

*Master Thesis*  
*Software Engineering*  
*Thesis no: MSE-2006:01*  
*January 2006*



# **Software Process Improvements in a Small Organisation**

**an Ethnography**

**Sebastian Stein**

School of Engineering  
Blekinge Institute of Technology  
Box 520  
SE - 372 25 Ronneby  
Sweden

This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

**Contact Information:**

Author(s):

Sebastian Stein

Address: Hermannstr. 24, 06108 Halle, Germany

E-mail: seb.stein@hpfsc.de

University advisor(s):

Kari Rönkkö

School of Engineering, BTH

School of Engineering  
Blekinge Institute of Technology  
Box 520  
SE - 372 25 Ronneby  
Sweden

Internet : [www.bth.se/tek](http://www.bth.se/tek)  
Phone : +46 457 38 50 00  
Fax : +46 457 271 25

# ABSTRACT

Software process improvements are required to increase the productivity of software companies. Generally, it is the aim to increase the quality of the produced software and to keep budget and time. Quality models for software process improvements were developed in context of large organisations and multi-national companies. In this study I investigated how software process improvements are done in a small software company. Ethnography was used as research method. It was the aim of this study to build up an understanding of how software process improvements are done and enabled in a small organisation.

Fieldnotes were taken and later analysed using template analysis. Ethnography as the chosen research strategy proved to be applicable and feasible in software engineering research. The qualitative research strategy resulted in a detailed description of how one software company did software process improvements from a bottom-up perspective. Despite the learning potential of “how real world contingencies and possibilities interact and shape software process improvement efforts”, such descriptions are rare in software engineering literature.

Based on the field experiences and the analysed fieldnotes, the following results were identified: In the studied small software organisation, software process improvement efforts were pushed by the initiative of single employees. The studied company did not have enough resources to implement a complete quality model. In addition, management was heavily involved in daily work and therefore had not enough time to initiate and lead software process improvement efforts. For small software companies in a similar situation, the following guidelines can be given: First, a bottom-up approach with delegating responsibility from management to selected employees is needed. Second, management must ensure to be available if decisions must be taken. Third, improvements must be visible and feedback must be provided contemporary to gain momentum in the whole improvement effort. In some cases it might be important to create awareness of possible improvements. Here, employees should create internal lobbies by involving and convincing other employees of the improvement’s importance. A joined effort will help to create enough pressure for change, so that improvement efforts get started.

**Keywords:** Quality System, Ethnography, Software Methodology, Process Improvement

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>2</b>  |
| 1.1      | Outline of this Thesis . . . . .                                 | 3         |
| 1.2      | The First Day . . . . .  | 4         |
| <b>2</b> | <b>Company and Study Context</b>                                 | <b>5</b>  |
| 2.1      | Introduction . . . . .   | 5         |
| 2.2      | The Company's History . . . . .                                  | 5         |
| 2.3      | Formal Organisation . . . . .                                    | 6         |
| 2.4      | Market and Products . . . . .                                    | 9         |
| 2.4.1    | Production Planning and Control System (PPS) . . . . .           | 9         |
| 2.4.2    | Supply Chain Management (SCM) . . . . .                          | 11        |
| 2.4.3    | Demand Planning Software (RETAIL) . . . . .                      | 11        |
| 2.5      | Informal Organisation . . . . .                                  | 12        |
| 2.6      | And still Two Companies . . . . .                                | 14        |
| 2.7      | Project Leader Day Meetings . . . . .                            | 15        |
| 2.8      | The International Part . . . . .                                 | 17        |
| 2.9      | My Previous Five Months Practical . . . . .                      | 17        |
| 2.10     | Locating Myself . . . . .  | 18        |
| <b>3</b> | <b>Ethnography</b>   | <b>19</b> |
| 3.1      | Ethnography in General . . . . .                                 | 19        |
| 3.1.1    | Elements of Ethnography . . . . .                                | 20        |
| 3.2      | Quantitative and Qualitative Research . . . . .                  | 21        |
| 3.3      | Ethnography in Software Engineering and Related Fields . . . . . | 22        |
| 3.4      | Related Ethnographic Studies . . . . .                           | 24        |
| 3.5      | Doing Ethnography in the Own Domain . . . . .                    | 25        |
| 3.6      | Own Influence . . . . .  | 25        |
| 3.7      | Writing Ethnography . . . . .                                    | 26        |
| 3.8      | Ethnography Applied . . . . .                                    | 29        |
| 3.8.1    | Ethnographic Question . . . . .                                  | 29        |
| 3.8.2    | Accessing the Field . . . . .                                    | 30        |
| 3.8.3    | The Fieldwork . . . . .  | 31        |
| 3.8.4    | Collecting Fieldnotes . . . . .                                  | 32        |
| 3.8.5    | Data Analysis . . . . .  | 33        |
| 3.8.6    | Self Reflection . . . . .  | 35        |
| <b>4</b> | <b>Software Process Improvements</b>                             | <b>37</b> |
| 4.1      | Theory . . . . .   | 37        |
| 4.2      | Continuous Improvements . . . . .                                | 38        |
| 4.3      | Quality Models . . . . .   | 38        |
| 4.3.1    | ISO 9000:2000 and TickIT . . . . .                               | 38        |

|          |  |           |
|----------|--|-----------|
| 4.3.2    | Capability Maturity Model Integrated (CMMI) . . . . .                | 39        |
| 4.4      | Improvement Strategies . . . . .                                     | 41        |
| 4.5      | Success Factors . . . . .  | 42        |
| 4.6      | Summary . . . . .  | 43        |
| <b>5</b> | <b>Software Process Improvements Applied</b>                         | <b>44</b> |
| 5.1      | Top-down Approach and Careful Analysis . . . . .                     | 44        |
| 5.1.1    | Specification Improvement Effort Revisited . . . . .                 | 47        |
| 5.2      | Improvement Pushed by a Single Employee . . . . .                    | 48        |
| 5.2.1    | Enterprise Resource Planning System Introduction Revisited . . . . . | 50        |
| 5.3      | Bottom-Up Approach . . . . .   | 50        |
| 5.3.1    | Improving Knowledge Sharing Revisited . . . . .                      | 51        |
| 5.4      | Lobbying and Bottom-Up Approach . . . . .                            | 51        |
| 5.4.1    | Document Management System Introduction Revisited . . . . .          | 55        |
| 5.5      | Getting Out . . . . .  | 56        |
| <b>6</b> | <b>Conclusions</b>   | <b>57</b> |
| 6.1      | Improvement Strategy . . . . .                                       | 57        |
| 6.2      | Success Factors . . . . .  | 58        |
| 6.3      | Ethnography in Software Engineering Research . . . . .               | 60        |
|          | <b>List of Figures</b>   | <b>62</b> |
|          | <b>List of Abbreviations</b>   | <b>63</b> |
|          | <b>Bibliography</b>  | <b>64</b> |

# Chapter 1

## Introduction

In 2004 the national German economy grew by 1.7% according to the official statistics provided by the German federal office for statistics [Sta05]. In contrast, the whole information technology and communication economy grew by about 3.4% and software and information technology services as a sub-part of it grew even about 5.5% [BIT05b]. This shows that the software economy is growing much faster than other parts of the German economy. Today the whole German information technology and communication economy already contributes 6% to the German gross domestic product [BIT05a].

Such a fast growing economic domain challenges the companies working in this particular domain a lot. For example, companies have to expand quickly, which means to integrate new employees, release new products and adapt to changing environments and market conditions. In addition, software companies have to maintain software products and systems developed before, so not all resources can be used for new developments. According to that, it is very important to increase productivity, strictly speaking: the produced business value per employee must be increased.

A possible solution is to define, manage, and optimise the software production processes. This is generally the aim of software process improvements, which are discussed in detail in chapter 4 starting on page 37. Software process improvements can be done by following a so called quality model like TickIT [Ins50] or the Capability Maturity Model Integrated (CMMI) [Tea02b]. Such models define which criteria must be fulfilled by a good software process. They also include mechanisms to improve the used processes continuously. Beside implementing a quality model, there are also different success factors to enable this implementation. Typical success factors are for example management commitment, user involvement, and feedback.

Unfortunately, there are only few studies investigating the application of quality models in small companies [NC00, WMM05, see e. g.]. Some authors like Conigliaro [Con01] state that those improvement methods might be even harmful for small organisations. Also it is not clear if the success factors are applicable in small software companies. Therefore, this study focuses on a small software company doing software process improvements. By using ethnography it is possible to experience the improvement process with its entire complexity.

According to statistics provided by the German research institute for medium-sized businesses IfM<sup>1</sup>, 99.7% of all enterprises in Germany were medium-sized in the year 2003 [Ins03]. Medium-sized is defined by the IfM as having between 10 and 499 employees and an annual turnover between 1 million and 50 million Euro. Those medium-sized enterprises employed in the year 2003 70.2% of all employees in Germany [Ins03]. This shows, medium-sized companies are the heart of the German economy. Of course many internationally known IT companies exist in Germany as well, but taking a look at the local yellow pages reveals, there are much more small ones. There are many system vendors providing IT services, there are many web design

---

<sup>1</sup>Institut für Mittelstandsforschung Bonn: <http://www.ifm-bonn.org/>

companies, there are many companies offering specialised IT packages for production industry, etc. This means, the majority of IT companies does not have several thousands of employees. Consequently, it is interesting for research to study medium-sized companies.

Therefore, it is the aim of this study to investigate how software process improvements are done in a medium-sized German software company. As research method I chose ethnography, described in detail in chapter 3 starting on page 19, because using ethnography enables the researcher to acquire an in-depth insight of how an organisation works. In contrast, it is not the aim of this study to provide a general overview of how software process improvements are done in German medium-sized companies.

In this context it was a great luck for me that I had access to a German medium-sized software company already. Two years ago, I had done a practical at the Company, which will be described in detail in chapter 2 starting on page 5. I already knew that there was a great need at the Company to improve their software processes, because the market they were operating in was changing quickly and the economic situation was challenging.

The reader might ask why I had chosen to study the Company, because there are so many companies to choose from. Of course one obvious reason is that I already had access to the Company. This access saved me several months of fieldwork, because otherwise more time would have been needed to get accepted in the Company. Still, it is not the only reason, as the university is cooperating with many other companies and it would have been possible to get access there as well. Another important reason is the fact that the Company fits very well in the definition of a medium-sized company. There are about 45 employees working at two company locations and the annual turnover is with about 2 million Euro in the year 2004 between 1 million and 499 million Euro per year. Therefore, this company is a perfect place to fulfil the research aim of this study.

In the conclusions chapter it will be shown that the studied software company applied a completely different improvement strategy as suggested in literature. This strategy will be called a bottom-up strategy. Also the importance of the success factors was different compared to literature in the studied company. Those results provide a different view on how software process improvements can be done in a small organisation. Concrete guidelines are given so that software engineers can enable and initiate software process improvements in their own small organisation.

## **1.1 Outline of this Thesis**

In the next subsection I am presenting my experiences of the first working day at the Company. This should provide the reader with a first impression of the studied company. I worked six months in the Company to study their software process improvement efforts in extend. I participated in those efforts. I tried to apply some of the software process improvement theories mentioned in literature like success factors [BHR03, GH95, e. g.], introduction strategies [NC00, e. g.], and quality models [Tea02b, Ins50, e. g.]. In this thesis I am going to report and reflect on the issues I have experienced.

As mentioned above, I used ethnography as my research method. As it is not a very common approach within software engineering, I studied in detail the advantages but also the limitations of ethnography. This also includes a discussion of ethnography's applicability within software engineering research.

All those different parts of my study must be discussed in detail in this thesis. There are many ways to present such a study. For example, it is possible to discuss theoretical and philosophical issues together with the field experience. However, the disadvantage is that discussions get repeated and that not the whole picture can be told at one point. The discussion is separated over several different places.

To overcome this problem, I decided to structure my thesis as follows. First I introduce the Company in chapter 2. This is not a short introduction, because all the experiences presented in

this thesis are a product of working at this company. This means, one can only understand the thesis and experiences if there is a understanding of this particular company, its products and the products' market.

After introducing the Company, I discuss my research method ethnography in chapter 3. Beside describing ethnography in detail, I enclose a discussion why this research method is applicable in software engineering research.

Before I present the software process improvement efforts of the studied company, I am introducing the topic in chapter 4 about software process improvements in general. This introduction is later combined with the actual field experiences presented in chapter 5. I reflect on those experiences in context of software process improvements and I discuss whether the theories and models are applicable in a medium-sized software company.

## **1.2 The First Day**

It is a very sunny spring day at the beginning of April 2005. The sun is shining and there are almost no clouds. I am entering the new office building and I am surprised. When I was last visiting the Company in 2003 the office was in a city quarter mainly known for it's drug dealers. Now everything has changed. At the roof of the new building's entrance hall is a very nice wall painting done by a well-known local artist. The office building is modern, it was built just 10 years ago. There is a butcher and a snack shop in the ground floor next to the street. In front of the office building is a tram station and on the other side of the street is a big shopping centre. The building is located a 5 minutes walk away from the central market square of Halle. It is a busy place. Many people are strolling around for shopping or are just waiting for the next tram.

I walk upstairs and do not use the elevator. The Company is located at the second floor. There are 2 companies on this floor, the other one is an administration department of a medical institution. Before entering the office I have to get through a big glass door. The Company's name is written on the glass. Next to the door is a doorbell. I ring the bell and I can hear the ring tone through the door. After some seconds the door opens and I step inside. There is a long dark corridor in front of me. At the end of the corridor I can see the sun shining through a window and there seems to be a copy machine in front of the window. I decide to walk down the corridor. Almost at the end of the corridor a woman is looking at me out of her office. Immediately, I remember her, it is the Company's secretary. She also recognises and welcomes me. I go towards her.

One of the Company's bosses is in her office. He also welcomes me and we step inside his office, which is directly connected to the secretary's office. It is a very nice bright office. In the middle of the room stands a big desk with a computer. There are also some chairs next to the desk, so that a small group of people may have a small meeting in here.

Immediately, the boss starts talking about how much the Company has changed since I left it 2 years ago. The Company is now part of a bigger international corporation. They now have several new products, new employees and there is a breeze of international big business. I am overwhelmed by all those information and news. I try to explain the boss what I am going to do the next months at the company. He tells me that my adviser won't be available in the next days, so we agree that I come back at the end of the week.

While leaving the office I have a small talk to some of the employees. I already know most of them, even though some of them do not recognise me in the first place. They ask me how long I will be working with them and I say that I will be around at least for the next 4 months.

I leave the office building behind. I am impressed how much the appearance of the Company has changed since I left it. I am happy that I was able to talk to the boss and that so many employees recognised me. I have a feeling of familiarity.



## Chapter 2

# Company and Study Context

### 2.1 Introduction

You might wonder why I just say “the Company” and not the real name of the Company. I asked the Company if they would allow me to use their real name and they agreed. However, I thought about this issue several times and came to the conclusion that it is better not to have the real name in this document. I am going to present a detailed description of the Company, of their products, of their efforts to improve their way of working but also of their internal problems. If I included the full name in this document, someone searching the Internet for the company name might find this document and just read about the problematic parts. He might judge the Company in a wrong way and I want to prevent this. Today, the Company is listed on several stock exchanges and a bad report might not be good for future business. That are the main reasons why I am not including the Company’s name in this thesis. If you are really interested in the name, you can find it out by all the different information I present here in this thesis.

### 2.2 The Company’s History

The Company was formed by a fusion of two independent companies at the end of the 1990th. Today the Company still has both locations of the former independent companies - one big development unit in Chemnitz and the headquarter in Halle, both in Germany.

The former company in Chemnitz was founded at the beginning of the 1990th. They developed and marketed a production planning and control system. The former company in Halle was also founded at this time. They sold a third-party production planning and control systems for the fashion production industry. Later they also marketed the product of the company in Chemnitz.

Both cities are almost equally in size with about 230.000 inhabitants. Because of the major economic and social changes after the German Unification many local manufacturers had to quit business. There was not a strong local customer base available. Both cities are in the eastern part of Germany about 150 km south of Berlin. To go by car from Halle to Chemnitz one has to drive about two hours. Both cities are not very typical for such a company, because most of the German software companies are located in the western part of Germany in cities like Frankfurt, Munich or Stuttgart.

It was not easy to found a company at the beginning of the 1990th in the eastern part of Germany. At this time the former German Democratic Republic (GDR) joined the Federal German Republic. That time is today known as the German Unification. The eastern part of Germany was not that advanced at this time. For example, the public infrastructure like streets and telecommunication was not very good. It was almost impossible to rent offices and also telecommunication was not available everywhere. In addition, many big companies collapsed and many people were laid off. People from eastern Germany did not know the political, administrative, and eco-

nomical system of the Federal German Republic very well. Basically, the people had to relearn everything they knew about business and society.

The boss of the Chemnitz company told me that many people thought he was crazy to found a company in this context and at this time. However, at last he was right, because after a short time they had more than 10 employees and business was going well. The boss and founder of the Halle company told me that they had to learn everything about business by doing and experiencing. None of them had a special business education. In fact, the boss proudly showed me an electronical circuit board he had done on his own at the end of the 1980th. He pointed out that he had turned from a programmer to a salesman over the time.

The employees and bosses of both companies had to learn the way business works by experiencing. Beside many good lessons, there were also bad lessons to learn. Some contractors and customers misused the situation and the Company signed contracts which later turned out to be ineffective. They lost a lot of money, money which would have been needed to develop the companies and products.

Both companies were already working together to market the production planning and control system developed by the former company from Chemnitz. It was obvious that both companies had to unite into one. One important point was to increase the company size. Normally, customers are afraid to buy an important software package from a small company, because a small company might disappear quickly and the software package gets unmaintained. Under those circumstances both companies were bought by a German semiconductor manufacturer, who was using the production planning and control system. The newly formed Company was almost independent from its owner.

One employee pointed out that it was planned to turn the Company into the main software producer of the semiconductor manufacturer. One can imagine this as some kind of external IT department. However, this never happened.

At the end of the 1990th both companies united and are the Company today. Furthermore, a new product was created. This product is a web-based supply chain management system as described later in this chapter. The Company was now developing and marketing both products. At some point there were about 45 people employed by the Company. However, this number decreased over the last years, because of the bad economic situation.

In 2004 the Company was sold by the semiconductor manufacturer to an Italian software vendor. Furthermore, both bosses sold their remaining shares to the Italian company. The Italian company is listed on several stock exchanges. This event turned the shape of the Company totally. Now they sell the products of the Italian company as well. Their supply chain management system is also now offered internationally. Previously, the Company was only focusing on the German market. Right now the number of employees is increasing again, because additional man power is needed to support the Italian products as well.

From this small historical overview one may see, there have been many changes for the Company in just about 13 years. It started as two small independent companies and is today a part of a big international stock company. Before discussing the Company's culture in detail, I first present the formal organisation.

## **2.3 Formal Organisation**

The Company is developing two products: a production planning and control system and a supply chain management system. In this thesis the first product is from now on called PPS and the second system SCM. The PPS product is described in detail in section 2.4.1 on page 9 and the product SCM in section 2.4.2 on page 11.

In addition, the Company is also marketing the products of the Italian company. That are different complex software packages for retail and replenishment planning. In this thesis all those products are now called RETAIL. A detailed description can be found in section 2.4.3 on page 11.

The Company's formal organisation is influenced by the marketed products as well as the fact that the Company is spread over two locations.

For all three products project management services are available. Normally, the products are introduced to a customer in a project. Depending on the complexity of the project and the customer's requirements, purchasing the different products is an investment from 50.000 Euro up to 300.000 Euro. However, much bigger projects are also possible.

Since the products PPS and SCM are developed by the Company, development teams for both products are needed. To acquire new customers and to generate new contracts, much effort is needed. Marketing, sales and pre-sales activities are done by several marketing and sales people. Beside maintaining and cultivating contact to possible customers, advertising and presenting the products at fairs, this also means producing a lot of product descriptions and marketing documents. Usually, before a customer orders one of the products normally several presentations are given and a detailed specification must be generated. Therefore, many people are employed in the marketing and sales department. At the moment the Company is even trying to extend this department with new employees.

After a customer introduces one of the Company's products, user support and maintenance is needed. There is support available for the own developed products PPS and SCM. At the moment, the Company tries to also train people to give support for the RETAIL products.

The Company's administrative overhead is pretty small. Both bosses each have a secretary. One of the secretaries is also doing accounting and financial control. She is also responsible for payroll accounting. The other secretary is supporting marketing and general administrative tasks. In addition, there is a system administrator. He is responsible for all hardware and software systems used at the Company as well as the internal network and the technical integration with the Italian company. He also supports project management during deployment of the products.

The logical structure is shown in figure 2.1 on page 8. This figure shows how the different functionality needed by the Company is separated without paying attention to the different locations of the Company. The number of employees is given in brackets. Those numbers are not accurate, because most employees work in different parts at the same time. Furthermore, the Company trains about five students, who only work at the Company full-time for some months during the year.

The view on the company structure changes completely, if the two locations are taken into account. By looking at figure 2.2 on page 8 it can be seen that many functions are spread over both locations. For example, there are project managers in both locations, the same is true for sales and marketing employees.

The sum of employees in figure 2.2 is bigger than in 2.1, because many employees are fulfilling several functions. For example, the boss in Halle is heavily involved in all marketing and sales activities. He says that he spends up to 80% of his time doing sales. The other boss is working much in general business administration like financial control and contract negotiation and handling. He is also doing project management work for the RETAIL products.

At this point the reader might be already quite confused. However, in another view on the organisational structure of the Company it can be seen how responsibility for the different functions is divided. Figure 2.3 on page 9 presents the official organisational structure.

This figure implies that the official boss of some employees is not located in the same office but 150 km away. On the other hand, the bosses have to supervise employees they normally do not work with all the day. The reason for this strange organisational structure can be found in the Company's history. Both former companies had their own marketing and sales people and of course project managers were also needed at both companies.

