any points on the list that evaluates experiences.
Knowledge and experiences is not evaluated.
Experiences on how to not do something are not stored.
A closure report is written on after projects, this is reviewed by a project-closure-board.
Meeting-logs are a source of knowledge documenting.
Experiences are stored during the entire project period.
An ongoing process.
Project manager is first and foremost responsible that experiences are stored.

Re-use of knowledge
No routines for evaluating knowledge/experiences for re-use.
Dependent on individual recognition of what can be reusable.
Create faq-databases with problems and how they were solved.
The idea is to store experiences, re-use them and share them with our customers.
Evaluates if the problem can reoccur then eventually document.
Over time, one learns what problems will reoccur.
There is no conscious re-use politic.
It is ad-hoc.
Experiences and knowledge is stored for re-use in the last stages of a project.
There are procedures that are supposed to catch these experiences.
Have databases but none that are meant for experiences and knowledge.

Asset management(Managing code)
Conveys relevant code to colleagues.
Perhaps once per project. Happens if there is time to do it.
We are very interested in re-use.
Try to make reusable code as much as possible.
Thinking ahead.
Have a common platform where object can be put together, almost like Lego.
Plug and play concept where objects can be put in without altering framework conditions.
Strategy is that customers shall buy additional functionality.
Code has become more modular, more and more re-use.
If I spot a module that clearly can be re-used it is stored in the library.
The library has many re-usable modules.
Code is modified in projects.
When I find outdated components I notify the responsible person.
Try to keep own components up-to-date.
Much that lie fallow, info is correct but not relevant any more. Do not delete anything myself.
Of course I have code that I re-use.
No evaluation of employee contributions to code database.
Making a re-usable component demands more than making one that isn't reuseable.
We have a re-use system, no formal evaluation of reusable components. Project leader is first and foremost interested in finishing project, not create reusable assets.
An area that we can work more with.

Architecture knowledge management:

Define architecture
Architecture is programming language.
Architecture is how busses and chips are connected.
Architecture is how things are built. Architectures are often inherited.
I define architecture as a rough sketch, a functional picture of things, how functions are connected, not necessarily how they are realized. A group of building blocks.

New architecture knowledge
Talk to new employees.

Company has a course-budget.
Courses have not been prioritized the latest years.
Dependent on qa to store experiences.
Get knowledge from web and mail from colleagues.
Believe the company can improve on this area.
I admit that I don't have any sources for getting new knowledge.
Acquire new knowledge on internet, searches.

Sharing architecture knowledge
Not much formal guidelines.
Asking colleagues often best option.
Mechanism for storing architecture knowledge would be nice.
Shared in halls and in meetings.
Have a db with peoples cv, though its not very thorough.
Mail, talk and presentations.
"Educational Friday" people are encouraged to do 10min presentations on current topics and selected solutions.
All architecture knowledge is available on intranet, few formal routines on sharing.
Every new thing that is made is published internally.
If one makes something one believes other can use, it is published.
"Project web" is used when new projects start, everything relevant is put there. Intranet.

Storing architecture knowledge
Has happened less often lately.
Routines for managing stored components could be better, have improved over the latest years though.
The amount of stored knowledge is very dependent on the project leader, his will and experience.

Re-use of architecture knowledge
Lesser extent of formally registered information.
Know what not to do.
During design we use other people's experiences.
Search for people who have done it before.
Often use previous knowledge and experiences.
Negative and positive experiences.
We use experience and previous knowledge.
Work practice shows what works and what doesn't.
If similarities to previous projects are identified, the previous projects are studied.
A set of building blocks ensures that the projects often are variations of the same base; there is not an indefinite sample space.
Re-use of experiences in the architecture hopefully increases efficiency.
My experience is that re-use does not impact the price that much, but it impacts the time needed to complete the project.
As new building blocks are added and are visible in the system, people are able to see relations between them and current problems.
When there is a database it is important that people see connections.
I search for similar projects, then I know how it has been done before.

There should be a forum for experiences and knowledge.
There is some activity on mailing lists.
Look things up if I don't remember. Either on internet or intranet.
Must re-use things in order to reach the goal as fast as possible.
Re-use helps projects to finish as fast as possible, doing things the way the have been done before.
Improve processes by seeing what worked and what didn't the last time. Find strengths and weaknesses.
Use what I have done before.
I make things that fit into a system that already has an architecture.
Often very much to look through in order to find reusable assets.
Must set aside time to look for reusable assets, don't know how efficient that is.
Don't necessarily trust that things are updated.
Not much re-use in architecture design phase. Not structured.
Finding and storing experience data can be difficult, hard to define a search, index words.
Very complicated to utilize experience data.
One can view a completed project and observe workload estimates, what was planned and what was needed. Then adjust current estimates.
As time passes more data is added, this increases the probability of finding usable experiences.
I use these methods, actually it's a demand.

