User Driven Software Development in a Small Company

Christina Hansson

University advisors:

Yvonne Dittrich, Department of Software Engineering and Computer Science
Dave Randall, Department of Human Work Science and Media Technology

Abstract

Field studies, interviews and participative observations have been the main methods in this master thesis where the author has studied, and partially participated, in the development process of a booking system called FRI. Idavall is the small company that develops this successful software despite small resources. Characteristic for Idavall is in the way they manage to involve their about 1300 users in the development process. The development process is highly flexible which is required when rapidly changes are common.

The overall question in this thesis is: How do Idavall manage to develop this software in a flexible way and successfully involve their users in the development process?

Idavall arranges FRI-meetings, courses and demonstrations where users meet each other and a representative from Idavall. All participants discuss FRI and have the opportunity to propose changes, improvements and report bugs. One of the most important parts of the business idea of Idavall is to offer a proper, friendly and professional support. The support is one of the most important possibilities to be informed about the users' needs, wishes and proposals. This, what I called, user driven development has a lot of common with Participatory Design (PD) where users participate in the design of new software. PD implies that users of software should take part in decisions that affect the system and the way it is used and designed.

Idavall make use of a kind of Customer Relationship Management (CRM), which is a concept used in the marketing arena. The objective of CRM is to create a strong, mutual and trustful relationship between supplier and customer. By using CRM in combination with PD it is possible to enhance the user participation and enable user driven development.

The company has a flexible development process, which is built on an informal way of driving the process. Informal meetings are common where decisions about changes and improvements are taken quickly. New versions are delivered about 20 times a year. The employees meet several times a day in the central located coffee room and discuss problems.

The development process has a lot of common with Extreme Programming (XP), which is a software development methodology that aims to make software development more flexible and focus on highly flexible environments with quickly changing requirements. XP is described briefly and comparisons are made to the development process of FRI as well as shortcomings of XP in the context of PD.

Keywords: Participatory Design, Extreme Programming, Customer Relationship Management, user driven development, flexible development process, FRI, Idavall

Table of Contents

Introduction

This master thesis will discuss how use-oriented software development in close cooperation with users can take place successfully in a small software company with small resources. Characteristic of this company is the way in which they manage to involve their (roughly) 1300 users in the development process. Another characteristic is how they develop generic software that is used in a highly flexible way through a flexible development process.

'This is not Participatory Design, this is something else', one of my colleagues suggested when I presented my field studies. I was astonished and started to think about his statement. What is 'something else' in this case then if it is not Participatory Design?

'Participatory Design (PD) is many things to many people' as J Greenbaum states, she continues 'yet there is a remarkable core to the ideas which have been built on common ground /…/ Computer applications need to be better suited to the actual skills and working places of the people using the systems /…/ The barriers between technical specialists and people using computer applications need to be broken down in order to build effective communication during the design process' (Greenbaum, 1993, p 27) PD implies that users of computer applications should take part in the decision that affect the system and the way it is used and designed. (Ibid) When I had read Greenbaum's explanation I became fairly sure about that 'something else' could be thought of as PD in this study. PD is not enough to develop a system that will suit many different users during a long period of time and survive a lot of changes. A highly flexible development process is needed as well to fulfil those requirements.

After the introduction, background and methods I will present parts of my field studies at Idavall in the section called 'User participation in the development process of FRI'. Idavall is a small software company which develop a booking system called FRI. I will go deeper into what is PD and relate it to my field studies. I will further describe Customer Relationship Management (CRM), which is a concept used in the marketing area. When the competition gets tougher, tying customers to a company becomes more difficult. Therefore, it is important to create a strong relationship to keep and gain customers. Idavall uses CRM in a way to develop their relationship to their customers and establish a trustful relationship by close contact with them. By using CRM in combination with PD is it possible to enhance user participation and enable a user driven development. The first part of the thesis includes a brief description of the most important methods I have used during my field studies as well.