Of course this separation of functionality and responsibility between both locations creates many problems. However, the Company also has an informal organisation and a very interesting company culture to deal with those problems. Before informal organisation and company culture are discussed, I present the different products in the next section. To understand the Company one

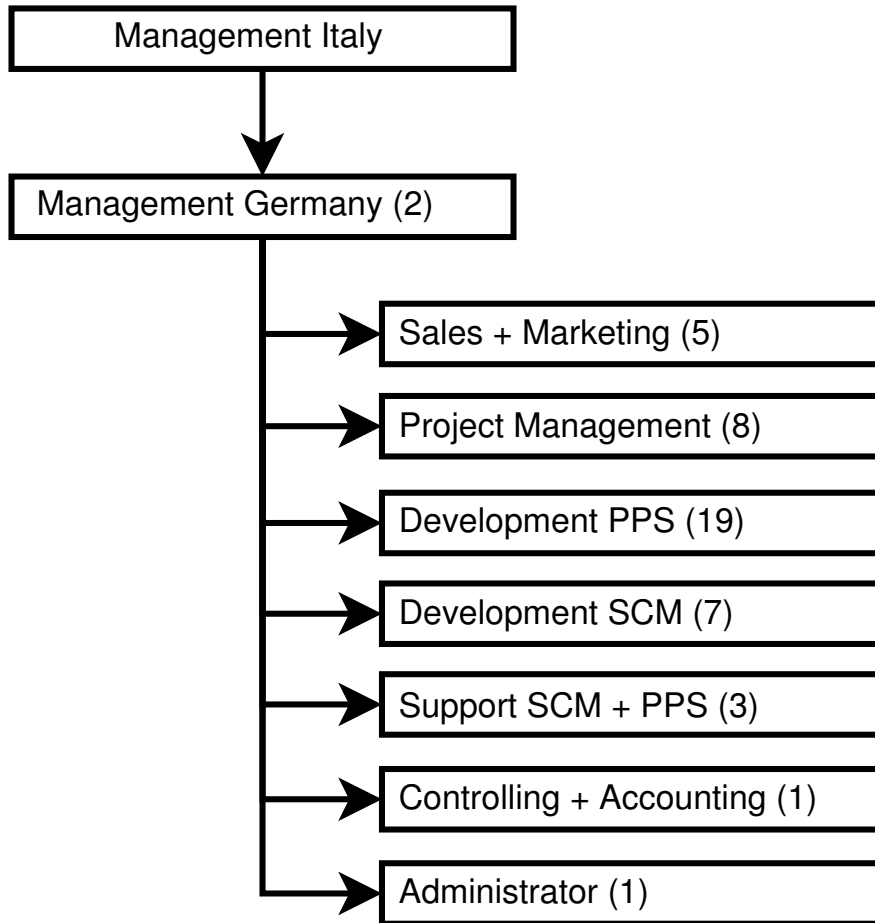


Figure 2.1: Logical structure of the Company

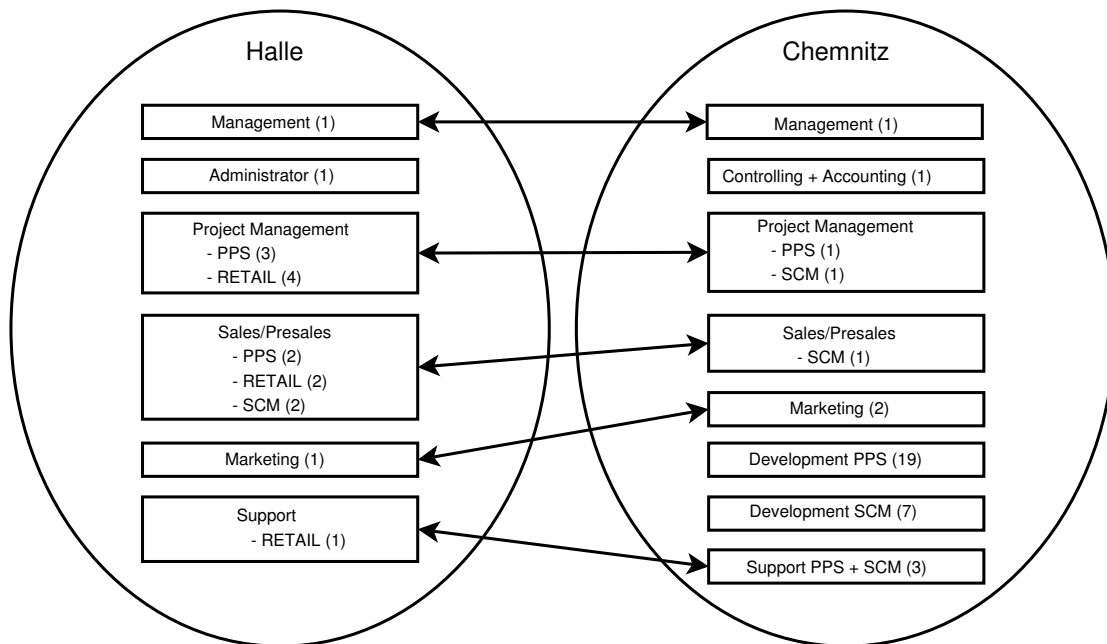


Figure 2.2: Separation of the different functions between both locations

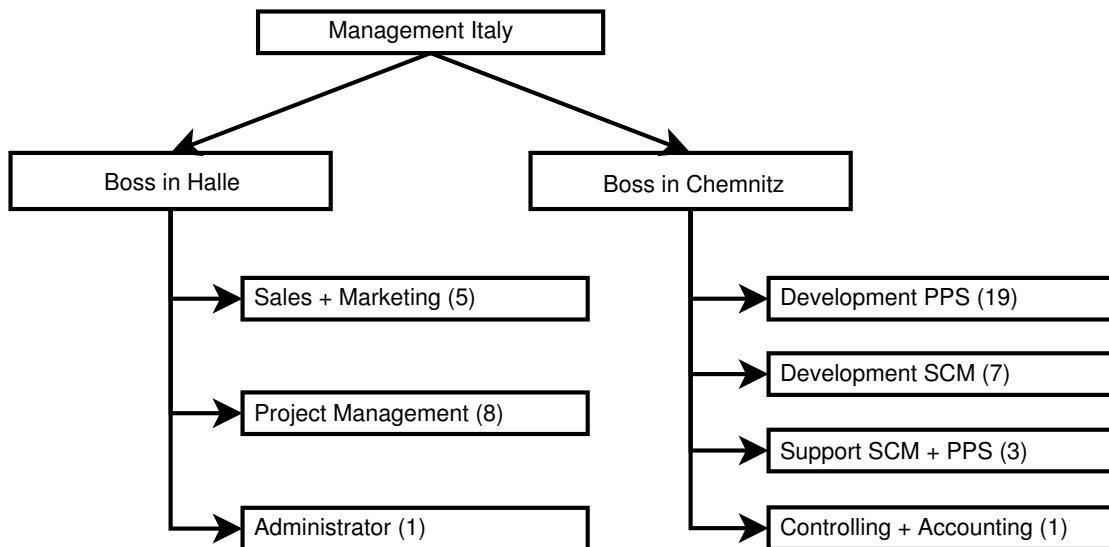


Figure 2.3: Official organisational structure

has to know the products and the way they are developed, deployed and marketed.

## 2.4 Market and Products

### 2.4.1 Production Planning and Control System (PPS)

According to many employees, the production planning and control system PPS is the heart of the Company. The former company in Chemnitz was founded to develop this product. About 100 man years were already spent in the development of this product. The boss in Chemnitz did much of the initial programming and he also worked as the project manager in many projects. Today, still more than half of the employed people are working on this product, either developing or marketing it.

At this point it might be interesting for the reader to describe the idea and functionality of such a system. First of all such a system has to know the products a company is going to produce. For each product a technology and a parts list is required. A parts list describes which (raw) materials are needed. The technology describes the steps the different materials are put together. Each step is an activity. In an activity often machines are involved as well as staff with specific training for those machines. Since there are so many different kinds of machines available, the general term resource is used. Beside a parts list, a technology, materials, staff and resources one might also need special tools for production.

Usually, the above mentioned resources, tools and staff are limited. This means, if a company receives an order, it has to plan exactly how to use them efficiently. To be able to do that, additional information about the resources, tools, and staff must be given. For example, a machine might have a limited amount of work it can do. A tool might have a limited durability and must be replaced after some time. Furthermore, staff is not always available, they might be sick or on vacations.

A PPS has to maintain all this data. Normally, it can do this not just for a single factory but for several factories at the same time. To reduce the amount of data needed, the so called variant management was invented. For example, a T-shirt must be produced in different sizes and colours. Instead of creating a technology for each colour size combination, all those combinations are handled as a single product. If a new order is created for this product, one has to choose the colour and size. Based on this decision the program selects the appropriate technology and parts

list, because for a bigger T-shirt more material might be needed. If there is not enough material available, the system can create a purchase requisition.

Depending on the size of the manufacturer, the system might have to maintain several thousand products and orders each year. Such a system should help to optimise the production. Therefore, the production is simulated in advance. For this simulation several different algorithms can be used. A basic algorithm is the so called MRP II (manufacturing resource planning) algorithm [SH05]. Here, the system just ensures that an order is spread over the different resources, so no resource is used over its limit. However, it is still possible that two orders are using the same resources at the same time and an overload of a resource is created. On a first glance, this algorithm might seem not very useful, but still it is widely used in industry, judging from the author's experience. Usually, a human operator prevents resource overload.

Another simulation algorithm prevents those situations on its own. Such an algorithm is often called APS (advanced planning system) [SH05]. The Company's products support both algorithms. Depending on the user's need the appropriate algorithm is chosen.

The Company has customers from many different industries. Even though the general planning principle is the same in all industries, there are also many differences. Therefore, the product is available in special versions for each different industry. Those special versions are based on the same core, but the modules around the core vary. It can be seen that the PPS product is developed as a software product-line [CN02, e. g.]. In addition, most customers also have unique functionality. This functionality is only available for this particular customer. However, the Company tries to generalise such functions and to include them later into the core assets.

From the product's description so far it should be clear that introducing the PPS product in a company is not just a task of installing a software package. Normally, several months of preparation are required. If a contract is signed, the Company creates a project. This project team consists of a project leader and developers as needed. The project leader is working together with the customer to model the system and specify required functionality. Modelling means, for example, to structure the customer's products in a way that it can be handled by the PPS software. Furthermore, the end-users must be trained and the deployment phase must be planned in detail to minimise system downtime. Usually, a customer also requires additional and often unique functionality. The project leader specifies those requirements together with the customer. This specification gets implemented by the team's developers and the PPS development department in Chemnitz. Normally, it is the task of the developers and the project leader to test the system before it is deployed. However, in case of drastic changes or complete new functionality additional support is provided by specially trained employees in the development department. The system is deployed at the customer either by the project leader, by the PPS support team or by the Company's system administrator. In most cases all those different people work together.

After deployment of the product the project leader maintains contact to the customer. Often customers have additional requirements after some months of usage. Furthermore, the project leader helps the customer to upgrade the system to a new version if requested.

The PPS product is written in a structured programming language. The client is only available for Microsoft Windows. All major database management systems can be used for data storage.

The PPS functionality is huge and complex. The system can handle almost every case one may think of. In this context it can be said that the system provides more functionality as a general purpose production planning and control system. Unfortunately, this complexity can not be hidden in the product. The user interfaces are quite complex. For example, one can change options of an order in more than 10 different configuration pages. This complexity also means that the PPS product is not made for the mass market. A typical PPS customer already tried different software packages without success. It can be said that the Company is a specialist in this market and only handling customers with very complex requirements. This also means that the projects get very complex.

Today the PPS product is mainly used in fashion production industry and semiconductor in-

dustry. There are also other manufacturers using the product. They are summarised as discrete manufacturers.

Even though PPS is still the main product of the Company, it is not going that well. There are several reasons. One reason might be that the general economical situation is not very good. On the other hand, many former manufacturers are moving their production far away from Europe and so they do not need production planning and control systems anymore. For example, most of the fashion industry today is not producing anymore in Europe. They just create the design and marketing strategies for the fashion products. The production is done for example in Asia.

Another problem is the very competitive software market. A quick market research<sup>1</sup> shows more than 100 software packages and solutions aiming at this market. It is not easy to stay competitive in such a context. Many of those solutions are supported by big companies.

At the moment it does not look that good for the product PPS. This might change if new customers can be acquired. If not, the Company will have to consider to give up the PPS product. Both, employees and management are aware of this problem.

#### **2.4.2 Supply Chain Management (SCM)**

The second product SCM is a web-based software for supply chain management. Supply chain management for a manufacturer means to integrate suppliers and also customers with its own enterprise resource planning systems. For example, a manufacturer might make an offer to a customer. At this point the manufacturer can not only check his own internal resources and stock, but also directly check the resources and stock of his suppliers. In most scenarios a supply chain management system is used as a business to business (B2B) platform [Cun02, e. g.]. This can include electronical market places. For example, a manufacturer can use the system to make a purchase requisition. Possible suppliers are informed about this and can make their offers. The manufacturer receives all the different offers and can decide which supplier should be contracted.

Normally, such a supply chain management system is introduced by a big manufacturer. This manufacturer forces its suppliers to integrate with the system. There are different possible levels of integration. A full integration means, the SCM's functionality is directly accessed through the used enterprise resource planning (ERP) system. Usually, such a high level of integration is only done by the manufacturer and very big suppliers. All the other participants use a web interface, which can be accessed with a web browser.

Depending on the level of integration the users have to pay a monthly fee for the usage. For a very small supplier this might be just several hundred euros per month, for bigger users the monthly fee is above 1.000 Euro.

The Italian company also had a similar product. It was decided to give up the Italian product and to integrate the missing functionality in the Company's SCM product. In recent months there have been some huge contracts with international companies using the SCM product on an international level. Because of this, some PPS developers were already moved to the SCM development team.

The SCM product looks quite modern. It is based on Microsoft .NET [Ric02, e. g.]. Still, a lot of knowledge taken from the PPS product is included in the SCM product. For example, both products use a similar data model.

At the moment the future of the SCM product looks promising, even though a lot of marketing and sales efforts are needed.

#### **2.4.3 Demand Planning Software (RETAIL)**

The Italian company also have several different products which are now distributed and marketed by the Company as well. The Company has some initial success in marketing and selling the

---

<sup>1</sup>see for example <http://www.softguide.de/software/produktion.htm>

Italian products. Different products are available, but at this point I summarise all those products as RETAIL, because the products can be used for retail planning. Retail planning means to plan how many products can be sold in a shop in a given time period, when new products are required and how the price has to decrease over time to ensure that the products really get sold. A typical fashion producer, for example, owns about 400 shops all over Europe with its own brand. The fashion producer wants to have a new collection in the shops each month, so it must ensure to sell a complete collection each month. However, it is also not possible to have nothing left in the shop at the end of the month. So the next collection must be moved into the shop step by step.

From this small example it can be seen, retail planning is not easy. It is still a manual work done by humans with long term experience in planning. The available software packages help the human planner to extract the necessary data and to visualise possible scenarios. But still the human planner must decide how many products will be sold and when to move the next products into the shop.

One main reason not to automate the planning process is that almost all companies plan in a different way. Some might have just four different collections per year, another one might not plan by collections but by colours to ensure that always products with similar colours are offered in a shop.

A typical installation of the RETAIL products costs around 300.000 Euro and even more. A project team is created consisting of about three consultants and a project manager. The consultants analyse and document the customer's planning process. Afterwards, they customise the RETAIL product so that it supports the identified planning process. This customising is not done on the code level as in the PPS product. The RETAIL product has an internal scripting engine to provide access to the internal data model. In addition, the consultants create customised data views. In the end the deployed product is looking like an advanced spreadsheet application. One can visualise estimated sales on different abstraction levels like all shops in a region or all shops in a country.

After the project is completed, usually the customer is just getting support. Normally, there are no additional change requests. However, often customers come back after some time to improve their planning processes even more with additional RETAIL products. In such a case a new project is initiated.

## **2.5 Informal Organisation**

The formal organisation as described in section 2.3 on page 6 puts some constraints on the Company. For example, people who should normally work together are separated by different company locations. As they would have to drive about 2 hours by car for a meeting, it is not easy to meet for them. Another problem is that development is happening in one location and sometimes the employees in the other location are not aware of newly available functionality. Another problem is sharing of electronical documents like templates, product descriptions, marketing material, technical documentation, database scripts, program versions, etc. All those files must be exchanged by a network connection. Therefore, it was a big improvement to establish a fast network connection between both company locations about 2 years ago.

One way to overcome the problem of not being able to meet personally is to have extensive phone calls. Several times I observed phone calls lasting several hours. According to the bosses they also phone each other several times a day. To phone each other is a very common task in the Company. I did not observed that employees felt disturbed when receiving a phone call. Furthermore, if an employee is out of his office, he normally runs back to his office if he can hear his phone ringing. It is very uncommon not to answer a phone call or not to call back if requested.

Beside just phoning each other, the employees also sometimes use remote desktop sharing. For example, if a project manager from Halle is experiencing a problem in his version of PPS or SCM a developer from Chemnitz can log on his machine and debug the program. A video



conference hardware is also available, but I only observed it being used in meetings between Halle and the Italian departments. Another communication tool is of course correspondence by email. However, emails are not used for discussion, but more often to summarise the current state. I often observed that something is first discussed by phone and later additional information like a file is sent by email.

The above described phone and email communication is the normal way work is done in a company. However, I do not consider the way people work together in the Company to be very common. For example, I did not observe much politics going on in the company and I did not encounter any workplace bullying. Every employee in the Company is respected and everybody really seems to want to help each other. If tasks get not done on time, it is mostly because of work pressure and time constraints and not because tasks given by another employee are not considered to be important. For example, once I asked several employees for documents and I received them at the same day by email. In addition, some were asking me personally about this issue and gave me additional tips. At another point I conducted a small survey in the company. I put the survey online on a web server by using an electronic survey application. Then I asked the employees by email to fill in the survey. More than 70% (30 completed surveys) of all employees participated. Some even send me additional comments about the issues I asked for in the survey.

In the English language the personal pronoun “you” is used to refer to the one talking to. In the German<sup>2</sup> language there are two different personal pronouns, the informal one is “du” and the formal one is “sie”. The first one is used when a close relationship like friendship exists between the two of them. In this case people also use their names like “Steve, could you (du) please explain this again?”. The second personal pronoun is used when having a more formal relationship. For example, if a student is talking to his teacher, he would never call his teacher directly by name (Steve), but would instead say “Mr. Meier, could you (sie) please explain this again?”. It is considered to be an offence if a student calls his teacher directly by name. Therefore, it is very complicated for Germans coming to a country where everybody calls each other by their names and not family names. In the Company I noticed that many people call each other also by their names and not family names. The informal “du” is used very often in the Company between the employees. This surprised me a lot. Usually, at work in Germany people call each other by family names. Only if the people are of the same age or share for example spare time activities, they use the informal personal pronoun. In Germany it is a common practice to call someone by the formal personal pronoun and his family name as long as the other has not offered the informal personal pronoun. During my study I was doing an interview with an employee. I had prepared all questions using the formal personal pronoun. At the beginning of the interview the employee offered me the informal personal pronoun. It was a challenge for me to reformulate all the questions during the interview using the informal personal pronoun, because it symbolises a different relationship to the interviewed person.

By using the formal personal pronoun one expresses some kind of hierarchical awareness to his communication partner. In the Company most employees use the formal personal pronoun while speaking to both bosses. However, some employees are called by the bosses by their name. I was not able to identify a pattern for the people being called by name, but I think that it heavily depends on how long they had worked together.

The usage of the informal personal pronoun between the employees is one example of the good and open relationship among the employees. One employee described this as a feeling of being a family. For example, there are no real hierarchies in the Company besides the two bosses. However, I would not characterise the Company as a typical IT startup company. It is a company, which is still small enough so that everybody can know each other. Therefore, hierarchies and formal orders were not needed. However, now the Company is going to grow again and it will be

---

<sup>2</sup>Here, German and Swedish are similar. The concept of different personal pronouns depending on the relationship of the people communicating is even more complex in Japanese. In Japanese there are 3 different personal pronouns, the most formal one is only used while talking to the emperor.

interesting to see how the company culture is going to change so that it can cope with the growing size.

Another term describing the Company very well is “down to earth”. This term means to be solid and not to imitate anything you are not belonging to or you aren’t. For example, within the Company job positions and departments have German names and not English ones. Another example is that project managers are called the German term “Projektleiter” and not for example consultant or project manager. The Company is not using many English words in their product description. In contrast, they try to find German words for many terms. Sales persons do not take the most expensive hotels but instead look for cheap offerings. One more example is the voice you can hear while waiting at the phone to get passed through. The voice in the waiting loop has a very strong local dialect and is not the perfect standard voice you can hear in the waiting loop of other companies. The employees are wearing ordinary clothes at work and only change to suits when visiting a customer or a customer is visiting the Company.

The Company also tries to fulfil its social responsibility. For example, employees can hand in bills of up to 50 Euro during Christmas time, if the money was spend for children clothes for the employees’ children. This is a not so common thing in most companies. Furthermore, there is a joint Christmas party for all employees each year during Christmas time. The party is done in a restaurant and the bill is paid by the Company. Another nice thing happens on birthdays. A sheet of paper with a birthday graphic is signed by all employees and given to the “birthday child”. Most of the employees also spend a breakfast or a cake on their birthday. While I was doing my study at the Company, one employee in Halle got a child. Employees were collecting some money for him. The same happened when a long term employee left the Company. This employee also gave a good-bye party for his colleagues. In general, the Company has a very low employee fluctuation. For example, in the last 3 years only one employee left the Company in Halle.

In summary it can be said that the people at the Company have found a very good way to work together. They avoid politics and also hierarchies are not needed to get work done. The relationship between the people is very good as well and the Company is trying to support its employees.

## **2.6 And still Two Companies**

Even though the employees try hard to work together, one can still see that the Company is a made up of two former companies. There are many things, which are still different in both company locations. For example, in Halle the doors to the different offices are open almost all the time. In contrast, in Chemnitz most doors are closed during the day. In Halle two employees share one office, in Chemnitz there are up to four people in one room. In Chemnitz there is a breakfast around 10 am. All the people gather in the meeting room to have a coffee and to eat something. The same happens in the afternoon, even though here not everybody joins. In Halle such a breakfast is not happening. In Chemnitz also many people go out at lunch time to eat together. In Halle only few people are getting together to eat something. Another ritual in Halle is the fact that one is going to all employees to shake hands at the morning when entering the office. However, some of the new employees do not follow this ritual.

Working times seem to be also a little bit different. In Halle most people start working between 7 am and 8 am and they do not leave the Company normally before 5 pm. In Chemnitz one can notice clearly that the office starts to get empty after 4 pm. Additionally, in Chemnitz some employees only work 6 hours each day.

Another difference is the relationship between boss and employees in both locations. In Chemnitz the boss has a bigger authority. He has to decide a lot, employees are consulting him frequently. In Halle the boss is mostly doing sales work and he is not that involved into the work of the other employees. Even though the other employees respect him, he does not seem to be as

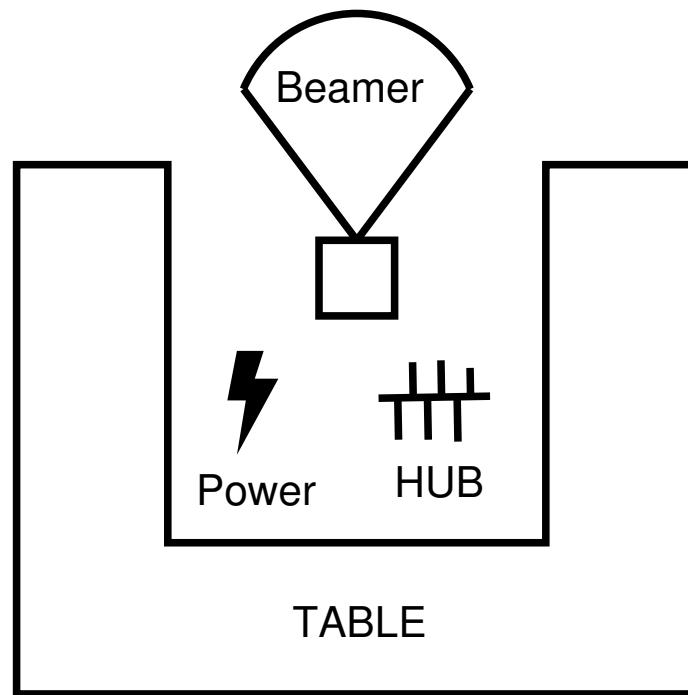


Figure 2.4: Meeting room in the Company

authoritative as the boss in Chemnitz. For example, someone not knowing the Company at all and observing a meeting in Halle might not be able to identify the boss on a first glance.

Another difference between both company locations is the amount of travelling the employees do. Most employees in Halle have to work most of the time together with the customer whereas the employees in Chemnitz are not required to work with the customer. Therefore, the employees in Halle are travelling a lot. It is almost impossible to find a day where all employees are in Halle and not out of office.