Company goals

Create knowledge database. Then find out how to utilize it as good as possible.
This is an ongoing process. Hopefully detailed processes will be developed in time, these will be used when planning a project.
The strategy is communicated to the employees through project forum, there we discuss issues and transfer knowledge.
We ensure that the strategy is followed through the project leader forum, discuss and debate with project leaders. Encourage them to use experience data. Also through blast-off process.
Encourage project to make reusable modules. Good reasons must be given for a module to not be made reusable. Employees are not allowed to make specific code modules.
Strategy is to reuse.
Managers have a responsibility to ensure that strategies are followed.

Gather new knowledge
By working in projects.
Employees are encouraged to follow their domain.
All projects evaluate whether new knowledge is needed.
Annual employee conversations, review their knowledge.
Currently planning intern training. Will happen by having qa-team perform training within project management and project leading.
Believe that practical learning by doing is just as good as extern courses. Have bad experiences with that, often nothing new is learned unless talking about something specific. Unless people work with it afterwards it is often forgotten.

Sharing knowledge
Post mortem analysis. Concentrated on negative experiences and systematized them.
Found causes and eventual reactions.
Manages agree on the best way to do things, then pass on the guidelines to the lower ranks.
Information coming from below is also evaluated.
Employees are supposed to share knowledge by informal communication.
The ones that work in the same area are physically situated close.

Storing knowledge
No exit interviews. It could be smart to formalize such a thing tho.
Employees is encouraged to store changes in readme files.
Knowledge and experiences are stored through the whole project.
Project leader is responsible for storing relevant knowledge.
Also the individual, it is not governed by management.
We are trying to establish routines for gathering exit information from employees.
Code standard determines how to write code. Taking over modules created by these rules works fine.

Managing assets
All modules that are made and are reusable are stored.
Modules are not sold separately, they are the building blocks in our systems.

Re-use knowledge
Allways reuse in our architectures.
Very few systems where new architectures are designed.
90% of the systems are based on the same architecture.
Have developed templates on how a module is supposed to look.
This is physically documented in form of a code template.
After a while one has aquired some knowledge on what information is available.
First I check the module list, if any modules have similarities.
Ideally the entire system can be created for reusable components. Or modified modules.

A3 Evaluated research questions.

1 Knowledge Management in architecture/design:

1.01 To what extent is KM used in the architecture?
1.09 What assumptions on KM are made during creation of the architecture?
1.10 Do any processes manage/store/reuse knowledge?

1.02 How much resources are used/saved by using KM when projects are designed?
1.05 Will there be significant costs in implementing a KMinA solution?

1.03 Is there a gap between optimal practice and what really happens?
1.04 Is it possible to do something with these gaps? Is it wanted?

1.06 Does the company have an ad-hoc relationship to KM in the design process?
1.07 If yes, does this have advantages and disadvantages?
1.08 Can this in any way be improved or standardized?

Re-use of resources:

3.01 To what extent is Product Lines/re-use(of code) used?
3.04 Is there a system for reuse? Is it popular? Why, why not?
3.05 Are reuse-objects documented and evaluated?
3.06 Are employees encouraged/required to create objects for reuse?
3.07 Is reuse of objects and knowledge rewarded and evaluated?

3.02 To what degree is PL/re-use considered in the architecture/start-up-phase?
3.03 Is re-use and product lines a priority?

Knowledge Management during development:

2.01 Is there a continuous KM/reuse culture in the company?
2.02 Is there a budget for managing knowledge? If not has it been proposed? What was the result?

2.06 Has a large-scale KM product been considered?
2.07 Why? Why not? What was the response?

2.08 What is the general view on KM?
2.09 Has it been tried? With what result?

2.03 Are there any ad-hoc or non-formal KM processes?

2.04 How efficient are they? Are they promoted? How are they in practice?

2.05 Are there learning-loops in the different processes, constructed to share/record tacit knowledge?

General questions:

2.10 How is the term architecture used in the commercial world? Architecture = design?

2.11 What are the architects/developers daily concerns? Could KM solve any of them?

A 4 Gantt diagram

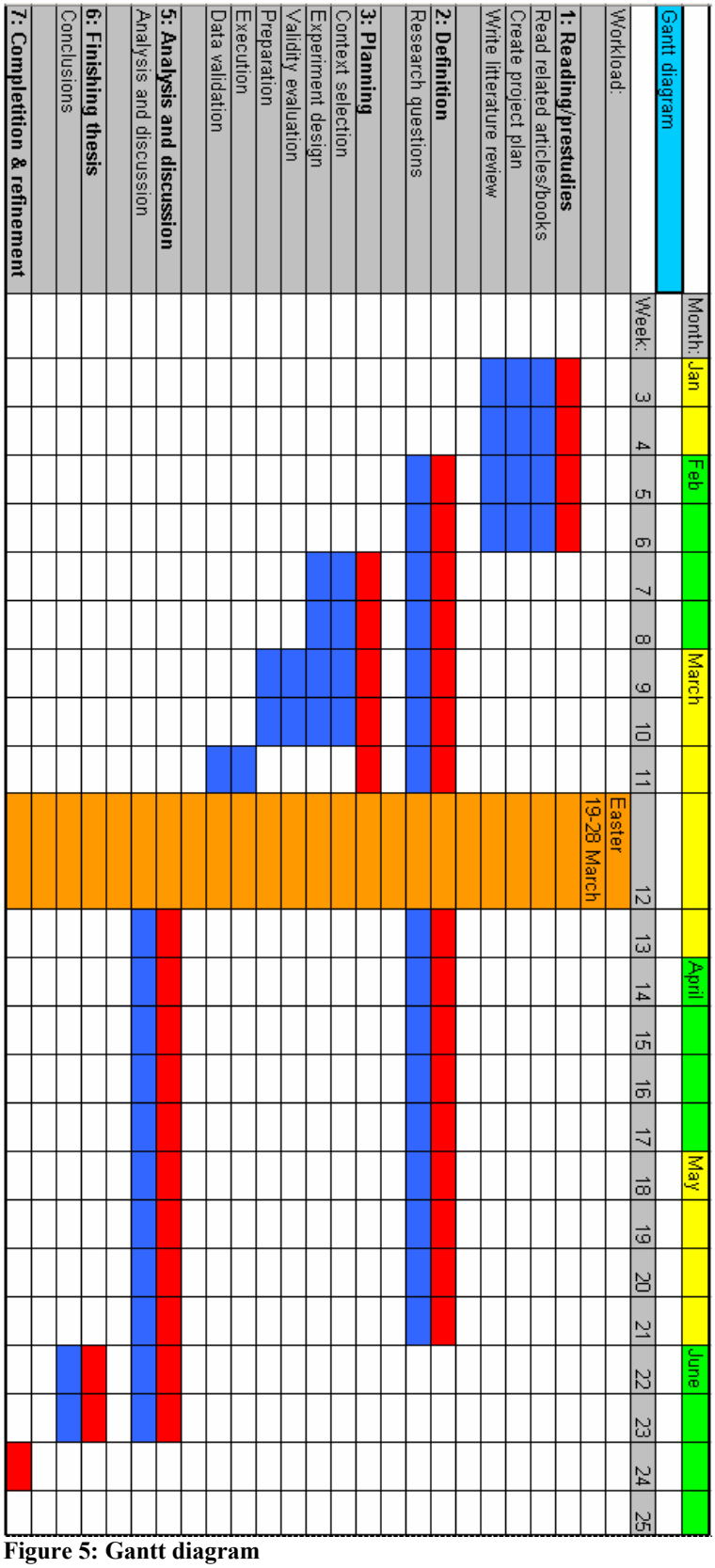

Figure 5: Gantt diagram

A 5 Project context

The thesis did not have a clear description in the beginning. The final description is a result of input from my tutors and me. The thesis is a part of the 10th semester in the masters program and it had to be completed within 20 weeks. The respective institute was the Department of Computer and Information Science at the Norwegian University of Science and technology. Some economic aid was given, the institute sponsored my trip to the company where I performed the interviews. I was assigned a semi-private booth with a computer. I was also offered the opportunity to work in an office in Sintef, but only when my tutors were there, thus the offer was not taken. Apart from doing a 13 week project last semester I have little experience doing this kind of research. A Gantt diagram that shows the work progress has been created and can be viewed in appendix 4.

I approached the challenge by first doing various studies of KM, PL and architecture. Then I created an interview guide that contained questions. These were formulated so they would enlighten the research question. When the interviews had been done the data was processed and compared against each other. There was also an ongoing literature study during the entire project.

When the thesis work started I did not have a specified research question. When the company in question was asked if they had any preferences about my research question I received the impression that they didn't. At first this sounded like a relief as I would not have a research question pushed over my head. However, I soon realized that creating a good research question without any prerequisites was very difficult. This resulted in a long process with much mails going back and forth between my supervisors and I.

A 6 Project evaluation

The thesis is aimed at people that are interested in KM initiatives and wish to se it described in a medium sized software company. Writing this thesis has an interesting task as I have been able to describe the process of KM in a real software company.

However I wish to state that my descriptions and conclusions might not be representative for the entire company because I have only interviewed a fraction of the employees. There are also numerous threats to the validity of the data and my interpretations. I had no personal interest in creating a glorious image of the company, or focusing on negative issues and problems, and have tried to view everything from several sides. Therefore I have presented an objective description of routines and their pros and cons.

My method of analysis not specifically documented in any literature but has been modified to fit the data I gathered. But I do believe that the nature of my data might prefer a discussion and reasoning, rather than creating statistics and graphs. This should advocate readability and understandability.

Generalization and relevance to other companies must also be done with caution. However I believe that companies with similarities in structure, culture, size or KM initiative, might discover that they have similar challenges or solutions. Then they can compare their results with this thesis and evaluate the outcomes.

I have had three very skilled and experienced supervisors when working with this thesis. Letizia, Tor Erlend and Torgeir have guided me when I have had difficulties, and helped me when I needed advice about KM, architectures and other domains.

I believe I have given an informative description about the daily management of knowledge, domain and architectures, and thereby fulfilling the research question. I am very satisfied with the results of this master thesis and hope it will be of help for software companies in yeas to come.