In the chapter, 'The development process at Idavall' I will describe the development process and architectural structure of FRI. The development process has to be highly flexible since system requirements are continually changing. Flexible processes and computer systems become more and more important because the needs, the use, the users and the organisations where computer systems are used are continually changing and demand a fast adaptation to new conditions. The complexity of the surroundings makes it impossible to anticipate these changes. (Henderson, Kyng, 1991) How, then, do Idavall manage to be that flexible and still have control over the system? I have noticed that the development process has a lot of common with Extreme Programming (XP). XP consist of a number of fundamental practices, activities, values and principles that not are new in software development process but the combination and packages of them are new. (Karlström, 2002) XP stresses, for among other things, user participation, small releases and short iterations to make the development process flexible. XP is mainly used in in-house software development processes. I will make some

comparisons between XP and the development process of FRI in order to point out some important similarities.

In the last part of this thesis I will tie a number of themes together, and point out some important issues that are cornerstones in the development process of FRI; user driven development from a distance, flexible development processes and the role of XP in the context of PD. These issues have been the focus in this thesis as well as the overall question; how do Idavall manage to develop FRI in a flexible way and successfully involve their users in the development process?

In the following I will use the word 'user' when I am writing about those who really are using FRI. 'Customer' is used when I mean those who have the responsibility to buy FRI. For example: The municipality of Sölvesborg is a customer, but Lisa who is working at the one-stop shop is the user of FRI, she has the opportunity to give proposals about new functionality but not to decide whether Sölvesborg would buy a new part of FRI. Sometimes the customer and user is the same person, especially when it is small associations or organisations.

Background and Methods

When I was about to start thinking of an interesting topic for my Master Thesis, Yvonne Dittrich at the Department of Software Engineering and Computer Science told me about Idavall. She has established a contact with Idavall through another project named *Design of IT in use – supportive technologies for public services* (DitA)[1] in which she takes part. Sara Eriksén at the Department of Human Work Science and Media Technology is the head of the project. Idavall develops a niche application for booking locales (conference rooms, football fields etc.). This application is especially appreciated by one of the participants of the DitA-project. The reason why is that they feel that this application is the most useful, functional and well designed application they use, at the same time as the support they receive is also working very well. They are able to have an impact on the application and thus participate in the ongoing development process.

Flexible software where the users have the possibility of participating in the development process through a kind of design in use is of great importance. Design in use renders a continuous development by adapting the software to prevailing circumstances. Circumstances are continually changing in, for example, municipalities where rapid changes are common. (Dittrich, et al., 2002)

In-house support is often missed in small organisations and municipalities. (Eriksén, 1998, Ekstrand, Hansson, 2001) A well functioning support structure is therefore an important opportunity that Idavall offers to their users.

I thought it sounded interesting and I thought it could be a good topic for my thesis to find out why Idavall and their application and support are so appreciated.

Idavall

Idavall Data AB is a small software company founded in 1987 located in Emmaboda in the southeast part of Sweden. Idavall has six employees of whom one is on the long-term sick list. In the first years Idavall was developing a number of different programs, but from 1991 they mainly develop FRI. Those who are working at Idavall are:

Gustaf who is responsible for marketing and selling; Torbjörn who is responsible for the technical part of FRI and who is the chief developer; Lotta who is responsible for the ASCII-generator and she develops as well new templates for different kinds of reports; Janne who is responsible for the development of FRI-applications to be used on the Internet; Mari who is the secretary, and responsible for the switchboard and other important practical tasks and Jasper who is on the long-term sick list. All of them take part in the support of FRI.

Idavall has about 300 customers and 1300 users around Sweden, Finland and Norway. The customers are mainly municipalities, sports facilities and some companies and museums.

Methods

To get an understanding and knowledge of how Idavall manages to develop an application like FRI where a lot of the users are involved in the process I had to use different kinds of methods. I participated in different activities where users were involved. The most important method during this study were interviews with the employees at Idavall and users I met during different activities I attended. I have also passively listened to a number of support calls.