## 2.7 Project Leader Day Meetings

About once in each period of six weeks an activity called project leader day is happening in the Company. This is a one day long meeting in one of the Company's locations. In such a meeting all sales and marketing people, all project leaders, both bosses and the heads of the SCM and PPS development departments are meeting. In general, only the SCM and PPS development programmer are missing as well as people being on holiday or who are ill. In most meetings between 15 and 20 employees participate.

This big group of employees is a challenge, because in both company locations there are only meeting rooms for about 12 people. The meeting room is illustrated in figure 2.4 on page 15. The meeting rooms are quite crowded. After the employees from the other company location arrive, everybody is gathering in the meeting room. Many employees bring their notebook along with them. For someone from outside it might look like a small fight as the employees with a notebook try to get access to the power supply and the ethernet network HUB. The table in the meeting room is looking like a big "Ut't". In the middle of the room are the power supply, the network HUB, and the video projector. The employees are sitting around the table. About half of them is having a notebook with them. In my fieldnotes I described the meeting as a "notebook gathering", because I often had the feeling all those notebooks were not really needed. Only some of the employees give a presentation. The notebooks are used to take notes, but mainly to check emails during the meeting.

The meeting never starts before 10 am, because employees from the other company location have to travel to the meeting. In most cases the meeting starts about 30 minutes later as scheduled. Even if the employees arrive on time, many conversations in small groups are going on before discussing important work related issues.

Normally, the meeting is moderated by one boss. He introduces the different presentations and discussions. In addition, one employee has prepared the agenda in advance and is also providing a protocol of the meeting some days after the meeting. The meeting is organised in different presentations and discussions. The scheduled time for a presentation or discussion is never longer than one hour. A presentation is usually given while sitting at the table. The cable to the video projector is handed to the person going to give the presentation. Beside being able to show a presentation, getting the video projector's cable also means getting the attention of the group. Only some presentations are given using a presentation program. In most presentations the PPS, SCM, and RETAIL products are shown to explain for example new features or special solutions (e. g. an individual feature for a customer).

Usually, the meetings take about 6 hours. Usually, there is a 30 to 45 minutes break for lunch around 1 pm. Many employees go together to fast food shops or small food restaurants near the office. The casual conversations during lunch are not always work related. Other employees do not go out for lunch but instead use the time to meet with other employees.

During the meeting not everybody stays in the meeting room. For example, if someone has another important task to do, he might only come back for presentations and discussions concerning his work. Another reason to leave the meeting room is an important phone call. It is accepted to leave the room if, for example, a customer is calling. I found those meetings very exhausting. It is quite hard to concentrate for such a long time. Especially after lunch it is hard to stay attentive.

At the beginning of the meeting, usually new features in the PPS and SCM products are presented. The bosses might report about meetings and conferences with the Italian owner of the Company. The presentation of a certain new feature is repeated. During my fieldwork I observed three presentations about the same feature at three different project leader day meetings. In the first meeting the new feature was presented and the meeting was used to gather ideas and inputs from the employees on how to implement it. In the second meeting a prototype was shown. In the third presentation the feature was almost implemented and it was presented again. Those presentations are also used to present the current development version of the PPS and SCM products, what was already implemented and what still needs to get implemented for the next release.

Usually, the first presentations and discussions run over-time. Therefore, there is a rush through the remaining points on the agenda after lunch. Some topics are repeated in each meeting. For example, the marketing and sales employees present how much was sold in the current quarter and if there are any big deals to come. They call this the "sales pipeline". The project leaders present a short summary about the current state of their projects. They also point out how they dealt with problems in their projects. Sometimes they also present problems they are currently facing with. In such a case the employees discuss how the project leader can solve the problem.

During the meeting the atmosphere is quite relaxed. Coffee and mineral water is available. Employees are doing small talks with each other in the breaks. Discussions are focused on the topic and not on the person. Furthermore, it is not forbidden to challenge the bosses' opinions. I was often surprised that very far-reaching decisions are taken after a short discussion. It can happen that something unresolved for months is solved after just some minutes of discussion. Beside such discussions the meetings are mainly used for news and knowledge exchange. In addition, it is a social event, because the employees from the different company locations meet personally. It can be said that the project leader day meetings are quite important for the Company. Therefore, it is a shame that they only happen every 4 to 6 weeks.

## **2.8 The International Part**

The Company is today a part of an international Italian company. One obvious change is that now all product documentation must be available in English. This is a challenging task, because this was not required before and most employees are not that trained in writing English documents. Therefore, English courses for the employees are organised. During my study a several months lasting English course was held in Halle. The course was given twice each week during work time.

Another way to practice English is to write English emails. There are some internal mailing lists and some employees started to send emails in English for practice. However, not everybody is joining those efforts, because it seems to be a little bit artificial.

Another change is the fact that especially the SCM product but also the PPS product are now offered on the international market. Therefore, some employees are giving training sessions in Italy for the sales employees there.

Once during my study the president of the Italian company visited Halle and Chemnitz. He also made a speech to the employees, which turned into a dialogue later. Some employees were interested how the whole company is going to expand in future. The president pointed out that he is satisfied with the German part. In addition, he was asking whether working together with Italy is going well. Some employees answered that in general things are going well, but that communication between both countries must be improved.

With help of the Italian company, the Company was integrated into their intranet. Some parts of the IT infrastructure like email servers is now managed in Italy. Furthermore, it is planned to integrate the Company's product descriptions, press releases, success stories, contact pages, etc. into the main Internet page. However, this has not happened yet.

Probably the biggest change is the fact that the Company is now also offering the RETAIL product. Already about five employees changed their job position to support this product for example as project leaders, sales people and support agents. The sales of the RETAIL products already make a significant part of the Company's turnover.

Today the Company is stock exchange listed. This means, financial data and company reports are available for the public. This creates an higher pressure for economical success not known before to the Company as both bosses confirmed. Now the company performance is judged each quarter of the year based on the value of the newly acquired contracts. The value is increasing from year to year by some percent. To catch up with those needs additional sales person are required and the sales and marketing department is extended.

## **2.9 My Previous Five Months Practical**

My first contact with the Company was in 2002/2003. I did a five months practical in the Company. This practical was part of my former studies of business informatics. In this study program the students have to do a five months practical in the 5th semester. Such a practical should provide a first insight in industry as well as a focus of interest for the remaining three semesters at university after the practical.

The contact to the Company was established with the help of a professor at my former university. First it was planned to do my practical in Chemnitz. However, since I was not able to get a cheap accommodation in Chemnitz at that time, I did the practical in Halle instead. This turned out to be a fortune, because the main work focus in Halle was on project management and I actually wanted to do my practical in this area.

One project leader was assigned as my advisor for the practical. I got a desk with a PC in his office. This office had a door to the office of the system administrator and another project leader. At that time the company office was located in a bad district of Halle known for its drug dealers.

Additionally, the office building was not very good, for example in summer the offices got quite hot.

The project leader was working on a project with the semiconductor customer. They wanted to extend the PPS product to include another simulation algorithm, which would better suite the needs of the semiconductor customer. Therefore, we developed a first prototype and discussed this prototype and possible solutions in a meeting with the customer. Afterwards, we specified the solution, which was implemented by the PPS developers in Chemnitz. In this context I got contact to some PPS developers.

This extension of the PPS product was not the only project I participated in during my practical. I also supported the marketing and sales employees by, for example, maintaining the Company's homepage. I also helped updating some product presentations and I checked out other marketing materials.

As my studies also included programming courses, I used my programming skills to develop some small tools used by the system administrator to deploy the PPS product at a customer. Furthermore, I worked about 2 weeks in Chemnitz to support a big test effort of the PPS product. This also helped to get in contact with many developers of the PPS product in Chemnitz. However, at no point I was involved in anything related to the SCM product. Consequently, I got almost no contact to the employees working on this product.

At the time of my practical the Company was still owned by the semiconductor customer. Even though the economic situation was not that good, there was said nothing about a possible sale to the Italian company. One of the bosses confirmed, this idea came up about one year later.

## **2.10 Locating Myself**

From the section above it can be seen that I got access to the Company during my five months practical. I have started this study again by working together with the project leader I already worked with. However, this time the tasks were more broad, not focusing on a single product. I participated in software process improvement efforts, which took place on a company wide level and are described in chapter 5 starting on page 44. All people working on the RETAIL product are located in Halle. Through casual conversations, by reading product descriptions, and by attending some presentations of the product, I also got a good overview of this product. However, during this study I did not have much contact to the SCM development team. I was only able to make initial contact to some of the team members. It can be said that I got a deep understanding of the PPS and RETAIL teams as well as an understanding of how the Company works in general. This includes how general business administration tasks are done within the Company. During my study I also participated in four project leader day meetings, which are described in section 2.7 on page 15. Those meetings were a good opportunity to get an impression how the whole Company works together and how current problems are solved.

## Chapter 3

# Ethnography

In this chapter I am describing the research method ethnography in general and the application of ethnography in software engineering in particular. I start with a general presentation of ethnography.

While explaining the purpose and nature of my study to friends and relatives I found it useful first to start with a general explanation before relating to software engineering. This way of presenting ethnography also reflects the way I was approaching the field. Before I studied specific ethnographic studies in software engineering, I first investigated general literature [Fet98, HA00, VM00, e. g.] about ethnography.

The general discussion of ethnography in section 3.1 on page 19 is followed by a general discussion of the philosophical view behind ethnography in section 3.2 on page 21. Afterwards, I discuss in section 3.3 on page 22 how ethnography is used in software engineering. In section 3.4 on page 24 I present some software engineering studies with a similar research design. In the remaining sections I discuss how I have implemented ethnography. This also includes a discussion of doing ethnography in the own domain as well as a discussion about how to present an ethnographic study.

### 3.1 Ethnography in General

The word ethnography consists of the two parts “ethno” and “graph” [Wol99]. The part “ethno” means something like people and “graph” means a picture or image. Based on those two meanings also the idea of ethnography can be understood – to make a picture of a group of people. When I was asked by friends what I am actually doing in my study, I always used the following example. Everybody has already heard of studies where a researcher is for example going to a small island or deep into the rain forest to live with a native tribe for several years. The researcher has to learn the language, culture and rituals of the natives. Afterwards, he returns home and writes an extensive report like a tale about his trip and the people he studied. This is an ethnography in anthropology. One of the first ethnographic studies done in anthropology was by Malinowski [Mal22] at the beginning of the 20th century. Later ethnography was also applied in other social sciences like sociology [Why93, e. g.], education research [TGA81, e. g.], but also in police and crime research [MVM78, e. g.].

The common base in all those examples is to study a group of people for a long time of at least several months and to write an extensive description of the studied group [Fet98]. Back in the old days of ethnography in the late 19th century, ethnography scientists were studying reports of salesman, sailors, and missionaries who travelled foreign countries. By combining those reports a broad picture of a group was given. A good example are the stories of the German writer Karl May<sup>1</sup>. He wrote detailed stories about US Indians and cowboys without ever having travelled the

---

<sup>1</sup>see e. g. [http://en.wikipedia.org/wiki/Karl\\_May](http://en.wikipedia.org/wiki/Karl_May)

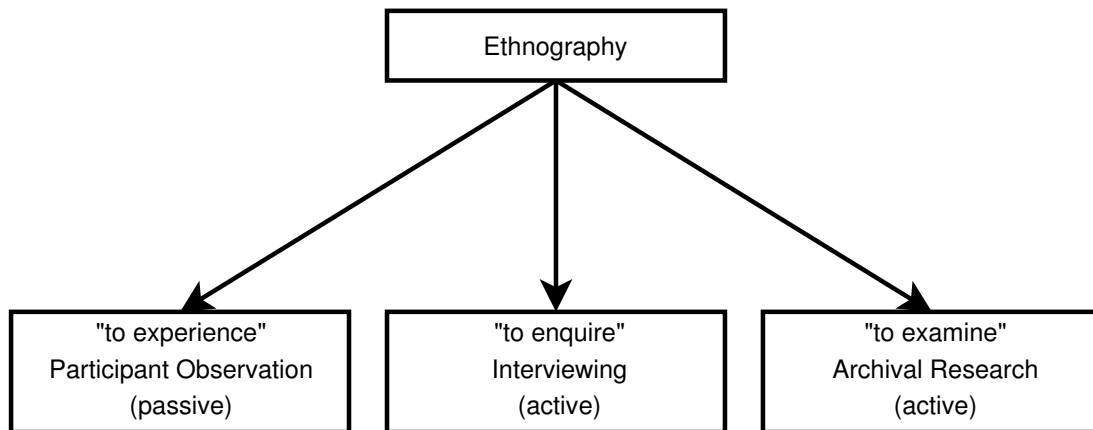


Figure 3.1: Elements of ethnography discussed by Wolcott [Wol99]

country. His writing was based on imagination but also on travel accounts, anthropological and linguistic studies from those areas. While reading the books the reader can imagine how the scene has looked liked and how his characters have lived.

Still, such reports have a major flaw – the researcher has never actually experienced the group of people he is writing about. To overcome this problem researchers started to travel to the people and live together with them. This was a major change and can be described with the phrase “going native”. Researchers started to embed themselves into the group they were studying. Still, in most cases people with completely different cultures like natives or low society groups like drug addicts were studied [Wol99]. Normally, the studied people never read the reports produced by the researcher about them [HA00].

Many ethnographic studies at the beginning presented the researcher’s view on the studied culture [HA00]. This attitude also changed over the time. Today it is the aim of an ethnography to present the view of the studied people and not the way the researcher is seeing the studied culture [HA00]. Also nowadays ethnographies are done among the own people. In addition, ethnography has not to be done by a social scientist, but can be done by anyone interested in fieldwork instead. This means, today the studied group does not have to be unfamiliar anymore to the researcher [Wol99].

From this small historical overview it can be seen that ethnography changed much. First only reports about other cultures were studied and combined. Those cultures were often not as “civilised” as the researcher’s own culture. The first change for researchers was to actually travel to the people they studied and to live with them. For example, Whyte [Why93] lived in a slum to study the life and culture. Later ethnographies were also written about one’s own culture and domain. For example, Wolcott [Wol99] worked as a teacher and school principal while doing an ethnography about schools at the same time.

### 3.1.1 Elements of Ethnography

Just to live with a group of people is not enough to do ethnography. We are all day living in some group of people, without necessarily doing ethnography. Instead, one has also to examine the past of those people like produced artefacts. Furthermore, it is not enough to just participate in their daily live, something called “participant observation” [Wol99]. Instead, one has also to ask direct questions for example by interviewing people. Those different inquiry methods can be structured according to Wolcott [Wol99] as shown in figure 3.1 on page 20.

The most often associated work of an ethnographer is *to experience* by participant observation. This means, the ethnographer is living and working with the studied group without any active action to investigate the field. This can also include casual conversations. This means, he is not



doing any surveys and is not asking direct questions to examine certain facts or knowledge. In participant observation, the ethnographer participates in the group actions as an ordinary group member would do.

In contrast, the ethnographer can take an active role and ask people direct questions. This can be done for example in formal interviews, but it is also possible to ask specific questions in a casual conversation. Therefore, this part of ethnography can be described as *to enquire*.

In addition, each culture is producing artefacts. For example, companies produce many documents while working. It is also an important task to study those artefacts. Since those artefacts are already produced the ethnographer just has to find them for studying. It is an active action, because the ethnographer is directly searching for something. This archival research can be described as *to examine*.

It can be seen that different research strategies are used in ethnography. It is not enough to rely on a single method like just doing interviewing. Through the different methods different perspectives are taken of the studied group. This helps to reveal a much more complete picture, something not possible by applying a single method. Also it ensures to find out the people's view of their culture and not the researcher's own view on the studied culture [HA00].

## 3.2 Quantitative and Qualitative Research

There are two strategies of inquiry [Cre02], which can be used in research. One is qualitative research and the other is quantitative research. The difference between both inquiry strategies can be understood from the meaning of both terms. To quantify means<sup>2</sup> to know how much of something exists for example by counting. Qualitative in contrast means to relate something against something else. Therefore, in quantitative research a result is expected, which can be measured. In contrast, qualitative research produces results showing relationships between the studied items.

By choosing an inquiry strategy, one has not yet defined which research method to use. Often methods can be applied for both research inquiry strategies, even though there are common methods for each strategy. For example, controlled experiments are normally used for quantitative research. On the other hand, any kind of extensive fieldwork like ethnography is normally used by qualitative research [Cre02].

Normally, there is also a philosophical perspective connected to qualitative and quantitative research [Cre02]. Doing a controlled experiment with careful evaluation of the measured data only makes sense, if the results produced help to gain knowledge about the studied object. If the controlled experiment does not help producing new knowledge and insight, than there is no reason to do it at all. This idea of gaining new knowledge by proving or falsifying an hypothesis is the base of the positivist world view. The basic idea is that the world can be observed and that theories can be created to explain the observations. Furthermore, theories can be proved by generating observations through experiments. If it is possible to observe what is going on, it is also possible to measure and count it – to quantify it [Cre02]. Therefore, positivism and all philosophical ideas based on it are often the philosophical base for quantitative research. The positivist view also creates some logical consequences. For example, to work scientific means to accurately design and control experiments and to measure results in a quantitative manner. Every research method not fulfilling those standards must therefor be unscientific and can not produce valid results [Cre02].

On the other hand, one might think that the world is just a construction of our own imagination. We can not describe the world, because there is no way to produce evidence. Everything we see, hear or feel must be first interpreted by our own senses and can therefore not be objective [Tal04]. This philosophical view is called constructivism. If all our observations are constructions of our

---

<sup>2</sup>see e. g. Merriam-Webster dictionary <http://www.m-w.com/>

own imagination, it does not make sense to create theories explaining something. Also measuring something exactly is not appropriate. However, it is interesting to learn about as many perspectives as possible. By comparing different perspectives common patterns can be identified. This comparison is like relating different experiences against each other. Therefore, constructivism is one way of explaining qualitative research.

Problems arise if for example a qualitative research study is judged based on the positivist world view [Tal04]. Of course there is no randomised selection of studied individuals and also no mathematical model is provided to explain what the qualitative researcher experienced. If one is going to judge the value of a qualitative inquiry, one can not do this from a quantitative view point. This is also true the other way around. Often social scientists claim that natural scientists doing quantitative research will never create an understanding of the world, because all theories are just constructions. Still, also qualitative researchers have to confess that many valuable techniques, tools, applications, etc. are created by natural scientists.

A common misconception of qualitative methods is that they do not require any methodological rigour. Of course it is not enough to just hang around in the field and later write a sufficient report. Instead different techniques as described in the section before must be applied. Also analysing qualitative data is not just about summarising what one has experienced. Later in this chapter I will discuss how I analysed my data using template analysis [Kin98, see] to clarify this point.

There is no easy solution for this philosophical discourse. Basically, it is impossible to say which view is correct and which is wrong. One way to succeed is to combine methods from both fields as some kind of mixed methods approach [Cre02]. An example how this can be done is given in the next section. However, for this study I did not chose a mixed methods approach. This study is pure qualitative research.

### 3.3 Ethnography in Software Engineering and Related Fields

There are not many examples of ethnography being used in software engineering. One often cited example is Sommerville et al. [SRS<sup>+</sup>92]. They report on how ethnography can be applied in requirements elicitation. In the described project, sociologists conducted an ethnography in an air-traffic control room. The gathered data was used to define the requirements for an user interface to a flight database.

Ethnography is also used as research method in the closely related research topic Human Computer Interaction (HCI). Here, an ethnography is conducted to identify the best design for a computer interface. Sommerville [Som04, p. 380] also points out that ethnography can be applied in interface design. Therefore, it seems useful to also look at ethnographic studies within HCI.

In their study of designing an image browser interface, Ormerod et al. [OMM<sup>+</sup>04] used ethnography to understand how photographs are used in private homes. They studied for example in which process and environment photographs are shown to a visitor. They identified that in many cases photographs are categorised among the dimensions *who*, *what*, *where*, and *when*. This categorisation is reflected in the user interface of the image browser. For each image the user can define the four different categories. This example shows that a user interface can be designed according to the findings of an ethnographic study.

Ethnography is also applied in Computer Supported Collaborative Work (CSCW) research. CSCW explores how people work together (collaboration) and how technology like groupware systems and computer networks can support this collaboration [Wil91]. Even though CSCW is not part of software engineering, it is also closely related. A study describing work processes can inform a software engineering effort trying to support the work with a new software [BD97, see]. A good example for a CSCW research effort using ethnography is the study by Nardi and Miller [NM90]. They investigated how spreadsheets are used within work environments. On a first glance it might look like that a spreadsheet is developed by a single person. However, often

a single person is missing parts of the necessary knowledge. Nardi and Miller found that developing a spreadsheet requires domain knowledge as well as technical knowledge. They consider developing a spreadsheet with interconnected formulas as programming. In their ethnographic study they interviewed 11 spreadsheet users. They also analysed the spreadsheets developed by the interviewees. One of their main result is that spreadsheets are in most cases not developed by a single person, but are instead a product of collaboration. A person might start creating a spreadsheet and cooperate with another person if certain domain or programming knowledge is missing. Nardi and Miller also found that in this cooperation, users teach each other.

From the few examples above it can be seen that there are different possible ways to apply qualitative methods and ethnography in particular in software engineering. The common point is that “we must pay attention to the complex interrelation of a number of organizational, cultural, technological, and economic factors” [Fug00, p. 3]. However, it can also be seen from the few examples above that ethnography is not a common approach in software engineering. In his PhD thesis Rönkkö [Rön05, p. 47] did a detailed literature survey. He concludes that there are almost no ethnographic studies in software engineering, which also use qualitative methods for data analysis.

Rönkkö [RLD02] identifies three main ways qualitative research and ethnography in particular is used in software engineering research. First, one can do a qualitative study in advance, to come up with and create hypothesis which will be falsified later using quantitative methods. Second, he says a qualitative method can be used to generate data later to be analysed using quantitative methods like statistics. Finally, he describes a third application of ethnography within software engineering. The ethnographer helps the studied group to design a software process. This means, to get aware of social processes and their complexity [RLD02].

Another classification of ethnography in software engineering is presented by Beynon-Davies [BD97]. The author talks about three categories:

- ethnography *for* information system development
- ethnography *of* information system development
- ethnography *within* information system development

*Ethnography for* information system development means to do ethnographic research to provide useful insights to developers and other practitioners. Those insights will help them to understand the way they work better. The author for example cites different studies, which revealed the importance of tacit knowledge within information system development [BD97]. Other studies cited by the author show that the end user must be involved in system development.

*Ethnography of* information system development means to study how software engineering is performed by practitioners. The author illustrates this with a study about how developers share development tasks among each other [BD97]. Other studies cited by the author investigated how communication is used during requirements elicitation [BD97].

The final application is *ethnography within* information development. Here, ethnography is used as a tool like development environments. An example presented by the author is requirements elicitation [BD97]. Here, ethnography can be used to create a domain description. Also software engineers can be sensitised for the domain by letting them do ethnographic work in the domain for some time before the actual software project is started [BD97].

Taking both categories into account it is now possible to locate this study into the offered categories. First, ethnographic work done in this study is also analysed using qualitative methods. For example, I do not measure how often a certain pattern occurs in the gathered data. Based on the classification provided by Rönkkö [RLD02], this study is a qualitative one. Still, it might be that someone reading this study is taking it as an input to generate hypothesis for future quantitative studies. On a first glance this study is an ethnographic study *of* software engineering. Still, it

might be that I reveal interesting insights for practitioners, which they can use in their daily work. In this case this study would also do ethnography *for* software engineering [BD97].

Of course there are different opinions about how ethnography should be used. For example, the often cited paper by Seaman [Sea99] emphasises to combine ethnography with quantitative methods. Here, ethnography is used to generate theories, which later can be investigated by quantitative methods. In section 4.5 on page 42 I am discussing a study by Beecham et al. [BHR03]. They used a qualitative interviewing technique to gather data. Later they analysed this data using quantitative methods.