---

[1] Funded by the Swedish Agency for Innovation Systems VINNOVA. The partners are five municipalities, two software consultancy firms, a Call Center and researchers from the Blekinge Institute of Technology

Participative Observations

One of the most important ways to collect data is to watch and listen to people's activities and to participate in different activities. To participate may mean to fully participate, e.g. to live and behave like a member of a group during a period or to act like the 'fly on the wall'. To act like the 'fly on the wall' means to be in the background and look at the activities from aside. Most of the researchers take participation to mean a course of activity somewhere in the middle of these extremes. (Ely, 1991) I participated in a FRI-meeting, a training course and in a demonstration of FRI. I participated as a full member in two of the activities (FRI-meeting, and FRI-demonstration) and as an observer in the training course because there was no computer available. At some times I acted like the 'fly on the wall' and just listened to the conversation when, for example, a support call was conducted.

Another kind of participation at Idavall was to participate in coffee breaks and weekly Monday meetings. During those meetings a lot of information was shared and design issues were discussed as well as proposals from customers.

Interviews

During the field studies I have used informal and formal interviews. Interviews that are performed directly during the participative observations are informal. They come up in a situation and are not prepared. (Ely, 1991) My informal interviews have taken place at for example coffee breaks at the FRI-meeting I attended and in other activities I have participated in. I asked, for example users about their opinion about Idavall and FRI and why they have chosen Idavall as their supplier. At Idavall I have used informal interviews after, for example completed support calls, and when I "hung around" I conducted spontaneous interviews when I felt that I did not understand or had questions about what I saw. I have asked questions in the light of the situation at hand.

Formal interviews are more prepared and are often performed in a special place where the observation does not take place to be able to talk without interruptions. (Ely, 1991) My formal interviews have taken place at Idavall in a special room or in the office where the work takes place. It was not always possible to do the interviews in an undisturbed place, because the employees had to be close to the telephone to be able to answer the phone if someone needed support. I had some prepared questions, which were followed by unprepared questions. Those formal interviews were taped and then transcribed. (Jordan, Henderson, 1994)

Field notes

A lot of my field material consists of field notes. Because of my long period of presence it was not possible to tape all situations. Transcribing is time consuming and it is not always dependable. Field notes are a good way to capture what is going on in a surrounding. It is important though to immediately write down your observations when you come back to your office. It is as well an opportunity to reflect over what you have seen and complete your field notes with additional comments. Your field notes contain your facts and it is only what is registered in your field notes that are available for research apart from your videotapes and audiotapes. (Ely, 1991)

FRI

FRI is one of the most used booking systems in Sweden. FRI is used in, for example, sports centres, associations, conference halls and hospitals throughout Sweden, Norway and Finland. However, the most important customers are still different Swedish municipalities.

History of FRI

The first version of FRI was developed in 1987. It was specially designed for a certain customer. This version is still the base of today's FRI. The early versions of FRI were designed to be used in a DOS-environment.

In the end of the 80´s Idavall wanted to develop FRI for Windows. They started to look around for a suitable tool for building FRI for Windows. They tried a Norwegian development tool, Winnix, that was especially developed to be used together with their previous database FHS - File Management System. They were even looking for an alternative database but did not find any that suited their needs better than FHS. They kept FHS but chose Visual Basic (VB) 1.0 instead of Winnix. Today they use VB 3.0.

The first version of FRI for Window was released in 1991 when two installations were delivered; in 1992 it increased to 25 new installations. Today, FRI has about 300 customers and 1300 users. In parallel the DOS-users changed gradually to the Window-version and in 1994 the last user of the DOS-version of FRI began to use FRI for Windows.

The first version of FRI with web user interface was released in 1999. FRI for the web is structured in almost the same way as the foundational FRI. To run FRI on the web you have to have the original FRI installed because FRI for the web uses the information from the same database as the original FRI. Once you make a change in your foundational FRI the same change occur in the web version. The programming languages are Visual Basic script, ASP and HTML. Janne is responsible for the development of FRI for the web.

For further information about the development of FRI, please read the section 'Software Development at Idavall' later in this thesis.

Different parts of FRI

FRI consists of a number of different program modules. (Fig1) The customer is able to buy those modules he is interested in and is not forced to buy them in a special order. The modules can, in principle, be installed in any order and in different configurations. The base of FRI, which is mandatory to be able to run the system, consists of modules for authorization, administration of the register of customer, a report-generator, the system settings and finally an ASCII-generator, which creates export files. Most of the customers buy the object register as well to be able to administrate bookings. All data is stored in a database, which is the heart of FRI. I will briefly describe some of the parts of FRI. Those who are interested of more information about FRI, please visit www.idavall.se.

Fig 1 Menu of FRI

FRI schedule (Fig 2) shows booked and bookable positions in a plain way. Different types of booking are marked with different colours. The colours make the schedule clearer. Besides the short names of the customer and booking number and additional information can be displayed in the schedule to make the view even easier to grasp.
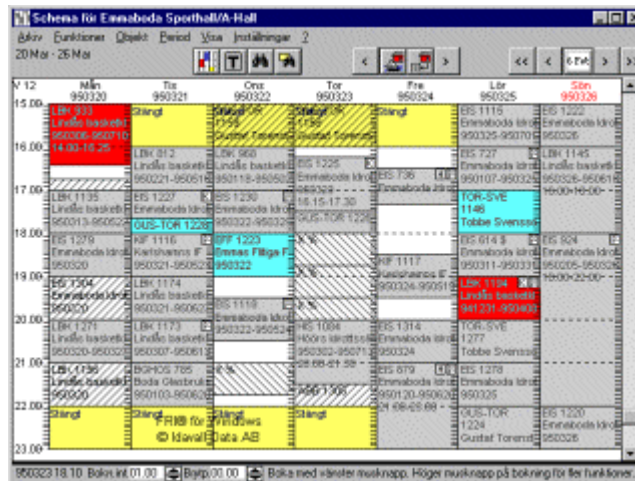
Fig 2 Schedule

The 'Booking in' schedule is one of the biggest advantages and strengths of FRI. The schedule is also the central interface. Clicking on a timeslot lets an editor window pop up. The user is able to get an overview of free and occupied positions. One chooses between several kinds of schedules, for example, one-week schedule, several-days schedule and several-weeks schedule.

The register of customers is the foundation of FRI (Fig 3) as I have mentioned. It consist of, among other things, personal data of the customer, information about what kind of

customer it is and so on. It is easy to search the register. Idavall allows the user to import previous registers of customers into FRI.
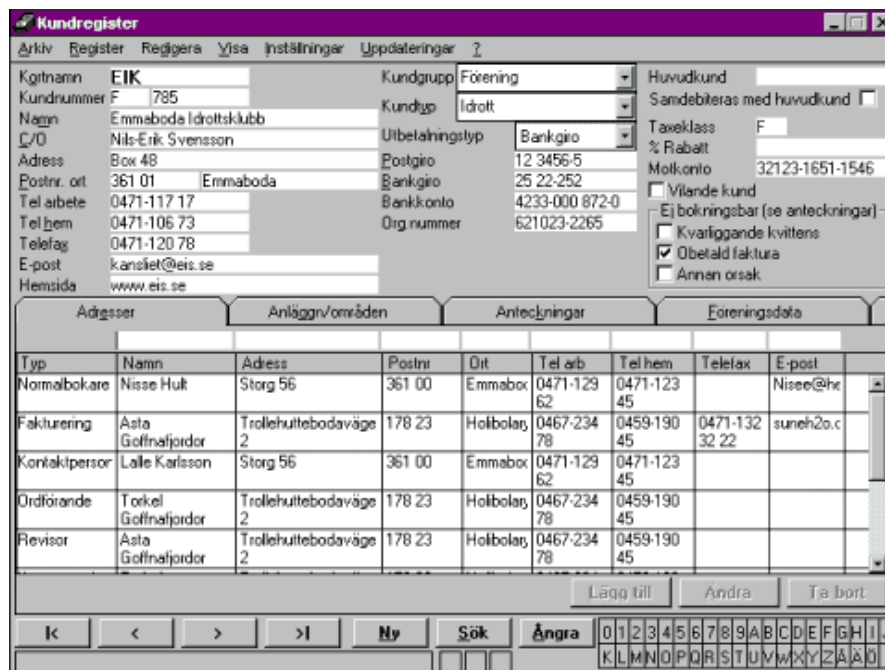
Fig 3 Register of Customer

'Debiting and compilation' is another feature in FRI. With the help of a ASCII-generator, it transfers files to different economic systems. The ASCII-generator is a homemade tool. It creates files that allow further processing of data by other systems. The ASCII-generator is built up round the fact that every supplier of computer systems has their own formats for files. So far, Idavall has managed to integrate all the formats they have encountered.