### 3.4 Related Ethnographic Studies

At this point I am discussing related ethnographic studies. I only consider studies as related, which also use ethnography to investigate quality systems and software process improvements. I do not discuss studies investigating quality systems and software process improvements using another research design. Such studies are discussed in the next chapter, where an introduction to quality systems and software process improvements is given. To identify ethnographic studies about software process improvements I searched different literature databases for the terms “ethnography” and “software engineering”. I also tried some broader terms like “field study”, because authors might have not been aware of the term ethnography. After, I investigated the used references in the identified papers. In addition, I checked all issues of the last 3 years of different scientific journals about software engineering and empirical software engineering. Still, it might be that I have missed related publications.

Sharp et al. [SWHR99] have done an ethnographic study at different companies. They investigated how software quality management systems are accepted and implemented in software companies. In their paper [SWHR99] they focus on one particular company, which they have been studying for one week. They interviewed different employees affected by the company’s software quality management system. In addition, they did participant observation of the work processes and studied produced artefacts. From this research design it can be seen that their study was much shorter than the study presented here. Another difference is that they studied in the whole research project different companies to compare them later. This study only focuses on one particular company. For data analysis Sharp et al. [SWHR99] used discourse analysis. They also tried to identify reoccurring patterns in the gathered data. It can be seen that this is a similar approach as I have chosen with template analysis (described later in section 3.8.5 on page 33). Since the study done by Sharp et al. [SWHR99] is more broad, also the results are more general. They found that new ideas and methods only work, when employees believe in them and start to practice them. Sharp et al. [SWHR99] also found that some employees have a bigger influence than others because of their reputation within a company. They also found that many employees do not consider quality or standards to be an important part of their daily work. This implies for Sharp et al. [SWHR99] that quality procedures are not well integrated in the day-to-day work practice. Besides, software engineering results Sharp et al. [SWHR99] also formulate some conclusions in context of their used research approach. First of all they think that their research approach revealed interesting and valid insights into software engineering practices. They conclude that ethnography combined with discourse analysis can be used in software engineering research. However, they found it challenging to avoid judgement. The studied employees expected more feedback. They expected the researchers to act more as process consultants, something they could not do because of their aim to avoid judgement.

At this point it is interesting to discuss an ethnographic study done in software process improvements. Dittrich et al. describe in their overview paper [DRL<sup>+</sup>05]<sup>3</sup> different studies they did in the last couple of years. The basic idea is to work together with practitioners to help them

---

<sup>3</sup>See [Rön05, p. 57] for a report about one of these studies.

improving their software processes. Dittrich et al. call this *co-operative method development*. In contrast, to the study done by Sharp et al. [SWHR99] they provide feedback to the studied employees. This feedback is an important part of their research design. According to [DRL<sup>+</sup>05] co-operative method development research consists of three phases:

1. A qualitative study done by researchers investigates the work processes of the studied company. It is of interest to identify possible areas where improvements are needed.
2. In this phase possible improvements are designed by researchers as well as practitioners. Also quantitative measures are developed so that the improvement effort can be assessed.
3. In the final phase the improvements are implemented. The researchers observe the implementation using participant observation. After some time the improvement effort is assessed. The assessment results can be used as an input for phase two. In addition, researchers can use the results to analyse if a suggested improvement will work in reality.

It can be seen that researchers and practitioners work together to improve the actual software processes at a particular company but also to advance software engineering research by testing different improvement ideas. Dittrich et al. [DRL<sup>+</sup>05] conclude that this research approach helps to make software engineering visible. They also found that this research approach should be complemented with experiments. For example, possible improvements can be tested before they are introduced to minimise the risk. Co-operative method development also reveals why certain methods work or do not work in practice.

This master thesis comes very close to co-operative method development. The main difference is that I was not seen as a researcher within the Company. I was accepted as a co-worker participating in the day-to-day work. Also I was not testing a certain software practice and how it works out in a real setting. Instead I was observing how a particular company tries to improve their way of working without receiving guidance by a researcher or consultant.

### **3.5 Doing Ethnography in the Own Domain**

At the beginning ethnography was only done about “the others” [Wol99]. It took several decades that it was accepted to also study ones own domain. This might be a reason why so many years were needed to use ethnography in software engineering research as well.

So it seems to be appropriate to study software engineering as a software engineer using ethnography as I did. Sharp et al. [SR00] even go one step further and say it is an important requirement that software engineers start studying their own domain, because this is the only way for them to understand what they are actually doing [SR00].

In this context warnings raised by Forsythe [For99] must be taken into account. She says that often natural scientists use ethnography as yet another tool. The danger lies in the fact that only interpretations of what one has experienced are produced and not the actual views of the studied group [For99].

By keeping those warnings in mind and acting upon them, there is no reason why one should not use ethnography in software engineering research.

### **3.6 Own Influence**

A flaw in ethnography, which is often discussed, is the presence and influence of the ethnographer on the study. The ethnographer is working with the studied people. This also means that the studied people are influenced by him. For example, the ethnographer might have additional knowledge, which can be applied by the group as well. On the other hand, people might behave

different, because they feel observed. Of course the ethnographer can try to minimise the effect of his presence, still he can not eliminate it [Han00].

Hannabuss [Han00] discusses that the ethnographer is a part of the ethnography. Beside trying to minimise the ethnographer's influence, the ethnographer must be aware of his influence and reflect upon it. For example, autobiographical discussions can help to visualise the ethnographer's influence on the studied people [Han00].

Ethnography can be understood as a translation done by the ethnographer as pointed out by Churchill [Chu05]. This means, the ethnographer interprets what he is observing and experiencing and presents this to the reader. Even if it would be possible to eliminate the ethnographer's influence on the studied people, still the ethnography would be based on his translation [Chu05].

Harrington [Har02] has another opinion on this issue. He analyses an ethnography done by another author and shows that the ethnographer was able to reveal major insights by consciously influencing the studied people. This means, the own influence can also be used as a tool by the ethnographer and it does not have to be a drawback in any case.

Especially at the beginning of my study I was heavily concerned of how much I am allowed to influence the Company. In some improvement efforts I was the only one who had enough time to do the actual work. Even though other employees supported me, the major part of the work was done by myself. However, I found no evidence that the fact I was working on an improvement effort determined the outcome of this effort. Some of the improvement efforts I supported were successful, others were not. Also I am convinced that it would have been possible for other employees to do my work. It might have been not possible at the same moment, but sooner or later it would have happened. So I think that my presence accelerated some improvement efforts, but it did not changed the overall scene drastically.

To illustrate this point even more, I like to present another example from my field experiences. During my study one employee left the Company. Everybody considered him to do major work for the Company and to be an organisational hero. When he told the other employees he was going to leave, many people felt like everything is going to collapse. However, I did not noticed a major change in the Company after he left. His tasks were taken over by other employees and also a new employee was hired. Of course the work tasks of some employees changed and maybe also the work load of other employees increased, but still the Company was able to go on. To me this shows that the influence of a single person is often exaggerated.

### **3.7 Writing Ethnography**

Till now I have not answered another important question. At the beginning it was not clear how to present my ethnography. Of course a written report is required, but still there are many ways to write a report.

In quantitative research writing a paper is just the final part. Guidelines [Daw00, e. g.] exist, which describe the expected structure of a quantitative paper. In addition, many journals also have extra guidelines for authors.

Writing an ethnography is somehow different. Different authors like Hammersley et al. [HA00] point out that writing the ethnographic report is an essential part of ethnography. Because of being able to write an idea or thought down means that we have thought about it extensively [EFS95]. Once a teacher gave me the advice that you have only understood those things you are able to teach others. This means, only thoughts you are able to explain in your own words are thoughts you actually understand.

In this context it must be also questioned, if an ethnography is finished when it was written down. Still, someone has to read it. Based on the constructivism view, this means someone has to construct his understanding of the study I constructed based on my understanding of others constructed world view [Tal04]. This means, "the representation or reconstruction of a social world depends on how we write" [HA00, p. 245], but it also depends on how the reader is working

with the representation [Tal04]. Now it is getting clear that not just the ethnographer is responsible for the ethnography, but also the reader. The ethnographer can only present his construction of the world he studied. By providing a “thick description” the reader can create his own understanding of the study [Tal04]. So it is not the aim of an ethnographic report to answer questions, but instead to give the reader a possible construction based on the field material collected. This also means, the reader can not just consume the report, but basically he is getting a part of the ethnographic study [Tal04].

Another quite philosophical aspect connected to the nature of qualitative research is the question, whether fictional elements can be used in the written account. Based on the assumption that everything is a construction, one can also argue to use fictional elements. One possibility would be to use elements of prejudices about the studied group. By repeating those prejudices the reader can easily get into the scene and the whole ethnographic account is looking more authentic [HA00]. For example to describe the core programmers at the Company I could write that they are hackers in sandals wearing T-shirts and shorts. However, this would not be true, even though every reader would be able to imagine how those employees look like. I do not think that it helps to consciously use fictional elements in the written account. Of course it is impossible to describe reality, because the reality I experienced is just my own construction. Still, I think the world we live in is already thrilling enough, so that we do not have to enhance descriptions about the world with fictional elements. Therefore, I tried my best to avoid any fictional elements in my descriptions. Nevertheless, the reader has to keep in mind that “qualitative inquiry is fiction, in the sense that it is made or constructed, but not in the sense that it is pure invention, lies, or imaginings” [Tal04, p. 2].

Another thing which must be taken into account while writing an ethnography is that the studied people might read it [HA00]. In former times this was something ethnographers did not have to think about. In contrast to those old days I am pretty sure that at least some employees of the Company will read those lines I am currently writing. From my own point of view I want to prevent that those people say I am a liar or that I am just presenting those things supporting my assumptions and theses. This puts constraints on the way I am writing my ethnography.

Beside those philosophical aspects of writing an ethnographic report, there are also more practical topics to be discussed. For example, I had to decide which writing style to use. Van Maanen [VM00] has identified three main writing styles shown in figure 3.2 on page 28.

*Realist tales* [VM00, see] are written in a third person narrator. This is the style an ordinary movie is done. The reader can see how the different people interact and how they accomplish their tasks. For example by watching a hospital soap like *Emergency Room* from the 1990th, the viewer gets the feeling to completely understand the scene. The viewer knows all main characters and their personalities. Realist tales present many details and combined with their third person narrator create a feeling of authenticity. The ethnographer never appears on the scene, so the reader is not getting aware of that the presented tale is just the product of his ethnographic work. Realist tales are often used to exemplify a theory like a social interaction pattern [VM00].

Another ethnographic writing style are *confessional tales* [VM00]. These are classical first person narrations. The ethnographer is describing his experiences and how he acted in the field. He does neither try to hide his existence nor his influence on the field. Often he describes his ideas about the studied subject before the study was started and shows how his own ideas and thoughts developed during the study. This also includes a discussion why the selected group was chosen for the study [VM00]. Those reflections about ones own development is an autobiographical element, which is considered to be important [Han00]. It shows how the ethnographer develops an understanding of the studied field, how the ethnographer is developing a construction of his experiences. Therefore, the ethnographer’s thoughts are discussed [Han00].

Finally, one can also use *impressionist tales* as an ethnographic writing style [VM00]. This style is often embedded in theoretical discussions to illustrate the theory using a field experience. It helps to place the reader in the fieldwork situation. This means, the reader can understand how

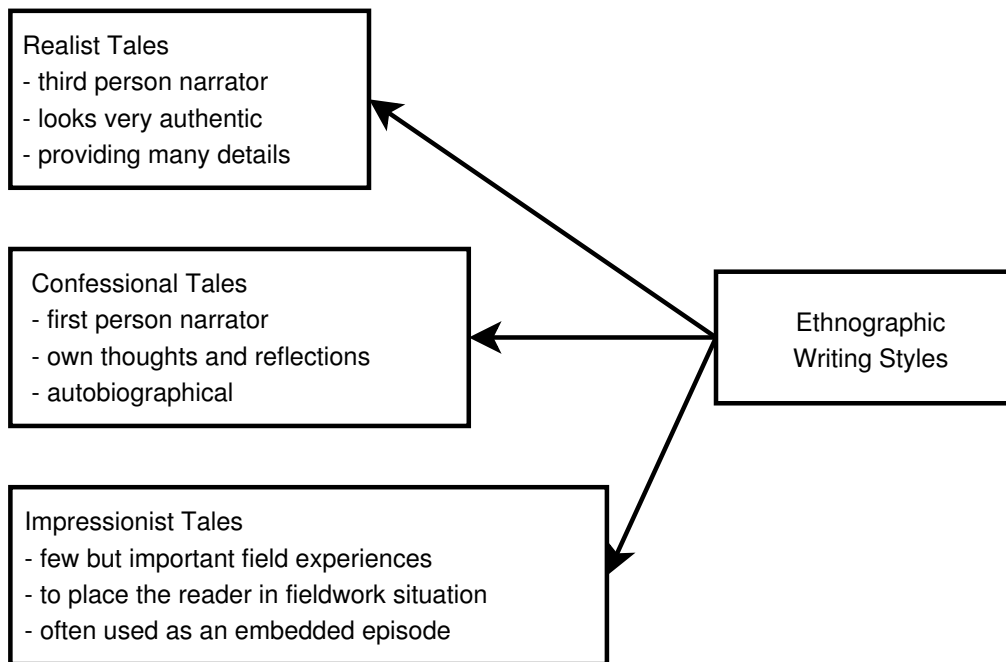


Figure 3.2: Ethnographic writing styles [VM00]

fieldwork is done and which problems must be dealt with.

Those styles are not fixed. It is possible to create another classification [HA00]. Still, I use this classification to define my own writing style. Most of the time I am using elements of the confessional tales style. I am writing as a first person narrator and I am definitely not hiding my presence at the scene. Furthermore, I am discussing my own thoughts about doing ethnography and also how my conception of ethnography has changed during the study. Later in this chapter I also provide a personal account why I try to answer the research question of this study. Still, I am also using some elements of the impressionist tales style. At different points I am using small episodes from the field to provide an example to better explain an idea or theoretical issue. This should also help the reader to examine whether this kind of study is interesting to do for himself.

Beside selecting a writing style, one also has to plan and organise the writing itself. An important point to be kept in mind is that the fieldnote should always be the starting point for anything written [EFS95]. However, it is not enough to just present the fieldnotes, because often fieldnotes only make sense for the one who has taken them. Therefore, context information must be provided to the reader, as it can be seen as an interpretation of the existing fieldnotes [EFS95]. To come up with a good argument or to present recurring patterns it is also important to categorise similar fieldnotes and to describe them together. Another important task before starting to write is to define the intended audience [VM00]. For me it was always useful to think of someone specific while writing. There might be different techniques available for defining an audience, but one good technique is to write down a short characterisation of the core readers [VM00].

Van Maanen [VM00] also points out another important issue. If an ethnography is created about a group of people which was studied before, it is important to “correct, specify, and elaborate on what is already known” [VM00]. As it will be shown in the next chapter, software process improvement efforts have already been studied at least by quantitative studies extensively. In my conclusions at the end of this thesis I will point out where I experienced differences and short comings of the existing theories.



## 3.8 Ethnography Applied

In this section I present how I did ethnography. Before I started I read Fetterman's [Fet98] step by step introduction to ethnography to get an overview and an idea about how to approach the field. Still, this guide is just one of many descriptions about ethnography. Actually, ethnography is a broad term used for many different kinds of studies [Har00]. Therefore, I do not claim that my instantiation of ethnography is the only valid one. There are many ways to do an ethnography. For example, in the before mentioned ethnography for requirements elicitation done by Sommerville [SRS<sup>+</sup>92] sociologists conducted the ethnography in the air-traffic control room. It was not the software engineers collecting the field data directly. Instead, the software engineers just used the data collected. A recent study done by Rönkkö et al. [RDR05] can be seen as an ethnomethodological informed ethnography. The authors observed a group of software engineers through interviews, hanging around in the office, focus groups, casual conversations and analysing work artefacts. They used the gathered data to investigate how plans are used in software projects. The chosen way of doing ethnography depends heavily on the ethnographer's personality. For example, a shy ethnographer will act differently than an ethnographer used to communicate with an unknown group of people.

### 3.8.1 Ethnographic Question

Before I started working on this master thesis I had to write a master proposal. This proposal was approved by the university and an adviser was assigned. In such a proposal one has to define the research aim and goals of the study going to be conducted [Cre02, e. g.]. Also one has to explain why a certain research design is chosen. This is of course also valid for ethnographic studies [Fet98].

Lethbridge et al. [LSS05] present in their paper different data collection methods for field studies. They explain that based on the research question, data required to answer this question must be identified. Afterwards, one can decide which data collection method is appropriate to acquire the needed data. Therefore, designing research follows different steps according to Lethbridge et al. [LSS05]:

1. Define the research question.
2. Identify data needed to answer the research question.
3. Evaluate which data collection method is best suited to collect the necessary data.

It is important not to mix the order of the different steps [LSS05]. Also in many cases just one data collection method is not enough. Therefore, different data collection methods are combined, which also helps to triangulate the results [LSS05].

In case of this study, a simplified version of the research question might read: "How does a small software company improve their software processes to get more productive and produce better quality?" There are different ways to answer this question. For example, one way would be to collect data from many different companies and to present a general answer. However, I did not want to provide a general answer but take a close look at one company instead. To be able to answer the question for one particular company I needed very detailed records about the chosen company, the work processes, the improvement efforts, etc. Based on this definition I decided to do an ethnography, which combines different data collection methods. Taking a look at the different data collection methods described by Lethbridge et al. [LSS05] it can be seen that some of them are part of ethnography (as described by Fetterman [Fet98]):

- Interviews
- Work Diaries

- Participant Observation
- Fly on the Wall (observation without active participation)
- Analysis of produced artefacts and used tools

By combining those different data collection methods it is possible to collect the necessary data to answer the research question above. It is important to keep the original research question in mind, because just collecting every data possible might also not be of great help. Wolcott [Wol99] says: “We do not and cannot simply *observe*, *watch*, or *look*; we must observe, watch, or look at *something*” [Wol99, p. 70]. This means, just doing an ethnography only for the own sake is not enough. There must be a purpose why ethnography is chosen. On the other hand, that does not mean to already have a hypothetical answer to the research question –a hypothesis– while approaching the field [LSS05].

From the discussion in this subsection so far it can be understood, why I have chosen ethnography as a research method. However, it is not yet clear, why I want to answer the above research question at all. How did I came up with this research question?

There are two main reasons why I was interested in answering this particular research question. First I am quite interested in the software process improvement field. I reckon that it is an often misunderstood topic. For example, before I started studying this topic I thought it was just about creating a formal documented process turning an organisation into a bureaucratic inflexible one. I had the feeling that formalism destroys creativity. However, after I had a deeper look into the SPI literature I understood that this nightmare I was thinking of is only turning to reality, if software process improvement is done wrong. On the other hand, there are many good reasons to have a look at all those different quality models, success factors, improvement strategies, process models, etc., because they incorporate many lessons learnt and good advises. Not applying SPI for various reasons seems today to me as a lame excuse for not being willing to change the own way of working and thinking.

The second reason is a little bit more complicated. I already knew the Company from my practical and I knew that they have problems with their software processes. I also experienced in my practical that they have problems improving the way they work. Still, the reasons for this inability to improve were not clear to me. I wanted to understand. I was convinced and I am still convinced that the people working at the Company are intelligent and capable. So the reason for not improving can not just be missing capabilities, there must be something else. It is the aim of this study to identify and understand this “something else”.

### 3.8.2 Accessing the Field

Before one can do an ethnography of course a field for doing the fieldwork is required. Already four months before starting my study I contacted the Company and asked if they would be interested in such a study. I emphasised that I can help them during the study to review and improve their software processes by applying some of the knowledge I have gathered during my studies. I also talked to the project leader in Halle I was going to work with and we discussed possible improvement areas. Based on this phone call I formulated the research proposal, which was accepted by the university. I also sent the proposal to the Company. It can be said that this particular project leader was my gatekeeper. If I would have been just an unknown student sending my proposal to the company bosses, the chance to get access would have been rather low. So I used this gatekeeper to get access to the Company.

At the end of March 2005 I returned to Halle and I visited the Company some days later at the beginning of April. I have already described this first visit in the introduction. Some days later I had a long discussion with the project leader on how to proceed. Basically, we defined that at the beginning we should focus on the PPS development process. At that day I also talked to several

employees I already knew from my practical two years ago. I explained to them what I was going to do the next several months in the Company.

The next day a big project leader day meeting happened in Halle, which was a fortunate coincidence for me. At this meeting almost all project leaders, many sales persons, and other employees gathered in Halle. One of the bosses introduced me and another student to the group. In addition, I also explained in some short sentences what I was going to do. At this point I do not think that most people understood what I was going to do. But at least they had heard my name and remembered me when I was mailing or phoning them later.

This day I took a copy of all available product descriptions home with me. I studied the various documents in detail at the weekend to catch up with what had changed since I left the Company two years ago. While reading the documents I also marked parts where I found some mistakes like spelling errors or misleading descriptions. During the following week I emailed the mistakes I had found to the different employees responsible for the documents. This proved to be a very valuable work for the Company as well as for my study. The employees recognised that I was able to really support them and that I do not just produce additional work for them with my study. It was also a good way to get into contact with those people I had not known before.

During my study it proved to be difficult to get into contact to all people I had not directly worked with. For example, it was quite impossible to access the SCM team members. Basically, I just got some basic access to the SCM team leader. Also it was not easy to get access to all employees in Halle, even though I was working in Halle most of the time. Some employees were only at the office on Monday and Friday. All the other time they were working in the customer's office and so they were not accessible for me. For example, it is not possible to ask direct questions about their projects, if they hardly know you. This always feels like spying. Therefore, it takes some time till people get used to the fieldworker and casual conversations are established.

Harper [Har00] points out that some time is always needed to get accepted by the studied people. After this period of time the ethnographer is considered as a part of the organisation, he is considered as an insider. The time needed to understand an organisation and to be accepted as an insider often differs [Har00]. I had the advantage that I already knew the Company from my five months practical I had done. People remembered me and I was already familiar with many rituals and work procedures in the Company.

### **3.8.3 The Fieldwork**

At the beginning I worked fulltime in the Company for about two and a half months. Later I worked halftime at the Company again for two and a half months. However, if there was a meeting or I was travelling to the other company location in Chemnitz, I spent more time at the Company. Almost all the time of my study I was located together with the project leader in his office. Both of us had our own desk and PC. My PC was connected to the Company's Intranet as well as to the Internet. I was able to access all internal servers and also parts of the Italian Intranet. In addition, I had an official company email address and people were able to call me by the project leader's phone. The project leader's office is located in the middle of the office in Halle. However, it would have been better to sit near the secretary and boss office, because there spontaneous talks between the employees happened sometimes. Still, the office location was good enough to observe what was going on.

If there was nothing to do at the Company, I used the time to read research papers at my office PC. However, I noticed that it is quite hard to concentrate in an office. This was one reason that I decided to work only halftime after some time. In addition, I had to read many books especially about ethnography. I did not want to do this in the office, because I think this would have made a strange impression to the other employees.

During my work at the Company I tried to participate in as many improvement efforts as possible. I tried to achieve this for example by offering my help.

Unfortunately, there were also times where nothing interesting could be observed in the Company. In these hours and days I had to learn that waiting is also an essential part of ethnography.