'Web-association' is a fairly new part of FRI. It helps the user to present a register of associations of the municipality on the Internet. Web-association gets the information that will be published in the register directly from FRI, which means that it is always up-to-date. FRI has to be installed as a foundation to be able to run Web-association.

Customisation of FRI
As I mentioned earlier, the first DOS-version of FRI was released in 1987. Every customer had a custom made and unique version in that time. When FRI for Windows was released in the beginning of the 1990's every customer used the same version of FRI as a foundation. All parts of FRI are delivered and installed at the customer's place at the first time. The customer buys licences to those parts he wants to use and the other parts are not available. This way of distributing FRI makes it easy to extend the use of FRI if the customer later wants to add a new part.

A representative from Idavall installs FRI at the customer's place. The representative helps the customer to customise FRI so it will suit his needs. For example, if the customer wants to use FRI for booking conference rooms, all rooms have to be organised in a proper way in FHS. Every room has to have a fee connected. It has to be decided when the room is available to book and so on. A lot of adjustments have to be made before FRI is ready to be used.

7

When FRI for Windows was released there were a lot of new development in the first couple of years. Since 1997 it has mostly been refining and no new big changes have been carried out. The latest part, the admission system was released in 1996. About 20 small releases are delivered every year. A small release might be a new text field or template added into the report generator, or more opportunities to customise FRI. New releases are distributed on the web site of Idavall and by mail to customers who do not have access to the Internet. Every release is described in a 'New' description. It describes what kind of changes have been implemented and which allow the user to choose himself whether the newest version is one he needs to download or not. A small release is only used to update the program parts, not the database. The intention is that it will be so easy to download and update small versions that every user is able to manage it. Once a year or when it is necessary, a major release is delivered. A major release has to be downloaded to the database and program parts, the workstations have to be updated as well. A major release can, for example, imply a new version of the database.

This part of background and methods was meant to be a help to make it easier to understand the following parts. It can be of interest to know a little bit about the functionality of FRI and the history behind the program. In the following part, I will describe among other things meeting places where Idavall meets its users, discuss the nature of Participatory Design, the idea of User Driven Development and Customer Relationship Management. I will try to unify these three concepts in the discussion of user participation in software development in small companies.

User Participation in the Development Process of FRI

Support and meeting places for and with customer and users
The first DOS-version of FRI was designed from the beginning to fit a certain customer/user. Still, the objective of Idavall's is to listen to their customers/users and develop FRI in a way that maintains satisfaction. Gustaf expresses his standpoint at a demonstration of FRI as: "The development is driven by our users not by ourselves". To be able to let the users drive the development it is important to listen to the users' needs and points of view. This is possible when representatives from Idavall meet their users in different kinds of activities. I will in the following describe those different activities.

Demonstration of FRI
Demonstrations of FRI are a way to meet prospective customers. Demonstrations are regularly held all over the country and prospective customers are invited. Idavall gets in touch with customers through advertisements in different types of technical papers, trade fairs and through existing customers. Gustaf, who often is responsible for those demonstrations, has nearly always been in touch with the customers before the demonstration and knows a little bit of their needs, technical background and expectations of FRI. Because of his knowledge of the background of the customers´ he is able to adapt the demonstration so it will fit with the participants.

During the demonstration, Gustaf explains those parts of FRI that are of interest to the customers. He also talks about forthcoming features and discuss what needs that might be of interest in the future. If someone has suggestions Gustaf writes down a note on a piece of paper and takes it back to Idavall for further discussion and investigation. If it is possible to develop and it seems to have an important impact on FRI, Torbjörn or Janne will implement it in a forthcoming version.

It happens that Gustaf tells a customer that FRI is not the best application to fulfil his needs. The customer might be angry, but Gustaf prefers being honest in the beginning to having a dissatisfied customer later. A dissatisfied customer might circulate bad reputation and costs a lot in support. One example was an owner of a squash-hall; he wanted FRI for the Web to let his customers book squash courts on-line themselves. This is possible in the application 'FRI for the Web/booking'. However on-line booking requires a permanent connection to the Internet. This was a problem for the owner of the squash-hall. It would be too expensive, according to Gustaf, to have a permanent connection to the Internet. It seemed that the potential customer had not thought of that and was angry at first when Gustaf told him about the problem. After a while, when Gustaf had explained the problem in a convincing way, he understood the problem. Gustaf offered to check out the possibilities of joining a web-hotel and/or a company-hotel and in that way solve the problem.

Support
One of the most important parts of the business philosophy of Idavall's is to offer a proper, friendly and professional support. The support, besides the FRI-meetings, offers one of the most important methods of staying informed about users' needs, wishes and proposals. The objective is to talk to the user precisely in the way that users ordinarily talk, avoiding technical jargon. No one should feel like they are stupid or crazy when they call Idavall for support. Support is given every day between 8 am to 12 am and is closed in the afternoon. Emergency

support is offered in the afternoon as well. Mari who is in charge of the telephone switchboard distributes the support phone calls to the individual who is available. If special knowledge is required she directs the call to the respective expert. Usually, users have their own favourite they want to talk to.

An agreement of support is normally signed at the same time as FRI is installed at the customer's place. The agreement lasts as long as the customer wants to and can be terminated at any time. The agreement of support covers the following topics:

Support: Idavall supports the bought version of FRI and later versions. Support is given by telephone, e-mail, telefax or by letter.
Upgrading: Upgrading is managed by floppy disks via e-mail, file transfer via modem (from the web site of Idavall) or by installation on site through Idavall. The upgrading includes the generic features and does not include specific customer related issues. If a customer wants a individual solution it will cost an extra amount of money.
FRI-meetings: FRI-meetings are included in the agreement of support.
FRI-letters: FRI-letters will be described below.

Some rules are connected to the agreement of support. The agreement for support is free for the first six months. After six months the support fee is 12% of the purchase price but not less than 1000 Swedish crowns/year. If a customer buys additional licences it will increase the annual fee for the next year.

Support by telephone is the most common way to give support. All of the employees at Idavall participate in the support service in one way or another. All of them are familiar with the basics of FRI and give support on that. Gustaf says that this is an advantage. It does not matter if someone is away because everyone has the knowledge to give proper support. In addition to the basic knowledge of FRI, each member of Idavall has a special knowlegde area where he or she is a specialist, Janne for example is responsible for the web application, and he is the first person to take questions about the web. Lotta is responsible for the ASCII-generator and supports question about that and Torbjörn has the comprehensive knowledge of the technical part of FRI and is called when pure technical questions pop up. In case the call taker cannot solve a question, he or she hands it over to the expert or asks if it is possible to call back later. The objective is always to answer the question in a clear and honest manner. Sometimes staff from Idavall call the user and ask whether the problem has been solved to make sure that everything worked out.

Almost every phone call to the support service is logged in a text database, the 'telelogg'. Every user and customer is present in the 'telelogg'. When someone calls in, the call taker at Idavall searches for the caller's name in the 'telelogg'. If the person is not present he/she will be added. The 'telelogg' contains among other things; user name, customer name, which version of FRI that is used, previous problems and how these were solved. During, or directly after the phone call the call taker inserts notes about the call. What was the problem? How was it solved? and how long was the call? All information is searchable within the 'telelogg', which makes it easy to search for a problem and get a hint concerning the way the problem was solved in other cases.

I will in the following give an example of a phone call to the support service to make it easier to understand how it might work. The example illustrates a typical user problem which prompts a call. In this case, she could not print out an order.

The user calls, and Janne will be in charge for this call. Janne starts to talk in a relaxed way to the user. How is life, did the common football team win their last game. After a while, they start to talk about the real problem. The user has a problem when she wants to print. Janne suggests her to try different things on her computer and printer. In parallel Janne is doing the same steps on his computer. Janne pilots her through different possibilities to solve her problem and tries to understand the problem at the same time. Janne opens the telelogg and searches for a similar problem to find, how it was solved that time. In our case it turned out that the user had to contact her local technician because Janne thinks it would be a local network problem. It is fairly common that printer problems depend on local network problems. Idavall is not responsible for solving those problems, if they can, they do it anyhow, it is a way to be service minded, Janne says. After a couple of hours Janne called the user and asked if the problem was solved to make sure that everything was all right.