### **3.8.4 Collecting Fieldnotes**

When I was in the Company I took fieldnotes. I wrote the fieldnotes down on paper. The fieldnotes look like a long list of notes. The notes are in most cases not longer than one or two lines. I took the field notes on paper for several reasons. Pencil and paper is available everywhere and it can be used in almost every situation. Also it is an accepted working instrument, so employees do not feel observed if fieldnotes are taken while speaking to them. Also pencil and paper provide the possibility to link different statements easily with lines and circles. It might have been also possible to use electronic documents to write down fieldnotes, but for example at the meetings or in the car rides between both company locations it would have been impossible to take notes.

For me it was obvious from the beginning that it is impossible to use a tape recorder. In some contexts and situations it might be possible to use a tape recorder, but this seems to depend heavily on the culture of the studied organisation. In Germany people have a feeling of getting spied if someone is recording them. A reason for this might be that the Company is located in Eastern Germany, which was the former German Democratic Republic (GDR) till 1991. In the GDR everybody was observed and spied by several different national agencies (e. g. Stasi) and also freedom of speech was not given. The majority of the Company's employees grew up in the GDR. Another problem with tape recording is that it hinders developing a good casual conversation. I am still convinced that a tape recorder would have hindered more than it would have helped. Of course it would have been nice to have a tape recording of the formal interviews I did, but it was also possible to do it without.

When I was at home I reviewed my fieldnotes and added missing things I remembered. The quality of my fieldnotes improved by time. When I started the study I had not yet studied ethnography and fieldwork topics extensively, because of several time constraints before. In this way I learnt taking fieldnotes by doing and also by making mistakes. For example, at the beginning I often just noted the interpretation of an observed action and not the observation itself. Therefore, I might have missed some things at the beginning of my study. However, I tried to overcome this problem later by re-asking some of the questions I already raised at the beginning.

Beside taking fieldnotes, I also examined the available Intranet resources extensively. Often I found documents not many employees were aware of their existence. I tried to trace the documents back to their source and ask the author additional questions about the documents.

Another important way to understand the Company is to continuously compare the culture at both company locations. As explained in the previous chapter there are many differences between both locations. Also the work done in both locations is different. For example, coding only happens in Chemnitz in the PPS and SCM development teams. This comparison of both locations revealed many interesting insights.

I also did some formal interviews. I prepared those interviews in advance extensively. First of all I studied in what tasks the interviewee was involved, what he had done in the past and which work artefacts he had created. I also tried to identify those employees the interviewee has to work with most of the time. Based on those investigations I prepared about 15 questions. At the beginning of each interview I explained the aim of the interview and I assured that the content of the interview will stay confidential and that it can not be traced back to the interviewee. Normally, I started the interview with some basic questions about the history of the Company like when the interviewee has joined the Company and about his tasks at this time. Mostly I also asked for a comparison of the tasks at this time and today. This always proved to be a good question, because the interviewee had to start to reflect on what he was doing. The question could not be answered by just listing facts. Other questions investigated the work relationship between the interviewee and other employees. Here, I was often interested how responsibility and tasks are shared among the employees. Usually, I finished the interview with the question of what the interviewee would

like to change in the Company, if there were no constraints to take into account. Even though this might seem like a ridiculous question, it turned out to inspire interesting answers with many new insights.

From this short overview of my fieldwork it can be seen that I used all three ethnographic elements (to experience, to enquire, to examine) as described in section 3.1.1 on page 20. Most of the time I did participant observation, which means that I worked in the Company together with different employees *experiencing* how they do and understand software process improvements. During this participation of course I also did many casual conversations. I used those casual conversations to *enquire* questions I was interested in, but in most cases I just tried to listen what the people had to say. The same is true for the formal interviews I did. Sometimes the interviewee misunderstood my questions and started to talk about something I had not expected. Still, I did not interrupt them. For example, I asked several employees how they think the Company is organised and what they do to organise the Company. In most cases the interviewee answered how the Company is administrated, but not how it is organised. I noticed that in German the word “to organise” is also used to describe how daily administrative tasks are done, even though there is a special word in German for “to administrate”. One interviewee also misunderstood the question about company structure and organisation in a complete different way. He thought I was asking if a works committee or some kind of labour union is needed in the Company. Based on this understanding he explained how he felt about the relationship between employees and bosses. He concluded that such organisations are not needed in the Company, because the Company is not trying to exploit its employees. This was of course something I had not expected, still it was interesting to hear his view on the company culture. Beside experiencing and actively enquiring, I also *examined* artefacts produced by the Company. Most of the artefacts produced by the Company are stored on a file server, so it was easy to get access to those documents.

### 3.8.5 Data Analysis

At the end I ended up with several thousand words of field notes, with about 20 important email threads, and several electronic documents which all needed to be analysed. In a first step I transcribed all hand written notes into electronic text files. If there was a drawing in the fieldnotes like a diagram, I described the drawing in the transcription and provided a reference to the belonging fieldnote. I transformed all electronic documents and all important emails into text files as well. Transcribing the hand written notes was a good step, because I re-read all the notes I had taken.

By reviewing the literature I decided to do something like a template analysis as described by King [Kin98]. The basic idea is to develop a hierarchy of codes. These codes summarise issues, which occurred in the study. This code network is called template. In a second step the fieldnotes are coded with the template. That means, passages in the fieldnotes are marked and a code is assigned to the marked section. Not in all cases a fitting code will be found for a section. Therefore, the code network must be changed. This shows that even the analysis process is a process of continuous self reflection and interpretation. The data is categorised by assigning codes, restructuring the code network also changes the categorisation.

One code I assigned is called “access”. I used this code to mark all section in my fieldnotes that describe how I got access to the studied group of people. Another code is “anxiety”. I tried to mark all section with this code, where I expressed some kind of uncertainty how to proceed in my study. However, I noticed that a single code to describe different feelings and emotions is not enough. Therefore, I split the “anxiety” code into several sub-codes. The parent code is now called “feeling” and it has several sub-codes like “anxiety”, “demotivation”, and “uncertain”. This process of re-codification and changing the code network is illustrated in figure 3.3 on page 34.

To do the coding of the fieldnotes I used the AnSWR software package. AnSWR is spon-

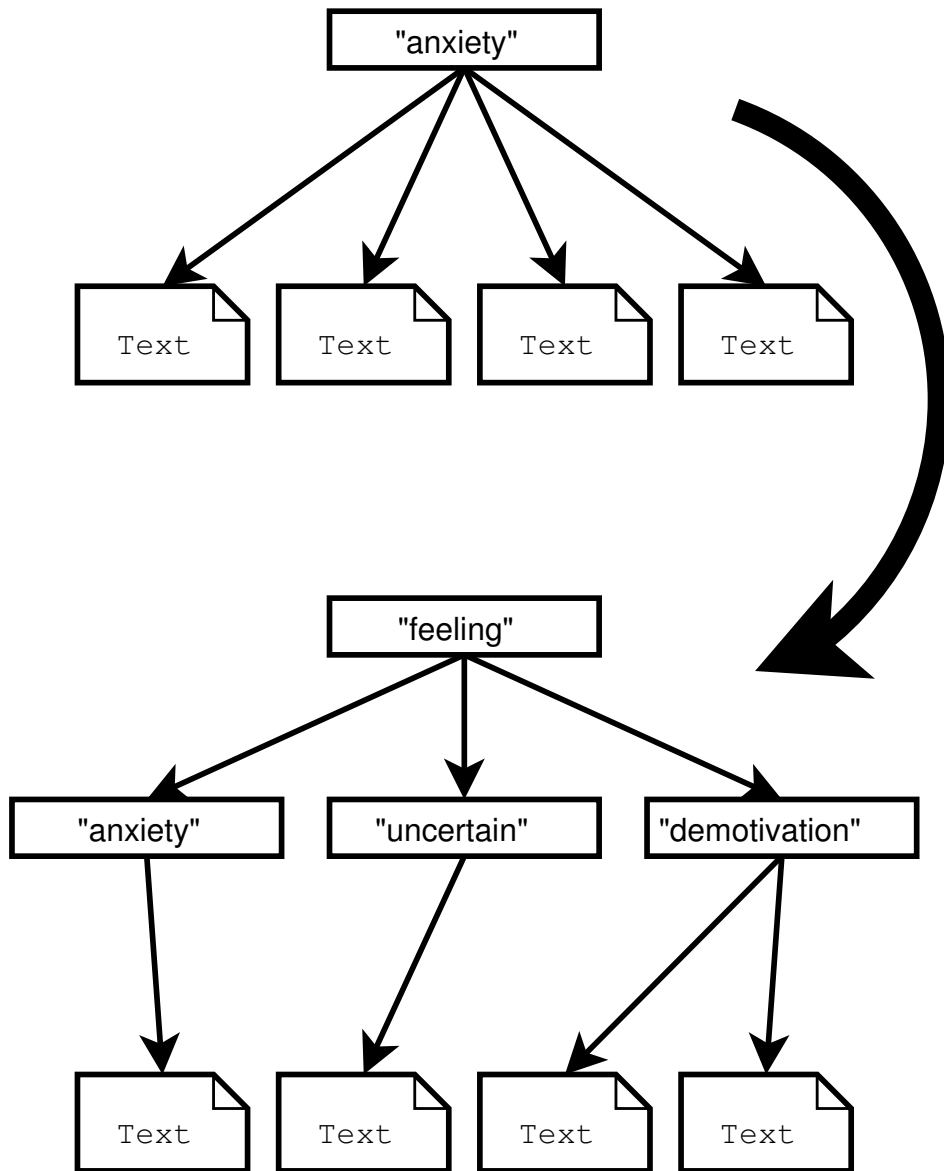


Figure 3.3: Development of a code network

sored by the US Department of Health and Human Services<sup>4</sup>. The software package is developed by the divisions for HIV/AIDS prevention and an executable version is available for free from the AnSWR's Internet page<sup>5</sup>. I used version 6.4.177 for Microsoft Windows 98, because the latest version of AnSWR does not work on a current German Microsoft Windows XP installation. AnSWR is not bound to a specific analysis method, but the basic suggested method is very similar to template analysis [CMO96].

AnSWR provides the functionality to create a code hierarchy and a source network. Before coding can start, all text files must be loaded into AnSWR. During coding text parts are highlighted and marked as a segment by clicking on a button. After, codes and sources can be assigned to each text segment. The number of codes and sources for one text segment is not limited. In addition, small notes can be added to each text segment.

The first version of my code network was based on software process improvements success factors found in literature. In addition, I used the structure of my formal interviews to identify even more codes. Then I started coding my fieldnotes, starting with the oldest fieldnotes. While coding I reworked my code network several times as described in template analysis [Kin98]. After I finished coding all text files I structured all available codes in a hierarchy. Coding all my fieldnotes and developing the code network took about two weeks of fulltime work.

The AnSWR software package provides many reporting functionalities. However, this functionality is not very well documented and I found no way to use this functionality for my purpose. Therefore, I first investigated AnSWR's database structure. After identifying the interesting relationships I wrote some small programs to put out lists with the data required by myself. Besides, a list with all coded text segments plus applied codes and notes I also generated a list with all codes plus the text segments they were assigned to. With help of both lists it is possible for me to look up in both directions the relationship between code and fieldnote.

It was very interesting while analysing my data, how several ideas about the study developed. For example, it is possible to identify common patterns in the data as it will be discussed later. Also it is interesting to see how the fieldnotes have changed over time. In some way it is like experiencing the whole study again.

It would have been possible to go on analysing the data using different methods. For example, I was also thinking of visualising the connection between different codes. Still, I decided to stop because of given time constraints, even though "we know our analysis is not finished, only over" [VM00, p. 120].

### 3.8.6 Self Reflection

When I started this study I tried to follow the guidelines given for ethnography [Fet98, e. g.]. I tried to improve my methods over the time by learning from my mistakes. I tried to challenge myself by thinking about the limitations of my study.

I often reflected on what I was doing and what I was observing to identify problems in the way I was doing ethnography and also to enhance my understanding of the studied organisation. Continuous reflection is seen as an important part within software engineering [HT05], because it is a way to become aware of the complexity involved but also of the important role communication plays. A software engineer may for example think about his basic assumptions, about how decisions are made, about how problems are solved and how models are used [HT05]. The same is of course true for any action – also for ethnography.

I often tried to reflect what I was actually doing and to improve the way I was doing it. For example, during the study I improved my ability to take fieldnotes, but I also learnt that it is not always possible to ask direct questions in casual situation. I once was asking an employee about the outcome of a meeting I had not participated in. The employee refused the answer by saying

---

<sup>4</sup><http://www.cdc.gov/>

<sup>5</sup><http://www.cdc.gov/hiv/software/answr.htm>

that I do not have to know everything. Maybe he had a bad mood at this day or for whatever reason, but to ask that directly by myself was too obtrusive in this situation. Later the same day I listened to other conversations and also talked to other people and in the end I found out what the meeting was about.

Doing an ethnography is not always an easy task. During my study I was wondering several times if my work has any value at all. On one hand, my amount of time I could spent to help the Company improve their processes was limited by my additional scientific tasks like reviewing literature. On the other hand, I did not always feel that I was able to generate something new. Furthermore, I also was concerned a lot about the validity of my study. How could a single study at a single specific company produce anything meaningful? Also I was concerned a lot whether I influenced the Company too much and that I was pushing the software process improvement efforts too much. On the other hand, are those small improvement steps made by the Company worth to be reported at all?

Cooper and Wodgar [CW93] emphasise that it is not enough to just do an ethnographic study. They say the ethnographer also has to reflect his work. This also means, studying the theory and philosophy ethnography is based on. I try to provide this reflection all over in this report. First I was thinking to devote an own section about everything I have experienced while using the research method ethnography. Later I decided that the reflections will be decontextualised if they are not presented in those contexts they were created by myself. Nevertheless, I have to confess that I did not started to reflect from the beginning of my study. This was also something which I had to learn. Maybe this awareness of being able to reflect about what I am doing is the most important outcome for myself.



## Chapter 4

# Software Process Improvements

At this point a short overview of software process and software process improvements is given. This is not an extensive discussion, because those discussions are already done in literature. Therefore, I focus on those points which later relate to the actual field experiences and to those points I like to challenge by my field experiences.

### 4.1 Theory

Software is the product of the software producer. This can be either a single person but probably most software is produced by a group of people working together. This group can be organised either in a company or people form a group themselves as it can be seen in the OpenSource and Free Software community.

To create a software several steps are required [Som04]. From a technical point of view source code must be written first. This source code must be compiled and the compilation is executed afterwards. If bugs are found, those bugs get fixed in the source code and the software is recompiled. Even though this is a very simple view on software creation it can already be seen that software development can be described as a process - a software process. Sommerville [Som04, p. 64] defines the software process as “a set of activities that leads to the production of a software product”. He identifies *software specification*, *software design and implementation*, *software validation*, and *software evolution* as the main phases available in most software processes [Som04].

The software engineering community has already identified and described various software processes like the Spiral Model [Boe88], the Rational Unified Process [Kru00] and agile processes like Extreme Programming [Bec99]. Those process descriptions consist of activities and artefacts produced by an activity. Before an activity gets executed conditions might exist, which must be fulfilled in advance.

This process idea is not new and definitely not invented by software engineering [Zah98]. For example, it was already used by car production industry for many years to improve car production [BK03]. It was shown that an improved production process also improves the quality of the final product – the car. This is the main point of all process improvement efforts: the quality of a product is related to the quality of the production process [BK03]. This means for software engineering, if software processes are improved, also the software quality will be improved. This positive correlation between process quality and product quality within software engineering is already documented in literature [GH95, e. g.].

At this point I do not want to discuss the term “quality” in detail. This has been done extensively already [BK03, e. g.]. At this point it is just important to keep in mind that high quality software does not just mean bug free software systems. There are additional components like that the software must fulfil user requirements, must be easy to use, must be maintainable, must be high-performance, etc [BK03].

## 4.2 Continuous Improvements

One of the main ideas of process improvements is not just to improve the process once, but to continuously optimise and improve the processes [BK03]. Therefore, an organisation will never have a final process, but instead change the process regularly. In this context it seems obvious that software processes given in literature can only be used as templates and must be adapted by the organisation implementing them [Zah98].

To improve the process the so called Deming cycle [BK03, e. g.] can be used. This cycle describes different phases. The first phase is to plan what should be improved. In the second phase actions are taken to do the improvement. In the third phase the improvement is evaluated. In the fourth and final phase actions are taken based on the results of the evaluation in the third phase. The Deming cycle seems to be a very natural way to improve the way a process is working. For example, a human child also learns how to perform actions like speaking step by step. Learning to speak is a continuous process of trial, error, planning, checking and practising. Therefore, it seems obvious to apply this idea in software engineering as well. However, it must be mentioned at this point that this is not the only theory on how to improve processes. The theory of business process reengineering [Ham90, e. g.] for example suggests to design a faulty process completely new and implement it at once. Here, the emphasis is not on continuous improvement but instead on doing it right the first time.

The continuous improvement idea has been applied in software engineering [Zah98, e. g.] and might be summarised in the term *software process improvement*. Here, the same idea as in the Deming cycle is applied within software engineering. A software process is monitored for possible improvements. The improvements are planned and implemented. Afterwards, the improvements are evaluated. Different software process improvement models exist and are summarised by the term *quality models* in general [Zah98].

## 4.3 Quality Models

In our daily life we are already used to quality models. For example, food is checked by governmental agencies to ensure high quality. There are also several high hygiene standards, which must be followed in the food production industry. Another example are cars. Cars must be regularly checked to ensure they work as expected. This prevents dangerous situations, which in the end could even harm human life.

In both examples, the product is checked and not the production process. This shift from assessing the production process instead of its outcome came up at the end of the 1980th [Zah98]. Today it is the base of the different quality models.

Another common feature of quality models is assessment [Zah98]. Assessment means to measure the maturity of a process. The quality models define the points a process must fulfil to pass the assessment. Here, it is important to assess how the process is actually implemented by the employees and not how it is meant to work. Depending on the quality model an assessment can be done by the company itself or it can only be done by external certified auditors.

Unfortunately, there are many different quality models available and an overview is getting quite confusing quickly [She97]. Therefore, only two models are discussed in more detail.

### 4.3.1 ISO 9000:2000 and TickIT

The historic roots of the ISO 9000:2000 standards go back till the 1930th [She97]. At this time the British military already defined standards for its suppliers. Those standards were developed and also later adopted by the NATO [She97].

The ISO 9000:2000 standards consist of the ISO standard 9000, 9001, and 9004, all released in the year 2000. Previous versions of this standard focused on the quality of products. The new

version defined the characteristics of a management system leading to high quality [Zah98]. The purpose of the 3 different standards is the following:

- ISO 9000:2000 defines the 8 fundamental quality management principles, and important terms and concepts
- ISO 9001:2000 defines the requirements for a quality management system
- ISO 9004:2000 defines a guideline for implementing ISO 9001:2000

Those standards are domain independent. To implement ISO 9000:2000 a company has to define an own ISO 9000:2000 system. To facilitate this task for software development companies, the TickIT [Ins50] guide was developed. It comments each ISO 9001 requirement with hints what it can mean for a software development process. Nevertheless, TickIT does not define any concrete processes, it is not a methodology.

The ISO 9001 standard is the most important one, since it states the actual requirements for a company's quality management system [Zah98]. It is possible to ignore certain requirements, if the company does not need such processes at all. To exclude a requirement a rationale must be given. ISO 9001 is divided into 4 main areas: Management responsibility, resource management, product realisation, and measurement, analysis and improvement [Zah98]. Each of those areas is further divided by several points.

For a company it is possible to become ISO 9000:2000 certified. Those certificates are often needed in Europe to bid for contracts by government agencies. An ISO 9000:2000 assessment can only be done by an external party. The result of the assessment is either the company passed or failed the assessment. The assessment is repeated every 6 months, otherwise the company loses the certificate [Zah98]. This means that companies are forced to really implement the ISO 9000:2000 requirements and not only for the assessment.

The ISO 9000:2000 standards are well known in Europe and widely accepted. Having an ISO 9000:2000 certificate is good for marketing, because other companies can expect a company committed to quality by having this certificate. If a supplier already has an ISO 9000:2000 certificate, an own assessment can be omitted. This saves money, because no additional assessment must be done [Zah98].

In former times it was a problem for software companies to get assessed for ISO 9000:2000, because the auditor did not understand software development processes very well [Ins50]. At this time most assessments were just done in production and telecommunication industry. Therefore, the TickIT framework was created. It describes in detail how ISO 9000:2000 can be implemented in an IT organisation. Also auditors assessing TickIT are trained in the special needs of IT companies. This means that improvement advices by the auditor are often more meaningful for the software companies.

The disadvantage of ISO 9000:2000 is that the requirements are quite abstract. This means, much effort is needed by a company to define a process according to the ISO standard [Rus04]. For example, one requirement is "establish your quality system"<sup>1</sup>. But what does it mean exactly?

Another problem of ISO 9000:2000 is that a company is not forced to go on improving their processes after receiving the certificate. Also today many companies have an ISO 9000:2000 certificate, but still the degree of quality focus and process maturity is quite different between those companies. The problem is that there are no levels of maturity in the ISO 9000:2000 model to compare the process maturity of different companies.

### **4.3.2 Capability Maturity Model Integrated (CMMI)**

The roots<sup>2</sup> of the Capability Maturity Model Integrated (CMMI) go back till the late 1980th. The Software Engineering Institute (SEI) at Carnegie Mellon University Pittsburgh, USA was founded

<sup>1</sup>This is requirement 4.1 in ISO 9001:2000.

<sup>2</sup>This history overview is partly based on [http://www.tantara.ab.ca/a\\_isorel.htm](http://www.tantara.ab.ca/a_isorel.htm)

by the US Department of Defence. The vision of this institute is to “help others make measured improvements in their software engineering capabilities” [Ins25].

At this time suppliers of the US Department of Defence had a lot of trouble to release good software within budget and time. Therefore, it is the task of the Software Engineering Institute to develop methods and tools to help those suppliers improving their processes. In 1991 the first version of the Capability Maturity Model for Software Engineering (SW-CMM) was released. During the 1990th several other CMMs were developed for different domains like Systems Engineering (SE-CMM) and Integrated Product Development (IPD-CMM) [Tea02b]. The US Department of Defence forced suppliers to implement those models, otherwise contracts were not given to them. It turned out that it was a complex task for the suppliers to choose the correct CMM. Often suppliers worked in several domains and they had to apply therefore several CMMs [Tea02b]. A research project was started at the Software Engineering Institute to integrate the CMMs into one general model. This new model is called CMMI (Capability Maturity Model Integration). The corresponding project was started in 1997 and in 2000 a first version of CMMI was released. Today the latest version is CMMI version 1.1, which was released in August 2002.

CMMI is a process improvement model. It describes the characteristics of good processes for software development, but it does not prescribe how to actually do the processes [Tea02b]. The CMMI is available in four different versions for the domains Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing [Tea02b]. However, the different versions are very similar, just different terms depending on the domain are used.

The characteristics for good processes are divided into 22 process areas. Typical process areas are Risk Management, Configuration Management, and Project Planning. Each of these process areas are divided into specific goals. For example, specific goal 1 for the process area Requirements Management says requirements are managed and inconsistencies between requirements document and current work are identified [Tea02b, p. 82ff]. Each specific goal is supported by specific practices. In case of the above mentioned specific goal the specific practice 1.3 says that requirement changes should be managed. Beside those specific goals also generic goals exist. Those goals repeat in the different process areas and are therefore called generic.

A company applying the CMMI uses the CMMI as a guide or framework of how to improve the company’s own processes. Of course it is interesting for a company to know how much of the CMMI they have implemented already. Therefore, a company can be assessed using an appraisal method. There are two different representations available to measure the implementation level of the CMMI: staged and continuous. The staged CMMI model defines five levels: Initial, Managed, Defined, Quantitatively Managed, and Optimising [Tea02b, p. 10ff]. The overall term is maturity levels. To reach a maturity level, all process areas belonging to this maturity level and all maturity levels below must be fulfilled. In the continuous CMMI model the company selects those process areas most important for the company. The maturity of each process area is assessed. There are 6 levels to measure the maturity of a process area within a company: Incomplete, Performed, Managed, Defined, Quantitatively Managed, and Optimised [Tea02a, p. 13]. Those levels are called capability levels. So in the staged version the maturity level is a measure for the whole company and in the continuous version the capability level is a measure for a single process area. A company has to choose between staged and continuous CMMI.

One clear advantage of the CMMI is the fact that it was developed for the IT industry by a research organisation focusing on software engineering. Even though a company might not follow the CMMI, it still can use the CMMI to identify important processes and to get ideas of how to improve them. In addition, by using the continuous CMMI a company can focus on the most important processes. Also by providing different maturity levels it is possible to rate companies according to their process maturity and quality focus.

On the other hand, the CMMI is a very big standard of about 600 pages. Also the CMMI was developed for the US military and might not be applicable in smaller IT companies<sup>3</sup>. Also

---

<sup>3</sup>See next section for a detailed discussion.

in CMMI the assessment can be done within the company and no auditor certification is needed. This might lead to a misuse of CMMI for marketing purposes.

A very important point and great advantage compared to ISO 9000:2000 is the focus on continuous improvements. Those continuous improvements are also considered to be important in ISO 9000:2000, but there it is just one of many important points. Also the CMMI defines process areas, which must be defined by the company. In using ISO 9000:2000 a company has more freedom to choose the process areas to focus on. This might also mean that a company is missing important process areas.

However, a company applying one of both standards consequently, is on a good way to improve their processes. Still it is not easy to implement the standards. Therefore, different improvement strategies have been suggested.

## 4.4 Improvement Strategies

As shown in the section above, there are quality models available describing how a good software process should look like. Still it is a challenge for a company to initiate an improvement effort. Companies are especially missing guidance on how to do the improvements [GH95, NWZ05a]. Therefore, improvement strategies and frameworks were formulated.

The IDEAL<sup>4</sup> (initiating, diagnosing, establishing, acting, learning) model was created by the Software Engineering Institute at Carnegie Mellon University as well. This model supports introducing quality models like the CMMI. IDEAL describes five phases, which are divided into steps. The first phase is the initiating phase. Here, the whole improvement action is initiated by a stimulus for change. It is the aim of this phase to define the context and infrastructure for the improvement effort as well as to get some kind of support or sponsorship.

After the initial phase a repeating cycle is started. As the Deming cycle it also consists of four phases. First of all possible improvements must be identified and described in the diagnosing phase. Secondly, improvements must be planned in the establishing phase. Thirdly, the improvements must be implemented in the acting phase. Finally, it is important to reflect upon the improvement done and to learn for future efforts. This is done in the learning phase.

Another method to introduce a quality model is to use the SPICE standard (ISO 15504:1998). SPICE<sup>5</sup> was an international initiative to define a standard for software process assessment. By assessing the current state of a software process, the people doing the process get aware of possible improvements. In addition, the SPICE standard also includes guidance on how to do process improvements. Beside technical issues on how to use the results of an assessment for an improvement effort, SPICE also discusses how to deal with people issues and management issues during the improvement effort.

However, at this point it is not clear if those improvement strategies and the quality models can be applied in small software companies at all. Some authors like Nunes et al. [NC00] even claim that those models are not applicable in small companies. Conigliaro [Con01] also states that methods invented for big organisations can be even harmful for small organisations. This can happen for example, if a small flexible organisation is turned into a bureaucratic one. On the other hand, assessment methods for small organisations exist, e. g. the FAME model<sup>6</sup>. According to FAME's Internet page, the method can be tailored to the organisational needs and characteristics. This means to identify the strength and weaknesses and to structure the software process by finding a balance between formal and ad hoc processes [Con01]. Wilkie et al. [WMM05] point out that the continuous version of the CMMI is more applicable in small organisations, because the small organisation can spend its limited resources on the most striking problems. Focusing on the biggest problems helps the employees to see the benefit of an improvement effort.

---

<sup>4</sup>see also <http://www.sei.cmu.edu/ideal/>

<sup>5</sup>see for example <http://www.sqi.gu.edu.au/spice/>

<sup>6</sup><http://www.iese.fhg.de/FAME/>

Niazi et al. [NWZ05a] have suggested an Implementation Maturity Model (IMM). This model can be used to assess the state of an improvement process. It can also be used to improve the way a company is trying to improve the software processes. The IMM model incorporates many interesting ideas and hints about software process improvements. Since more than 10 years such hints are now collected as success factors for process improvement.

## 4.5 Success Factors

Beside explicit strategies, there are also various success factors for software process improvement. Success factors are more general hints helping to ensure the success of a software process improvement effort.

An early study about success factors is Goldenson and Herbsleb [GH95]. The paper is based on a survey sent to companies and people involved in CMM improvement efforts at the beginning of the 1990th. The survey data was analysed using quantitative methods. According to this study, the most important success factors are management commitment and compensated responsibilities [GH95]. Also quite important according to this study are that the people involved are well respected in their organisation, that technical staff is involved in the improvement process, that enough resources are given to the improvement project, and that process improvement goals are clearly stated and well understood in the organisation [GH95]. Beside identifying success factors Goldenson and Herbsleb also found some typical barriers like ongoing organisational politics, turf guarding, bad experiences from past improvement efforts, and the feeling that improvement efforts only create extra work without helping the employees to solve their work task [GH95].

Those success factors and barriers are confirmed by an Australian quantitative field study [CS01]. This study was done in small Australian companies using the SPICE assessment method. In addition, the study identifies a software process improvements mentor as a possible success factor. This means, someone is available to guide the organisation through the software process improvement effort. Also it is pointed out that companies should try to do the improvement effort in small steps not longer than five to six months. Another important point is to train the people in software process topics [CS01].

Beecham et al. [BHR03] approached the question of success factors differently. They asked if the kind of problems depends on the maturity level already achieved. Furthermore, they asked if different groups within an organisation like developers, project managers, and managers face different problems. They interviewed employees in groups of five to six people. Later they analysed the data to identify problem clusters. Those clusters were correlated for example to show if there is a connection between a group of people and a success factor. It can easily be seen from the study design that qualitative methods were used to collect data, but later the data was analysed using quantitative methods. This is one way according to Seaman [Sea99] how qualitative methods can be used within software engineering research. The study identified that the problems an organisation is confronted with depends significantly on the already achieved maturity level. Also different groups of employees face different problems, even though this correlation is not that significant [BHR03]. The most often mentioned barriers in this study are organisational ones like not having a good (process) change management, bad communication, cultural problems, misunderstood goals, and politics going on in general [BHR03].

Another study [Dyb05] shows that employee participation is important. Also company goals and improvement efforts must be aligned. This success factor is called business orientation in the study. The study also shows that improvements must be measured, otherwise it is impossible to judge the result of an improvement action. In addition, the study also emphasises that already existing knowledge in the organisation should be used for improvement. This is called exploitation of existing knowledge [Dyb05]. The paper is worth to study, because theories from organisational behaviour [Rob00, e. g.] are applied within software engineering.

In an experience report Guerrero and Eterovic [GE04] describe an improvement project in a

small software company with 80 employees. More than 50.000\$ were spent on this project, a big investment for a company of that size. They report that process related training for all employees, developers' involvement, continuous feedback, and visibility were the main success factors in this project [GE04]. Visibility means to show early results of the improvement efforts. This helps to convince employees that the improvement is useful for their work. It also ensures that the effort is not forgotten over time. This point is also stressed by Green et al. [GHWC05]. They report that improvements must not be easy to implement as long as positive results are visible and feedback is given.

In a recent study by Dangle et al. [DLSZ05] researchers acted as software process improvement consultants for a small software company introducing CMM. As success factors they identified alignment of improvement efforts and business goals, prioritising improvements, and testing changes in pilot projects. This study is of special interest, because the company of the study is very similar to the Company in this study. Both companies are of the same size and also both companies are lead by a technical manager (chief programmer) and a sales manager. Interestingly it was a challenge for the two managers in the study to accept a need for delegation of responsibility. The managers also had to understand that changes were needed to enable further growth of their company. Both problems will be a major theme in the remaining sections.

In another Study by Niazi et al. [NWZ05b] management commitment was also identified as a main success factor. An additional identified success factor is awareness. Awareness is meant in the way that the organisation recognises the improvement effort is a process by itself, which also can be improved. So an organisation is not just aware of software process improvements, but also knows that the way processes are improved can be improved as well [NWZ05b].

In summary, most studies agree on a basic set of success factors like management commitment, feedback, visibility, staff involvement, resources, and introduction strategies like IDEAL.

## 4.6 Summary

This chapter provided an overview of the theory behind software process improvements. As starting point the software process was defined and characterised. Afterwards, a brief discussion of the term quality was provided. Here, it was important to understand that bug free software does not necessarily have to be of high quality. Combining the ideas of software process and software quality means to say that the quality of a software is determined by the software process producing it. If this hypothesis is accepted, it must be the aim of a software company to steadily improve the quality of the software process. With the family of ISO 9000:2000 and with the CMMI, two prominent quality models were presented. Those quality models should be used by software companies to enable continuous software process improvements. Both presented quality models are complex and therefore improvement strategies like FAME exist to support the actual improvement process.

This theoretical overview is the base for this master thesis. The ideas of software process improvements, quality models, and improvement strategies were used during the fieldwork to understand how the studied company is doing software process improvements. The Company has no knowledge of CMMI nor ISO 9000:2000. Also they have no quality vision. Still they try to improve the way they produce their software products and services.

Another important discussion in this chapter is the presentation of success factors for software process improvements. Various researchers have investigated what enables and hinders software process improvements. They identified success factors like management commitment, employee involvement, improvement responsibility, and continuous feedback. On the other hand, they also found typical barriers like organisational issues or politics.

During my field study I paid special attention to the success factors. I checked if the success factors in literature also occurred in the field or if other factors were more important. A summary of the field experiences is presented in the next chapter.

## Chapter 5

# Software Process Improvements Applied

In this chapter I am going to present my field experiences. During my field work I have participated in several different process improvement efforts. Of course those efforts are connections among each other, but I decided to separate them. This way the reader may easily follow the explanations. As the presented improvement efforts are connected among each other, I point to those connections throughout the text.

### 5.1 Top-down Approach and Careful Analysis

At the beginning it was not clear where to start software process improvements. In a long discussion with the project leader I worked with, we came up with several possibilities. For example, he described his dissatisfaction with the end user support. The support team is not using a bug tracking system. Problem reports and bugs are stored in spreadsheets on their computers. There is no knowledge base with solutions for common problems available. Furthermore, the employees doing software tests do not cooperate with the support team. This means, they also have their separated lists with bug reports. Another problem mentioned by the project leader is that it is impossible to get the number of problem reports for a specific customer release. This information is not gathered by the support team. So it can easily be seen that many problems exist in the cooperation between the support team and other parts of the Company. However, the improvements and changes needed seemed too big for a first effort. In addition, I did not have any access to the support team yet. Also the support team is located in the other Company location in Chemnitz, so it is not easy to work together with them from distance.

Therefore, we looked for another area to be helped. The project leaders for PPS projects in Halle are working together with the customers. They gather the requirements, they do most of the contract work, they model the PPS software so that it reflects the customer's way of producing, etc. It can be said that they are the major interface between customer and the Company after a contract was signed. The development teams on the other hand are depending on the information gathered by the project leaders. They need to know what should be developed. In the other direction they provide the final PPS software as well as the documentation and they also provide the user support. It easily can be understood that much information is exchanged between development teams and project leaders. Unfortunately, project leaders and development teams are not located in the same office, they are about 150 km apart from each other. This means, it is not possible to walk across the floor to clarify issues easily. Such a discussion can only be done by phone, email or by having a formal meeting.

In this context it seems natural to standardise the communication between project leaders and development teams. For example, one would expect that a standardised template for requirements



specification exists. However, it does not. At the moment the following process happens: The customer together with the project leader are eliciting the requirements. For example, a customer might have a unique way to distribute his products. This distribution process should also be supported by the PPS software to be introduced. In such a case the project leader is comparing what is already possible with the current implementation to what is actually needed by the customer. The project leader tries to identify how the customer's needs can be fulfilled by using as much functionality as is available already. Afterwards, the project leader specifies in a document how to achieve the required functionality. This also includes adding additional technical information like describing user interface changes. This document, which is called specification in the Company, is handed out to the leader of the PPS development team. He reviews this document and also the estimated efforts. Normally, project leader and development team leader discuss the issue several times and rework the specification. If the specification is finally accepted by the development team leader, the specification is used as part of the contract signed by the customer.

Unfortunately, each project leader uses a slightly different way how to write this specification document. Furthermore, there is no defined way how the process between project leader and development team leader has to look like. For example, some project leaders discuss the specification by phone, others collect a pile of specifications and meet the development team leader in person. This means, for the development team leader that he has to know as many kinds of specification documents and processes as there are project leaders. It can easily be seen that this is not a very effective way of working.

Therefore, we decided to improve this process. The aim was to create a standardised specification template, to define the work process between project leaders and development team leader, and to create checklists for the project leaders to help them doing their work. Furthermore, the new process should be illustrated by transforming an already existing specification to the new template.

Connected to the specification document, the question is where to store specifications. Currently a folder for each customer exists on a file server. The folder structure is standardised, even though many employees say that they do not fully understand the purpose of all sub-folders. Still it seems that this part of the process works and so no additional efforts are needed here. Also the parts of the specification used in the contract with the customer are now stored in an enterprise resource planning system (ERP). However, this was introduced during my study as described in a later section.

After we had decided to tackle this problem first, I wrote a proposal describing what we would like to do. I also described which employees must be involved in this improvement effort. Furthermore, I added a project plan with estimates how much time the different involved people will have to spend on it. It can be said that I prepared a detailed project plan for this improvement effort. After the project leader reviewed the proposal and project plan, I sent it to the boss responsible for the development part of the Company.

After one week I received a phone call by the boss and we discussed some issues he found while reviewing the proposal. For example, he requested to involve additional employees, who might also be influenced by the improvement. I changed the proposal the same day and sent it back to him by email. In the email I proposed to him that he should spread the document to the involved employees, so that they can see the improvement effort is supported by management. The next working day the boss send an email to all involved employees asking them to support me. He also attached the improvement proposal, so that they could comprehend the aim of the effort.

The same day the improvement effort was officially announced. I sent an email to all the involved project leaders asking them to send me three specifications they had produced in the past. Of course I could also have taken those specifications from the file server, but I wanted to demonstrate their involvement. This proved to be a good way, because in the next days several casual conversations came up, where they explained problems and limitations of the current pro-

cess to me. For example, one project leader showed me his copy of a specification. He used the specification to do project controlling as well. He marked the current state of implementation in the document with different colours. In addition to the estimates, he also noted the time actually needed by the developers. So he was able to do a project calculation at the end by comparing efforts spent and efforts paid by the customer. Also some project leaders had invented different methods to track changes in the specifications. One project leader mentioned that the specification is never finished as long as the final software is not delivered. He said that the customer is always changing his requirements, even though they were already fixed in a contract.

After some days I had received example specifications by almost all project leaders. I had the feeling they were supporting my effort very well. Based on the provided specifications I developed a general template. I noticed that all project leaders have a common set of things in the specification like describing the changed data model and user interfaces. However, each of them also used some unique ideas. For example, one project leader described first the goal to fulfil by the feature before he specified the feature in detail. It was amazing to see how good the specifications could be, if the project leaders would exchange their different ideas on how to specify. I tried to include and combine as many of the ideas as I found in the different specifications.

Beside discussing the specification template with the project leader I was working with, I also wanted to discuss it with the development team leader. So I went to the other company location to meet him. In the meeting another employee normally responsible for testing the PPS product joined us as well as a project leader from Halle. I showed them the specification template I had produced so far. They found several minor problems, which I was able to fix easily. During the meeting an interesting discussion came up. The project leader expressed his concern that such a template might not fit in all circumstances. Especially he was afraid that the whole process gets too much formalised and that the process won't be flexible enough anymore. The development team leader and the test employee argued against his view and said that there are always exceptions, still a generally defined process is needed. The project leader agreed but still pointed out that the process and the templates should not be too complicated to use. As an example for a too complicated template he mentioned the folder structure of the project folder on the file server. He said that most of the directories are not needed and that most project leaders do not understand how to use them. As a result every project leader uses them in a different way.

I did not join the discussion, but observed instead. It was very interesting to see the different ideas about how a process and a template should look like. The discussion also touched the topic of how the Company should act in the relationship with the customer. The project leader pointed out that because of the economic pressure the customer may do almost everything. For example, the customer can change the specification during the project and the Company has to accept this. If the Company does not do so, it might lose the customer in the end completely. However, the development team leader said that this is not the correct way to go. He wants a relationship between customer and the Company based on a good partnership. Still he is aware of that in the current situation the customer can misuse his position too much.

Back at the office in Halle, I discussed the specification template again with the project leader I was working with. He said that he is not yet satisfied. He wanted to have more guidelines and formal processes. He was thinking of how the specification process should be embedded in the overall development process. He was also expecting a discussion of the interfaces to other process areas like if the support team can give feedback about how many bugs occurred in a particular customer release. Because many bugs would mean that the specification might not have been sufficient and that the project leader must do future specifications more carefully. He was thinking of some kind of organisational learning with several feedback loops, even though he did not use such terms to describe the issue. Also he wanted the process improvement effort to be developed from top-down, which means that first the overall process must be specified, before sub-processes like specifying can be defined. This was of course a complete different view compared to the other project leader, who was already afraid that a specification template might

be too formal.

For me this meant to find a balance between a very flexible process requested by one project leader and a very formal process on the other hand requested by the project leader I was working with. If both views are compared, it can be seen that they are quite different.

To satisfy those different requirements I created several documents. I created of course a template for the specification document, I created a guide for how to do a good specification discussing several ideas I had identified in the various example specifications. I also illustrated the usage of the specification template by transforming an already existing specification to the new template. To satisfy the need for a more formal view on the whole process I described the process using process charts. Another employee already created a process handbook for the Company. Because he was working in the sales and marketing department he only had focused on those processes. He did not investigate the development processes. Therefore, I could add the specification process as an initial step to document the development processes as well. This also helped to at least define the interfaces between the specification process and the financial processes like contract handling, issuing an invoice, and issuing a delivery bill. However, I was not able to provide a complete process view at this point. For example, I was not able to describe the interaction with the support team and I refused to define the processes at this point.

I was able to convince the project leader I was working with to at least try to introduce the standardised specification template and process, even though the other processes were not defined yet. So I sent all documents prepared so far with a detailed explanation of their purpose to the boss responsible for development. I asked him to take the decision to introduce the template and process. And then nothing happened for weeks. I tried several times to remind the boss about this issue by phoning him or sending him an email. Still nothing happened. In the meantime the boss took over a RETAIL project as project manager. He was now working three days a week at the customer. He was not available to take the decision. In addition, his work load was that big that he might have just forgotten about it. Later in an interview he said that there are always so many different things he has to do that he is not able to do them all. He has to prioritise them. This means that at the point I was asking him to take the decision other things were more important and so he was not able to decide if the new process and template should be introduced.

When I left the Company at the end of my fieldwork, the new template was still not introduced. I handed all created documents and process descriptions over to the development team leader. Still I do not know if the improvement effort was ever finished. I consider this improvement effort to be not successful, because it did not change or even improve the way the employees are doing their work.

### **5.1.1 Specification Improvement Effort Revisited**

In this improvement effort I tried to apply what I have described in the previous chapter about software process improvements. For example, I provided rationals for the change so that management gets convinced and supporting. To some degree this worked. For example, management announced the improvement and asked all involved employees to support me. So it can be said that the success factor management commitment was fulfilled. I also involved all concerned employees and they actively participated in the improvement effort. I gave feedback for example by quickly providing updated versions of the documents. I also did detailed planning of the improvement effort.

Still the improvement effort failed. The main reason is that management was not available when needed. I estimate the time to take the introduction decision to about 30 minutes. Those 30 minutes were not available or something else was more important. Another solution would have been to delegate the responsibility for this decision to the development team leader and project manager I was working with. Another problem might have been that the pressure for change was not big enough yet. The Company is still able to produce specifications and the quality of the specifications is good enough. I also had the feeling that the different views about process

definition have hindered the introduction. The employees were not able to create an internal lobby to support this improvement effort.

## 5.2 Improvement Pushed by a Single Employee

When I started my study I noticed that since my practical two software packages were introduced in the Company to support business administration. One package is a software for customer relationship management. Beside maintaining customer contact data, also a so called sales-pipeline can be maintained. If a sales person makes a contact to a possible customer, the contact information is entered in the system. In addition, the contact is classified according to the product he is interested in. Furthermore, a percentage value is added to each contact representing the probability that the contact will sign a contract. This software was introduced because the Italian company required it. They base the payment of each salesman on the number and volume of acquired contracts. The software can also be used to control the general business development and to get an overview of the current state of the sales-pipeline. If a contract is signed, no further work is done in the customer relationship system. Instead the introduced enterprise resource planning system is used.

An enterprise resource planning system (ERP) helps to administrate the general administration tasks in a company. The Company has already started introduce an enterprise resource planning system of a partner company. The Company is offering the enterprise resource planning system together with its PPS product as an integrated package. Normally, such an ERP system also provides accounting functionality. However, the Company was already using another application for accounting, so it was not using the functionality of the ERP for this purpose. Since the used accounting software and the ERP are not integrated yet, the accountant has to re-enter every necessary data in the accounting system manually. It is planned for the near future to buy additional components, so that both systems can work together and become integrated.

When I first started working at the Company the ERP was not used by many employees. The system was used to generate offers, contracts, and delivery bills. To be able to do so one employee added a complete list of all products to the system. This was a major task, because many data entries had to be created. It seemed to me that this employee was pushing the whole ERP introduction effort. On the file server I found several documents describing the current introduction state and project plans about when to start the next phase. The accountant, who also did many general business administration tasks, used the ERP to generate the needed documents. For example, to create a new contract, first a customer must be selected and a new entry must be created. Afterwards, the products to be included in the contract must be added as well as the contract conditions must be selected. Often additional documents like a specification must be added to a contract. By generating the contract the contract gets printed so that it can be signed by one of the bosses. Finally, the contract must be send to the customer, so that he can sign it as well.

It can be seen that those administrative tasks are a repeating process. Several steps like creating an offer, transforming the offer into a contract, transforming the contract into a delivery bill, creating an invoice, and registering an incoming payment are required. Also it can be seen that information from different employees is needed. For example, only the project leader knows exactly the negotiated conditions for the contract and he also must provide the complete list of products to be delivered as well as the belonging specification. But at this time, all the work within the ERP system was only done by the accountant.

During my study the usage of the system was extended. First the employee pushing the introduction together with the system administrator configured the system so that it can be accessed remotely. This means, every employee within the Company can log on the system through remote desktop access without having to install a client on his local PC.

Beside such technical tasks, the employee adopted organisational tasks. For example, he doc-

umented and defined the processes needed. The employee was not able to work continuously on the introduction, because he had many other tasks to do like selling and marketing the Company's products. So often nothing happened for several weeks, but then quickly much was achieved within some days. After he had defined some processes using a general chart package, he presented the processes to the boss responsible for sales and marketing. The boss reviewed and signed the processes. Sometimes this also took some weeks.

The defined processes were collected in a process handbook. As I noticed, the employee created the process handbook on his own initiative and not on request by his boss. The process handbook only dealt with administrative processes and not development processes. However, also some development processes already existed. For example, there is a guide how source code version handling must be done and how a bug fix must be integrated into the different development versions. Those processes were created by another employee also on own initiative. Both process descriptions are not integrated with each other and also different notations are used.

It was the aim of the ERP introduction that also the project leaders use the system to directly create the offers and contracts. Not all project leaders liked the idea. For example, they were concerned about the extra work they had to do. Also they were concerned about missing important legal points when generating a contract.

I got involved into this project by first reviewing the processes in the process handbook and second by writing work instructions on how to use the ERP system. Those work instructions were meant as a help for the project leaders to get used to handling the ERP system. I created a first version of those work instructions. They contained many screenshots with step-by-step instructions. I mailed the work instructions to the accountant for review. She added several important points I had missed. The work instructions were later added to the process handbook as additional references on how to implement the process.

A workshop was organised to evaluate the current state of the introduction but also to get to know the ERP system better. The accountant, another employee partly working in administration, the secretary of the boss in Halle, the employee pushing the introduction, and me as well as a consultant of the ERP production company participated in the workshop. At the beginning the way chosen to structure the products in the ERP system was discussed. Later the process to create the different documents like orders and offers was discussed as well. Several minor problems were identified. For example, some of the document templates still contained errors and also minor bugs in the software were identified. At the end all participants agreed that the introduction looks very promising and that it will help to improve the Company's internal processes.

Only about two weeks after the workshop another meeting was announced for the project leaders. All project leaders and also the boss responsible for sales participated in this meeting. This meeting was a training session to show them how the ERP system must be handled and also how it should be used. Also the work instructions were handed over, so that they could read about the process by themselves again. Some of the participants requested a test system, so that they can try the system without damaging anything important. At the end of the workshop everybody seemed to be satisfied and the consensus was that the introduction of the ERP system is a good step. This workshop also helped to convince all project leaders that the system is easy enough to use and that they can not make major mistakes while generating legal documents like contracts. This was also an important outcome of this workshop, because now everybody was sure the ERP was going to be introduced and so a resistance would be useless.

At the same day after the workshop the employee who was pushing the introduction sent an email announcing the availability of a test system. I was impressed on his fast reaction, because this clearly showed that the participants' concerns were seriously picked up.

After the introduction, from time to time project leaders dropped by in my office to ask questions on how to use the system. For example, there were problems using printers and also selecting the correct template for the various documents. In conclusion it can be said that the introduction of the ERP was generally accepted. It reduced the work load of the accountant, also less infor-

mation had to be transferred within the Company. Another major outcome was the creation of the process handbook defining important processes.

### **5.2.1 Enterprise Resource Planning System Introduction Revisited**

From an ethnographic view, this was a very interesting improvement effort. I was not too much involved so the outcome of the effort did not depend on my behaviour. Still, I was sufficiently involved, so that I was able to observe it quite good. However, one has to note that this improvement effort was already started before I started my study. For example, when I arrived at the Company, the decision to introduce this particular ERP system was already taken. Also the responsibility for this introduction project was already assigned to the single employee. It turned out that this was a major success factor, because this particular employee put much effort in this project. He was pushing the project very much. For example, he created the process handbook by defining those processes to be managed in the ERP system. He worked together with the accountant, who was also very supportive.

Another important factor for the success of this project was that the ERP system was already known in the Company because of the cooperation with the ERP producer. Almost all employees had at least a basic knowledge of the system.

At the beginning of this project only few employees supported the effort. In contrast, some project leaders were not satisfied with the decision to introduce such a system. By providing detailed instructions and tutorials, all concerned employees were persuaded. Here, it was of great help that a running system was available to illustrate the new work processes in a running system. Another important point might be that the management decision was already taken and that management showed support for the introduction. Even though management showed support for this effort, this does not mean they were involved in this project heavily. They had delegated all responsibility to the single employee.

## **5.3 Bottom-Up Approach**

During the introduction of the ERP system it turned out that it would be convenient to have a central storage place for the process handbook. Of course it would be possible to store it on a file server in some folder. However, there were already about 100.000 files on the server and people in general had the feeling to get lost in all those different directories. During my study I was able to find many interesting files on the file server, files nobody was aware of that they still exist. Placing the process handbook there would have meant to put it nowhere, because only those people knowing about it would find it there.

The employee pushing the ERP introduction was thinking of a central Intranet place to store such company wide information. Therefore, he developed a small web site in his spare time. He used a presentation program with different slides for the sub-pages to create the web site. He generated the web site code using the program's HTML export function. This was a quick and dirty solution. Introducing this solution would have meant that only a single person is able to maintain the web site. This also means that the web site only is updated when this person has enough time. Judging from what I daily observed, I was quite sure that this is not going to be a long term solution.

Therefore, I installed several different OpenSource forum systems and also two content management systems on a web server. I showed those prototype installations to the system administrator and to the other employee. The employee agreed that this were much better solutions than generating a web site out of a presentation program. However, at this point time was running short so a quick solution was needed. Therefore, we agreed to put only a forum system online and to store the process handbook on the file server. At no point a management decision was involved. The decision to install a forum was taken by the employee, the system administrator, and me.

The availability of the forum was announced by email to the involved employees like the project managers. At the same time the PPS development team leader was asking for ideas about how to improve the user interface of the PPS product in an email to all employees. I phoned him back telling him that now a forum system is available, which can be used for such discussions. He agreed to use the forum for this purpose.

After a short time already about 30% of the Company's employees had created an account for the forum system. However, discussions were still rare. At another point a PPS core programmer created several categories in the forum for development discussions. This turned out to be an important step, because now the forum was also used for internal discussions about development environments, coding hints, technology descriptions, etc.

At the beginning I had configured the forum system so that everybody could post an article without having to register. It was my aim to decrease technical barriers as much as possible. However, one core developer reconfigured the forum system. He told me that in the Company everybody should be skilled enough to use such a system. Also he said there is no need in the Company for anonymous posts, because of the familiar relationship within the Company. He could not imagine that someone might be interested to post a question for example about a very simple technical issue anonymously.

At the end of my study the forum was mainly used by the PPS developers. Also I observed that the forum is not known by each employee even after several months of existence. Still I think the forum system helps the Company to share some of the employees' knowledge within the Company, even though it might be just a small step.

### **5.3.1 Improving Knowledge Sharing Revisited**

First I did not even notice that this was also a process improvement. It always seemed to be a sub-chapter of the ERP introduction. But taking a look at threads in the forum shows that it is mainly used for technical discussions. The original aim to support the ERP introduction was not achieved. Still something valuable was created.

Interestingly, in this effort no management decision was required. Introducing the forum software was therefore just a matter of some days. Also at the beginning of the ERP introduction it was not planned to have this forum system for knowledge sharing. The need for it emerged and was satisfied immediately. This is clearly a bottom-up approach – employees are faced with a problem in the current work processes and they fix it by themselves.

## **5.4 Lobbying and Bottom-Up Approach**

One of the first questions I was asked while visiting the other company location in Chemnitz was about the introduction of a document management system. At this early point of my study I was not able to give a positive answer. Of course it was clear that such a system is required urgently, still I pointed out that for such a drastic change a management decision is needed.

Documenting is an essential part of software engineering [Som04, e. g.]. Requirements must be defined, design decisions must be formulated, project plans must be maintained, user documentation must be generated. In addition, a company is always producing many documents like letters, contracts, marketing material, presentations, etc. Of course this is also true for the Company.

For several years the Company was using a file server for providing and exchanging documents. Source code is managed by a version control system. Also by introducing the ERP system and customer relationship software, documents are attached to the action they were used for in those systems. This means, a specification is now attached in the ERP system to the contract it was created for.

Nevertheless, there are still many documents left, which also need to be managed. For example, many documents are available describing the PPS and SCM products. Since the Company was bought by the Italian company, documents also started to get translated into English. Beside such product descriptions, also many documents are created within the different projects. For each customer a folder exists containing all documents used for communication with the customer. Other documents contain process descriptions, training material, meeting protocols, etc. The major problem is that the folder structure on the file server grew over several years and it is hard to identify an overall order. Of course the employees working with the file server since several years know pretty well where to look for a file. However, for many employees it is a nightmare to work with the file server solution.

Access to the file server is not restricted. Anybody can add, modify, move, copy, and delete any file or folder on it. On the other hand, I also found some documents, which are protected with a password. This was very surprising, because normally all information is shared within the Company. The bosses both said that for them it is very important to share all information and news with the employees. Here, "all" means good as well as bad news!

In some folders some kind of manual document management system was done. The file name contained an abbreviation for the employee who changed the document last as well as the date of the last change. For example, a file changed by myself would have been named "050428ses filename.txt". If a change to a file was necessary, it was common sense to copy the file and save it using a new file name. This meta information about the author and last change date was added at the beginning of the file. The drawback of this approach is that alphabetic file order was not helpful anymore, because files get ordered first by the date, second by the author and only at the end by their name. In that way it is easy to miss a newer version of a file, because not all versions of a file are found next to each other in the folder.

There are many other problems involved with a file server solution. Once I noticed that there are two different files for a marketing document available. For me it was not clear which document to use. So I called the employee responsible for maintaining this document. The employee also did not know where the second document came from. Even though someone was assigned responsible for this document, still redundant and invalid documents existed. This shows that it is very hard to manage documents with a file server solution.

In another example one secretary sent an email to all employees in Halle. She was asking why the folder containing all document templates like for letters and presentations is empty. This means, someone had deleted or moved away all those documents by mistake. This problem did not turned out to be bad, because the secretary still had a copy of all templates on her local PC, still it shows the possible danger of a file server solution without access restriction.

Compared to the other improvement efforts this effort was quite different. In the beginning it was not announced or formally started, the improvement effort emerged instead. While talking to the different employees I noticed that most people were aware of that something needed to be done and that a better solution is required.

Having finished half of my study, I started a small survey in the Company. The survey listed several modern technologies like Extreme Programming, garbage collector, CMMI, design patterns, but also some more traditional things like market analysis and waterfall model. For each of those items the employees had to answer if they know about it, have in-depth knowledge about it or have never heard about it at all. At the end the employees also categorised themselves into different categories like sales person and programmer. After the survey was finished, I put a description of all the different items in the forum system also providing links to additional information resources on the Internet. One item considered the question if the employee has ever heard of the term document management system. In the description I provided a link to a demo of an OpenSource document management system, accessible online without installation. Later some employees told me that they had tried this system. One project leader also asked me, if it would be possible to install such a system for their next project.



Also in some interviews I did, I asked the interviewee about his opinion on how to solve the file server problem. In one interview the interviewee agreed that a better solution is needed. As a reaction he put this topic on the agenda of the next project leader day meeting. He asked me to present some easy to implement solutions, which could be extended in the future.

I prepared a presentation for this meeting. First I was looking for a very easy solution. I installed a Wiki software system<sup>1</sup> on a test server. The idea was to have an Intranet portal providing access to the various documents. The documents themselves would still be stored on the file server. However, to locate them one would use the Intranet portal. Also I was thinking of that this solution might be used for other purposes. In the past a Intranet page existed in the Company, presenting development information for the PPS developers. This effort seemed to be stopped in the meantime, no new descriptions were added for a long time. By using a Wiki it would have been possible to also reactivate this very good information resource.

On the other hand, I also wanted to show a full featured document management system with collaboration functionality. So I installed Microsoft Sharepoint Services and Microsoft Office 2003 on a test system. Already during this early phase I noticed that many employees were supporting me. For example, the system administrator provided a test system, the software, and he helped with additional advices. Other employees noticed the added point on the agenda and asked me what I was going to present. This started some kind of rumour that something important is going to happen at this meeting.

I structured my presentation into three main parts. In the first part I was describing the problem on some slides. Those slides summarised what I have said about a file server solution so far in this section. On the other hand, I listed the features needed like version handling, full text search, keyword and meta information for each document, access control, accessibility from each workstation within the company network, etc. In the third part I first presented the Wiki solution and afterwards the solution using a real document management system. Already after presenting the Wiki some employees agreed that this is the solution for this long lasting problem. After demonstrating Microsoft Sharepoint almost all employees were convinced that a solution is possible. The point was not to convince them that a solution was needed - almost every employee knew this already. Instead it was important to show real world examples like running systems to visualise how a possible solution could look like. During my practical two years ago there was already the idea to introduce a workflow system like Lotus Notes. Presentations about the benefits and advantages of such a system were given by employees, who worked in their former employments with such a system. Still, nobody was able to present a running system. Also the investment needed for introducing such a complex commercial product were not possible at this time. I was aware of that the Microsoft Sharepoint solution is also not possible to introduce, because of the current economic situation and the required high investment. A discussion between various employees started. Some pointed out that an lightweight solution based on a Wiki is better than no solution at all. On the other hand, employees said that a Wiki is not worth the effort. Especially they criticised that in such a scenario the file servers are not removed. Also different employees mentioned that version handling and full text search is not possible using a Wiki. The discussion showed that no decision was possible at this meeting. Therefore, one of the two bosses decided to start an internal project to find a solution. He named several employees as well as himself to participate in this project.

After the meeting I recognised that no date for an initial meeting was set yet. So I emailed the different assigned project members. I had the feeling that now I have to push this effort. Also just some weeks were left till the end of my study. In the next days I discussed the project with several different employees. Many of them were concerned that the bosses are going to choose a cheap and easy solution, which will not solve the complete problem. Also they were afraid that the project will need several months or that in the end the project is even stopped and not carried forward anymore.

---

<sup>1</sup>see e. g. <http://en.wikipedia.org/wiki/Wiki>

Before the initial meeting I sent an email to all project members containing a list of documents which I think have to be managed. I did not receive an answer nor did I see a printout of my list used at the meeting. At the beginning the boss introduced the meeting by repeating some of the arguments I came up with in my presentation. During the meeting it was decided to define document classes to group the documents to be managed. In addition, it was decided to define the process and responsibilities for document creation. However, the involved project members said that this process already exists, because employees are responsible for documents already. Finally, it was decided to look for a possible technical solution. This task was assigned to me, because I had already looked for different possibilities while preparing the presentation. During the meeting I understood that the Company is not looking for a collaboration system. The main aim is to have a central document repository providing versioning and full text search. In addition, write access to the repository should be limited to only those few employees, who are responsible for the documents. Also it was not the intention to include all documents currently available on the file servers. The main aim was to manage all marketing and sales documents like presentations and product descriptions. Also document templates should be managed by the document management system. It was not intended to store any project or customer specific documents in the system. Such documents should remain on the file servers. One employee suggested to use the version control system as the repository and to implement a small client for accessing it. However, this idea was rejected quickly.

After the meeting I was convinced that only a technical solution gets accepted, which does cost almost nothing and which is easy to implement. It was now my task to find the best solution within the given constraints. I decided to take a look at available OpenSource document management systems. I found out that only few are mature enough to be used in a real production environment. Some of the systems had many bugs or a very bad user interface. Others required special software on the clients, something I also did not like very much. At the end I only identified a single system. I decided that this is the only solution, if a real document management system should be used.

A decision was required about which system should be introduced. Therefore, I asked for another meeting. Most project members agreed immediately, but the involved boss did not have any time in the next days and the other boss was on vacation. This is a recurring theme in my study. Every time a decision by one of the bosses is needed, they are not available. Therefore, projects like an improvement effort are delayed for several days. In this case more than a week was needed. Nothing happened in the meantime.

On the meeting day I went to the other company location, because almost all project members were working there. The meeting was started, but without the boss. He had received an important call and now something else more important had to be done. So I presented the technical solution to all the other project members. They were surprised that I was able to identify a real document management system free of charge. Everybody agreed that this is a very good technical solution and that there is no reason not to introduce it. Interestingly, one of the first questions coming up was about the technological base of the presented software. I explained that the system is programmed in Java Server Pages combined with Java classes and that a Java application server like Tomcat is needed to run the system. This technological base was seen as a disadvantage, because nobody in the Company has worked with this technology so far. Also some problems in the user interface were pointed out and I mentioned some bugs I discovered while testing the software. Still no major bugs were found. I proposed to pay the developer of the system to fix the remaining bugs. I had found out that the developer of the document management system was once studying in Halle, so he should be reachable easily.

After the meeting before I returned to Halle the boss dropped in. I again presented the system to him and he agreed that it is a good solution. At this point the decision was made to introduce this software. After I returned to Halle the same day several employees were interested what decision was taken. They were happy to hear that the system is going to be introduced in the near

future. Before I left the Company for a two weeks vacation I discussed the hardware requirements for the system with the system administrator. He agreed to order a new server, which was available after I returned from my vacation.

Immediately after I returned from my vacations, I started installing the system. This also included a detailed test of all functionality to identify all bugs in the system. I created a list with a description of all bugs I discovered. I also contacted the developer of the software and asked him if he would like to fix the bugs for a payment. Even though he seemed not to have much time, he happily agreed to work on it if he can receive money for the work. I immediately emailed the boss so that he can decide about whether the developer should be hired for this work. But once again nothing happened. At the end of the week I tried to phone him. His secretary told me that he was already on his way to Halle for a meeting and really an hour later he arrived. Still, the whole day he had no time for this issue. At the end of the day he left the office in a hurry and still no decision was taken. At this point I was quite frustrated, because to take the decision would just have meant 10 minutes of discussion. Also other employees noticed that the boss had no time to decide and they were disappointed that an important improvement is not done quickly.

I decided to fix the major bugs on my own. So I started to set up the needed development environment and development tools. I downloaded the latest source code and started fixing the bugs. After two days I found workarounds for most bugs and other problems, even though these were not good solutions in all cases. There was no other way to go, because it was the last week of my fieldwork. Also another employee started to look into the document management system. He had some experience in the used technology. I also prepared many different documents describing in detail how to install the system and how to change the source code if needed. At the end of the week I handed out the whole work to the system administrator. It took almost a whole day to explain all the various things I had done so far and also to explain how I would proceed.

In the last week several employees dropped in to get a small presentation of the system. All agreed that it would be a great improvement for the Company to introduce it. One employee discussed that at the beginning there might be some resistance against the system, but that in the end all employees will understand that it is better to use such a system instead of a file server.

I also sent out an email to all project members explaining the current state of the project and what needs to be done to go on. Immediately after my email I was phoned back by a project member. The project member wanted to know if I had handed everything over and that my work won't be lost. I also sent an email containing the contact information of the document management system developer to the boss and the system administrator. In this email I also summarised what I had already discussed with the developer.

I left the Company. I was not satisfied, because I was not able to finish the introduction of the document management system. I hope that they will eventually manage to introduce the system. However, it might need several more months.

#### **5.4.1 Document Management System Introduction Revisited**

From the description it can be seen that there was no linear progress in this improvement effort. Of main interest is how the project got started. When I entered the Company almost all employees wanted a better solution for document storage. However, nobody did anything about it. I had the feeling that the employees were waiting for some kind of stimulus by management. However, this stimulus did not happen. Instead of looking for an alternative, the employees got used to the file server solution, even though they knew it was not a really good solution anymore. The improvement project got started after my presentation of the problem at the project leader day meeting. However, I did not give this presentation on my own request, instead another employee had asked me for this. Even before the presentation, I was already sure that most employees would support my effort. In the different interviews and casual conversations I did, they had already expressed their dissatisfaction with the current state.

I used the project leader day meeting to create a lobby for this improvement effort. I had the feeling that management was a little bit surprised by the strong reactions of the different employees during the meeting. For example, one employee said that if the Company is going to implement just a quick and dirty solution, he won't help at all. Other employees even demanded a decision during the meeting. I did not participated in the discussion. I had already done my part by initiating the discussion. After the meeting the problem was public, meaning that enough pressure for change was evident.

This lead to the creation of the actual improvement project. In this project it was my task to find a technical solution as described above. Only two additional meetings were needed to start implementing the change. However, again the improvement process was slowed down, because management was not available. It was a great advantage that the processes and responsibilities for changing documents were already defined, so here no additional work was required.

During this improvement effort I felt great interest by various employees. Maybe this was because almost all employees were affected by the change. I discussed with them the proposed solution and explained how their daily work will change.

This improvement effort got started because of a strong lobby, which grew during several years. In this case management was not supporting the introduction at the beginning. Instead, I had the feeling that management was persuaded by the employees. Even though I did not directly involve many employees, many of them very really interested and offering support. At any point, this improvement approach remained a bottom-up initiative of the Company's employees.

## **5.5 Getting Out**

I have been five months at the Company. It is not easy to define a point where to stop the fieldwork. Many authors discuss ways to get in, to get access. However, I have not seen authors discussing in detail how to get out. At the end I was heavily involved in the improvement efforts. I am convinced that I had pushed especially the document management system effort pretty much. However, as far as I have observed, all major changes in the Company have been pushed by a single person, so I am convinced that I acted naturally.

At my last day I brought a cake to the office. All employees joined in a small breakfast with coffee and cake. They thanked me for the work I did. They also said I am always welcome to come back.

## Chapter 6

# Conclusions

Due the ethnographic research approach, I was not sure about what the contribution might be at the end. It is an essential part of ethnography to not formulate an hypothesis at the beginning and to test it during the study. Instead the researcher investigates the problems the practitioners are dealing with and what possible solutions they come up with during the study's proceeding. Such an approach fitted well what I wanted and did not want to contribute. It was not my aim to test if a certain method can be applied in a small organisation. I wanted to participate in software process improvement efforts and experience how they are done in reality. This methodological choice also influenced this thesis' results and the way the results are presented. To exemplify, this study does not provide a complete model for software process improvements in small organisations. Instead it presents the identified problems and their solutions in an intertwined manner. This results in some general guidelines how software process improvements can be done in a small organisation.

Even though the study had no defined goal to prove a theory or to create a model, still there are several contributions to be made. I got aware of those contributions while doing the study as well as analysing my fieldnotes. In section 3.8.5 on page 33 it is demonstrated how I created a code network. This network revealed groups of codes that I have used very often while coding the data. It also turned out that codes in a group are used for similar observations. In the following sections I am discussing the different identified code groups and the implications I draw from them.

### 6.1 Improvement Strategy

Taking a look at quality models like CMMI [Tea02b] shows that those models propose a very formal way about how they should be used. A possible implementation of an improvement effort according to those models consists for example of an improvement group continuously looking for possible improvements. Such a group also steers improvement efforts or creates projects for them. Possible members of such a group are managers, project leaders, and process specialists.

This is a logical approach. However, in the studied company it would never be possible. They cannot afford to create such a group and they cannot afford all discussions and group meetings necessary. The reason is that each employee has to create business value, which can be sold.

From a broad perspective quality models suggest a top-down approach, something not applicable in the studied company. For a small company like the Company it is not possible to spend several "man days" for creating an overall process model and to implement it in later phases.

Still, it is clear that a small company also needs to improve. *Therefore, this master thesis suggests a bottom-up approach adding top-down methods if needed. This means, if employees identify a repeating problem in their work processes, they should try to solve it without needing first to initiate an improvement project. If the solution proves to be good, actions must be taken to implement the solution in the whole company.* This might mean that an official improvement project is started to ensure the solution gets implemented in the whole company and to align the

improvement with the overall processes and company goals. Therefore, it must be the task for managers to encourage employees to come up with improvements. The employees must prepare the solution in advance as much as possible, so basically at the end only a formal decision is needed by management. In this case management must be available and make the decision quickly. This can also mean that management is relying on its people and is delegating the decision altogether. In the improvement effort for the specification process basically no management decision was needed. Both, project leader and PPS development team leader, agreed to the new process and template. They are the people who have to work with the process and template. If they think that this new process and template will improve their work, they should be able to take the decision to introduce the improvement. However, this did not happen, the improvement effort is still waiting for a management decision to become introduced.

The knowledge sharing improvement effort showed that management decisions are not always needed. Based on the discussion between the employee pushing the ERP effort, the system administrator and me, we decided which software system should be used. We decided to introduce a forum software. Neither the structure of the forum nor the question if access restrictions are needed were discussed with management.

Doing process improvements bottom-up also means that employees have to start initiatives on their own. While talking to several employees I noticed that they all have great ideas on how to change the way work gets done. However, they do not act. They do not try to initiate change. They seem to wait for management to give them a sign to start. However, at least in the studied company this won't happen, because both bosses have so much to do that they do not have enough time to also commit themselves to improvement efforts. To implement successful software process improvements in such a small company means that both employees and managers either take and delegate responsibility in a flexible way.

A bottom-up strategy for software process improvements has also some drawbacks. For example, the single employee might not be able to align an improvement effort to the overall processes and business goals. The employee might miss important aspects because of his focus on his own day-to-day work. Also employees might only tackle those problems they are personally faced with. This means, more broad improvement efforts might not get started. Those problems can be resolved by also integrating a top-down approach as described in quality models like CMMI. Even though management should delegate responsibility for improvement efforts, this does not mean to delegate all responsibility! Some improvement efforts must still be initiated and guided directly by management.

## 6.2 Success Factors

In section 4.5 starting on page 42 I discussed various success factors for software process improvement efforts mentioned in literature. In this section I am reflecting on how I experienced the importance of the success factors management commitment, awareness, visibility, feedback, and involvement during my study.

One often repeated success factor is management commitment. *Judging from my experience, direct management commitment is not always that important. It does not always have the same influence on work if or if not management directly supports and pushes the improvement effort.* Same results can also be achieved without management by the employees themselves. However, it is important that management is available if needed. This means, if a decision is needed, which can only be taken by management, management must be available to do so. An improvement effort is heavily hindered if it has to be stopped for several weeks each time a management decision is needed. There are generally two possible solutions to overcome this problem. First, management can delegate the responsibility for the decisions. However, just delegating responsibility is not enough, also control must be delegated. This means, if an employee is assigned responsible for an improvement project, he must be allowed to take decisions and he must get a budget. The

other way to solve the problem of management availability is that management is ensuring to have enough time to take decisions if needed. In the studied company this seems to be impossible at the moment, because management is heavily involved in the ordinary day-to-day work. Both bosses are working besides their management tasks also as salesman and project manager. Therefore, in the Company one feasible solution supported by this study is to delegate responsibility.

Another important success factor according to literature is awareness. Awareness is meant as recognising that process improvement itself is also a process, which needs to be managed and improved. In a first coding step I assigned the code awareness very often to many different observations. Later, while I was re-examining my codes I noticed that I might have had a different understanding of the term. I understand awareness in a much broader sense. *In my view, being aware of software process improvements means to understand that software is developed in a process and that such a process can be improved. Only the next step would be to also recognise that improving itself is also a process.* I often had the feeling in the Company that they were not thinking about to improve their processes. Maybe they are too used to the way they work and have found a workaround for most problems. However, I also think that their very low employee fluctuation is not just positive, but also has some drawbacks. Having new employees from time to time increases the amount of new ideas on how to best perform the work. Since the Company was bought by the Italian company, already some new employees joined the Company. However, they are basically not working with the PPS and SCM products, so their influence on the processes in those areas is quite small.

Visibility of improvements early during an improvement effort is another important success factor according to my observations. This is also pointed out by literature. Visibility means to communicate improvements so that employees can experience that something is changed. This also means to present improvements as early as possible. Improvements do not have to be big changes, more often it was enough to quickly do initial steps. For example, in the ERP introduction effort a test system was requested by the future users. This request was fulfilled within some hours. Another way of visibility is to use demo systems and prototypes instead of discussing improvement ideas in general. *Judging from my observations, other employees were not interested in theoretical improvement discussions. Instead, they wanted to see at least a sketch of how an improved process might look like. After making an improvement visible very early, most employees immediately started supporting the improvement effort.*

I encountered feedback as another important success factor. Of course feedback is connected to visibility. Feedback means to react upon concerns and fears raised, and not to defer the raised issue till the next official meeting. If the involved employees do not see that their concerns are taken into account, they will not support the improvement effort. This can even mean that resistance to change is created or increased.

Finally, I like to discuss the success factor involvement. It is very clear that those people, who are going to be influenced by a changed process, must be involved. For example, it would have been no good idea not to involve the project leaders in the specification improvement effort. At no point I had the feeling that employees at the Company were not interested in getting involved. Of course all of them have a lot of work to do, still they always tried to help and to provide their input if needed. Beside involving employees, one should also try to create a lobby. In the Company several single employees tried to improve some processes. Still, they never tried to create a lobby to push necessary changes together. Lobbying also means to use the informal network to create pressure for change. Lobbying can be done in many different ways. For example, employees can be asked for input, but one can also provide input for their process change proposals. Creating the needed pressure for a change is not always easy. *A company like the Company, which is not doing any formal software process improvements only seems to react if the need for a change is very high. By creating a lobby, which overemphasises this need, changes can be achieved earlier.* For the introduction of the forum software we created a small lobby consisting of the one employee, the system administrator, and me. By pointing the PPS development team leader to the new forum

system, I also tried to get his support. Another very effective lobby was created for introducing the document management system. Even before my presentation at the project leader day meeting, several employees agreed that something had to be done. At the meeting the necessary pressure for change was reached and so the bosses had to take a decision to start the improvement effort officially.

At this point I like to point out that lobbying should not be misused. Of course it is ok to involve other employees in an improvement effort. But this does not mean to create internal groups, who refuse any other solutions. Lobbying in the sense of this master thesis does not mean creating internal politics. Also, if it is not possible to create a lobby, one can either reformulate the problem description or move to another possible improvement. It is not useful to insist that something gets improved, which nobody else is interested in. In such a case the employee should invest his energy in another improvement effort.

I consider management availability, visibility, and lobbying as the most important success factors based on the experiences of my fieldwork. Those success factors must be integrated in a bottom-up strategy for software process improvements. During this study no single way for such a bottom-up strategy was identified. Still, there are repeating elements of such a strategy, which can be formulated as guidelines:

- Employees realise that they can improve the way they work by themselves (awareness).
- Employees do not wait for management to initiate the necessary changes, because everybody is responsible for software process improvement (bottom-up).
- Employees try to find other employees having the same problem (lobbying) and to create enough pressure for change.
- In a joined effort they try to find possible improvements together (involvement).
- Identified improvements get implemented quickly, maybe starting with small steps (visibility).
- Management delegates as much responsibility for the improvements as possible to the employees (bottom-up).
- Management takes needed decisions as quickly as possible (availability).
- Concerns of involved employees are taken into account (feedback).

From this short list it can be seen that in a bottom-up strategy it is the single employee, who has to initiate a software process improvement effort. This single employee must look for other employees to build a lobby and to create enough pressure for change. Judging from the field experiences in many cases a single employee has to push the whole improvement effort. It can be seen that the success of an improvement effort heavily depends on the commitment of the involved employees.

Combining the success factors with a bottom-up strategy for software process improvements reveals that there is no reason why small software development organisations should not be able to improve the way they work.

### **6.3 Ethnography in Software Engineering Research**

Beside looking at possible contributions in software process improvements research, it is also important to take a step back and reflect if doing an ethnography within software engineering research proved to be useful.



First of all it proved to be *possible*. This means, one can use ethnography in software engineering. There is no reason why it should not be possible to apply a field method like ethnography in software engineering. Software engineering is a task done by different people working together. A social science research method as ethnography can of course be applied in such a context. This is also discussed by Sim et al. [SSS01]. They did a workshop where qualitative research and specific research methods like ethnography were discussed in detail by many well-known software engineering researchers.

This study was done using ethnography as a research method. The study provides insight on how process improvements are done in a small company. There might be even lessons for the reader he was not aware of before reading this paper. This means, this study eventually has contributed to the body of knowledge, even though the contribution might be small. In general, this means that ethnography applied in software engineering research can contribute new insights. This is also one result of the workshop done by Sim et al. [SSS01]. However, they stress the point that the contribution must be well formulated. Also one must be aware of the disadvantages of qualitative research.

Judging from my own experience doing an ethnography as a software engineer is a very good way to get aware of the subject's nature. Using a qualitative research method helps to understand that software engineering is more than models, technologies, and formal methods. For example, the need to create enough pressure for change is not a technical problem but a social one. Social interaction is a very important part of software engineering, something which was already pointed out by many different authors many years ago [Bro01, DL99, Hum97, e. g.].

However, to get the most out of applying a qualitative research method training in front is needed. I had to learn almost from scratch how to do ethnography. While I already started my study, I was still reading up about possible threads and problems. A possible solution would be to include qualitative work in the education of software engineers. This means for example to include courses about organisational behaviour, reflection seminars, and communication training. Even though I did not have this education before, still the study was a big enrichment for my own understanding of software engineering and I am still thrilled having done it. This need for education about qualitative research is also stressed by Sim et al. [SSS01]. They say empirical education should be included early in the software engineering curriculum, because a student will need some time to fully understand the philosophical view behind it.

Rönkkö et al. [RLD02] stress another important point. They say that ethnographic studies in software engineering often reveal "bad methods". Bad methods means that software engineering methods and processes are not applicable in day-to-day work and that they are theoretical. On the other hand, Rönkkö et al. [RLD02] say software engineering often reveals "bad practices", which means that practitioners do not correctly or completely implement software engineering practices. Therefore, the methods fail. This study is an ethnographic one and reveals bad methods. The way the studied company is doing software process improvements is not described by software engineering literature. In contrast, it seems that the Company's way of improving is completely different from what is described for example in quality models like CMMI. On a first glance this seems like an unresolvable clash. However, one has to understand that both views are correct, because both present a part of the whole problem.

# List of Figures

|     |  |    |
|-----|--|----|
| 2.1 | Logical structure of the Company . . . . .                             | 8  |
| 2.2 | Separation of the different functions between both locations . . . . . | 8  |
| 2.3 | Official organisational structure . . . . .                            | 9  |
| 2.4 | Meeting room in the Company . . . . .                                  | 15 |
| 3.1 | Elements of ethnography discussed by Wolcott [Wol99] . . . . .         | 20 |
| 3.2 | Ethnographic writing styles [VM00] . . . . .                           | 28 |
| 3.3 | Development of a code network . . . . .                                | 34 |

# List of Abbreviations

|              |  |
|--------------|--|
| APS .....    | Advanced Planning System               |
| CMM .....    | Capability Maturity Model              |
| CMMI .....   | Capability Maturity Model Integrated   |
| CSCW .....   | Computer Supported Collaborative Work  |
| ERP .....    | Enterprise Resource Planning           |
| HCI .....    | Human Computer Interaction             |
| IMM .....    | Implementation Maturity Model          |
| IT .....     | Information Technology                 |
| MRP II ..... | Master Resource Planning 2             |
| PPS .....    | Production Planning and Control System |
| SCM .....    | Supply Chain Management                |
| SPI .....    | Software Process Improvements          |

# Bibliography

- [BD97] P. Beynon-Davies. Ethnography and information systems development: Ethnography of, for and within is development. *Information and Software Technology*, 39(8):531–540, 1997.
- [Bec99] Kent Beck. *Extreme Programming Explained: Embracing change*. Addison-Wesley, 1999.
- [BHR03] Sarah Beecham, Tracy Hall, and Austen Rainer. Software process improvement problems in twelve software companies: An empirical analysis. *Empirical Software Engineering*, 8(1):7–42, 2003.
- [BIT05a] BITKOM. *Daten zur Informationsgesellschaft: Status quo und Perspektiven Deutschlands im internationalen Vergleich*, 2005. <http://www.bitkom.org/>.
- [BIT05b] BITKOM. *Kennzahlen zur ITK-Branchenentwicklung*, 2005. <http://www.bitkom.org/>.
- [BK03] Bo Bergman and Bengt Klefsjö. *Quality: from Customer Needs to Customer Satisfaction*. Studentlitteratur, Lund, 2003.
- [Boe88] Barry W. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, 1988.
- [Bro01] Frederick P. Brooks. *The Mythical Man-Month*. Addison Wesley Longman Inc., New York, 15th edition, 2001.
- [Chu05] Christian J. Churchill. Ethnography as translation. *Qualitative Sociology*, 28(1):3–24, 2005.
- [CMO96] J. W. Carey, M. Morgan, and M. J. Oxtoby. Intercoder agreement in analysis of responses to open-ended interview questions: Examples from tuberculosis research. *Cultural Anthropology Methods Journal*, 8(3):1–5, 1996.
- [CN02] Paul Clements and Linda Northrop. *Software product lines: practices and patterns*. Addison-Wesley, Boston, 2002.
- [Con01] J. P. Conigliaro. Hybrid methodologies for small scale engineering. In *Change Management and the New Industrial Revolution (IEMC)*, pages 446–453, 2001.
- [Cre02] John W. Creswell. *Research design: Qualitative, quantitative, and mixed method approaches*. Sage Publications, Inc., Thousand Oaks, CA, 2nd edition, 2002.
- [CS01] A. P. Cater-Steel. Process improvement in four small software companies. In *Software Engineering Conference*, pages 262–272, Australia, 2001.
- [Cun02] Michael J. Cunningham. *B2B: How to Build a Profitable E-Commerce Strategy*. Perseus Books Group, 1st edition, 2002.

- [CW93] Geoff Cooper and Steve Woolgar. Software is society made malleable: The importance of conceptions of audience in software and research practice. In *PICT Policy Research paper no 25*, Swindon, UK, 1993. Economic and Social Research Council.
- [Daw00] Christian W. Dawson. *The essence of computing projects: A student's guide*. Pearson Education Limited, Essex, 2000.
- [DL99] Tom DeMarco and Timothy Lister. *Peopleware: productive projects and teams*. Dorset House Publishing Co., New York, 2nd edition, 1999.
- [DLSZ05] K.C. Dangle, P. Larsen, M. Shaw, and M.V. Zelkowitz. Software process improvement in small organizations: A case study. *IEEE Software*, 22(6):68–75, 2005.
- [DRL<sup>+</sup>05] Y. Dittrich, K. Rönkkö, O. Lindeberg, J. Erickson, and C. Hansson. Co-operative method development revisited. *International Conference on Software Engineering, Proceedings of the 2005 workshop on Human and social factors of software engineering*, 30(4):1–3, 2005.
- [Dyb05] T. Dyba. An empirical investigation of the key factors for success in software process improvement. *IEEE Transactions on Software Engineering*, 31(5):410–424, 2005.
- [EFS95] Robert M. Emerson, Rachel I. Fretz, and Linda L. Shaw. *Writing Ethnographic Fieldnotes*. Chicago Guides to Writing, Editing, and Publishing. The University of Chicago Press, Chicago, 1995.
- [Fet98] David. M. Fetterman. *Ethnography*. Sage Publications, 2nd edition, 1998.
- [For99] Diana E. Forsythe. “It’s just a matter of common sense”: Ethnography as invisible work. *Computer Supported Cooperative Work (CSCW)*, 8(1-2):127–145, 1999.
- [Fug00] Alfonso Fuggetta. Software process: A roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, pages 25–34. IEEE Computer Society, 2000.
- [GE04] F. Guerrero and Y. Eterovic. Adopting the sw-cmm in a small it organization. *IEEE Software*, 21(04):29–35, 2004.
- [GH95] D. R. Goldenson and J. D. Herbsleb. After the appraisal: a systematic survey of process improvement, its benefits, and factors that influence success. Technical report, CMU/SEI-95-TR-009, 1995.
- [GHWC05] G. C. Green, A. R. Hevner, and R. Webb Collins. The impacts of quality and productivity perceptions on the use of software process improvement innovations. *Information and Software Technology*, 47(8):543–553, 2005.
- [HA00] Martyn Hammersley and Paul Atkinson. *Ethnography : principles in practice*. Routledge, London, 2nd reprinted edition, 2000.
- [Ham90] M. Hammer. Reengineering work: Don’t automate, obliterate. *Harvard Business Review*, pages 104–112, July 1990.
- [Han00] Stuart Hannabuss. Being there: ethnographic research and autobiography. *Library Management*, 21(2):99–106, 2000.
- [Har00] R. H. R. Harper. The organisation in ethnography - a discussion of ethnographic fieldwork programs in CSCW. *Computer Supported Cooperative Work (CSCW)*, 9(2):239–264, 2000.

- [Har02] Brooke Harrington. Obtrusiveness as strategy in ethnographic research. *Qualitative Sociology*, 25(1):49–61, 2002.
- [HT05] O. Hazzan and J. E. Tomayko. Reflection and abstraction in learning software engineering’s human aspects. *Computer*, 38(6):39–45, 2005.
- [Hum97] Watts S. Humphrey. *Managing technical people: innovation, teamwork, and the software process*. Addison-Wesley, Boston, 6th edition, 1997.
- [Ins50] British Standards Institute. The tickit guide: Using iso 9001:2000 for software quality management system, January 2001 (5.0). <http://www.tickit.org/>.
- [Ins25] Software Engineering Institute. Vision of software engineering institute, 2004/11/25. <http://www.sei.cmu.edu/about/about.html>.
- [Ins03] Institut für Mittelstandsforschung Bonn. *Jahrbuch zur Mittelstandsforschung 2/2003*, 2003. <http://www.ifm-bonn.org/>.
- [Kin98] Nigel King. Template analysis. In Gillian Symon and Catherine Cassel, editors, *Qualitative Methods and Analysis in Organizational Research: A Practical Guide*, pages 118–134. Sage Publication, London, 1998.
- [Kru00] Phillippe Kruchten. *The Rational Unified Process: An Introduction*. Addison-Wesley, Boston, 2000.
- [LSS05] Timothy C. Lethbridge, Susan Elliott Sim, and Janice Singer. Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, 10(3):311–341, 2005.
- [Mal22] B. Malinowski. *Argonauts of the Western Pacific*. EP Dutton, New York, 1922.
- [MVM78] Peter K. Manning and John Van Maanen. *Policing: A View From the Street*. Goodyear Pub. Co., 1978.
- [NC00] N. J. Nunes and J. F. Cunha. Wisdom: a software engineering method for small software development companies. *IEEE Software*, 17(5):113–119, 2000.
- [NM90] Bonnie A. Nardi and James R. Miller. An ethnographic study of distributed problem solving in spreadsheet development. In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, pages 197–208, Los Angeles, United States, 1990.
- [NWZ05a] M. Niazi, D. Wilson, and D. Zowghi. A framework for assisting the design of effective software process improvement implementation strategies. *Journal of Systems and Software*, 78(2):204–222, 2005.
- [NWZ05b] M. Niazi, D. Wilson, and D. Zowghi. A maturity model for the implementation of software process improvement: an empirical study. *Journal of Systems and Software*, 74(2):155–172, 2005.
- [OMM<sup>+</sup>04] T. C. Ormerod, J. Mariani, N. J. Morley, T. Rodden, A. Crabtree, J. Mathrick, G. Hitch, and K. Lewis. Research methods in hci: Ethnography meets experimentation in image browser design. In *EHCI-DSVIS*, Hamburg, Germany, 2004.
- [RDR05] Kari Rönkkö, Y. Dittrich, and D. Randall. When plans do not work out: How plans are used in software development projects. *Computer Supported Cooperative Work (CSCW)*, 14(5), 2005.

- [Ric02] Jeffrey Richter. *Applied Microsoft .NET Framework Programming*. Microsoft Press, 1st edition, 2002.
- [RLD02] Kari Rönkkö, O. Lindeberg, and Y. Dittrich. 'Bad practice' or 'bad methods': Are software engineering and ethnographic discourses incompatible? In *Empirical Software Engineering, International Symposium on*, pages 204–210, 2002.
- [Rob00] Stephen P. Robbins. *Essentials of Organizational Behavior*. Prentice Hall, New Jersey, 2000.
- [Rus04] Terry Russel. Centre for ISO 9000, 2004. <http://www.iso-9000.co.uk/>.
- [Rön05] Kari Rönkkö. *Making methods work in software engineering: Method deployment - as a social achievement*. PhD thesis, Blekinge Institute of Technology, Ronneby, Sweden, 2005.
- [Sea99] Carolyn B. Seaman. Qualitative methods in empirical studies of software engineering. *Software Engineering, IEEE Transactions on*, 25(4):557–572, 1999.
- [SH05] Peter Stahlknecht and Ulrich Hasenkamp. *Einführung in die Wirtschaftsinformatik*. Springer, Berlin, 11th edition, 2005.
- [She97] Sarah A. Sheard. The frameworks quagmire. *CrossTalk: The Journal of Defense Software Engineering*, September 1997.
- [Som04] Ian Sommerville. *Software Engineering*. Pearson, Boston, 7th edition, 2004.
- [SR00] Helen Sharp and Hugh Robinson. Software engineering: Community and culture. *IEEE Software*, 17(1):40–48, 2000.
- [SRS<sup>+</sup>92] I. Sommerville, T. Rodden, P. Sawyer, R. Bentley, and M. Twidale. Integrating ethnography into the requirements engineering process. In *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*, pages 165–173. IEEE Computer Soc. Press, 1992.
- [SSS01] Susan Elliott Sim, Janice Singer, and Margaret-Anne Storey. Beg, borrow, or steal: Using multidisciplinary approaches in empirical software engineering research. *Empirical Software Engineering*, 6(1):85–93, 2001. <http://www.csr.uvic.ca/icse2000/>.
- [Sta05] Statistisches Bundesamt. *Bruttoinlandsprodukt 2004 für Deutschland*, 2005. <http://www.destatis.de/>.
- [SWHR99] H. Sharp, M. Woodman, F. Hovenden, and H. Robinson. The role of 'culture' in successful software process improvement. In *EUROMICRO Conference, 25th*, volume 2, pages 170–176, 1999.
- [Tal04] Susan Talburt. Ethnographic responsibility without the "real". *Journal of Higher Education*, 75(1):80–104, 2004.
- [Tea02a] CMMI Product Team. Capability maturity model integration version 1.1 continuous representation, 2002. CMU/SEI-2002-TR-028.
- [Tea02b] CMMI Product Team. Capability maturity model integration version 1.1 staged representation, 2002. CMU/SEI-2002-TR-029.
- [TGA81] Enrique T. Trueba, Grace Pung Guthrie, and Kathryn Hu-Pei Au. *Culture and the bilingual classroom : studies in classroom ethnography*. Newbury House Publishers, 1981.

- [VM00] John Van Maanen. *Tales of the field : on writing ethnography*. University of Chicago Press, Chicago, 2000.
- [Why93] William Foote Whyte. *Street Corner Society: The Social Structure of an Italian Slum*. University of Chicago Press, 4th edition, 1993.
- [Wil91] Paul Wilson. *Computer Supported Cooperative Work: An Introduction*. Springer, 1st edition, 1991.
- [WMM05] F. G. Wilkie, D. McFall, and F. McCaffery. An evaluation of CMMI process areas for small- to medium-sized software development organisations. *Software Process: Improvement and Practice*, 10(2):189–201, 2005.
- [Wol99] Harry F. Wolcott. *Ethnography: a way of seeing*. AltaMira Press, Walnut Creek, California, 1999.
- [Zah98] Sami Zahran. *Software Process Improvement: Practical Guidelines for Business Success*. Addison-Wesley, Harlow, 1998.