Often, support calls generate proposals for new functionality. It might, for instance, be a wish for an extra text field to be added to be able to fill in mobile phone numbers. Another proposal can be to increase the opportunity to change different settings in the web application. The supporters make a note of those proposals and wishes. They almost never promise anything. They want to discuss with the others issues such as whether it is a good proposal; if it is possible to implement it, if it is a proposal that can be an advantage to many users and if they want to implement it (e.g. if they think it is fun to implement this proposal).

I have talked to several users who use the support service. All of them think that the support of Idavall is excellent. One participant at the FRI-meeting in Umeå expressed his feelings about the support like 'The support of Idavall's is marvellous'. The users feel like they are important and mean a lot to Idavall. No one I spoke to was able to identify any other company support that is like Idavall's. Some users say that the employees at Idavall feel like friends and they can talk to them about things that are not necessarily a problem of FRI. They also mention that they do not feel stupid when they call Idavall, a situation they encounter when they call other suppliers. They think it depends on the manner in which the employees at Idavall speak; their language is simple and understandable even if they, sometimes, talk about difficult subjects. A proof of appreciation was when a user brought a present to Lotta at Idavall because she has been such a help to the user for a long time. They had never met before but it looked like they had known each other for a long time.

FRI -meetings
Every year about 8-10 meetings for existing customers are held throughout Sweden and Norway. The purpose of these meetings is to meet in a relaxed form and discuss FRI, news, future and further development and answer questions together about FRI. FRI-meetings are a way to meet other users of FRI that are located in the area. FRI-meetings offer an opportunity to create networks that makes it easier to get in touch with each other and thus cooperate around common questions or problems.

The meeting starts with a presentation of the participants. Gustaf, who often is responsible for FRI-meetings, is familiar with everyone who attends the meeting and acts in a personal way. He knows in what way the customers use FRI and about possible problems of the

users. During the meeting Gustaf starts to go through the web site of Idavall and explains the structure of it. Gustaf continues by talking about FRI, what is new and he also pushes for some features that he knows the users do not utilize to a great extent. These features, for example, templates for writing reports and confirmations of bookings are not applied as much as they should be according to Gustaf. Gustaf means that the users should apply these templates to create professional printouts instead of using home made versions.

FRI-meetings are also a meeting place to discuss different solutions to problems. Gustaf discusses together with the participants and gives advice concerning how to solve diverse problems.

Example 2
One of the users wants to be able to erase date from occasional bookings because it is duplicated information she thinks, as the date is present at the current day as well as in the booking itself. Gustaf understands her problem but explains to her that it will slow down the system if this feature is implemented. He shows her a way to erase the date through making changes in the settings. To erase dates in this way is on the other hand a disadvantage because the dates of all bookings will disappear. Anyhow, Gustaf makes a note of the customer's proposal and brings it back to Idavall for further discussions with his partners.

Example 3
Another user wants to lengthen the text field of the bank account number because the last figures in the number disappears if the number is to long.. He says that he has to write the number by hand on the invoice. He suggests that it should be possible to adjust. Gustaf promises that this proposal will be implemented because he knows that it ought to be an advantage to other customers as well and it will be easy to implement.

Every FRI-meeting has its own link on the web site of Idavall where participants, proposals and possible discovered failures are documented. A photograph of the participants is included as well. Idavall uses the proposals when developing FRI.

To attend a FRI-meeting is free; it is included in the agreement of support. The host customer books the premises and meals and the representative from Idavall brings his own equipment like projector, portable computer, papers to make notes on, pencils, sweets and so on.

FRI-letters
FRI-letters are a kind of information that is sent out for about ten times a year. All registered users get a copy of it via e-mail. The FRI-letters inform customers of, for example, new versions of FRI, previous and forthcoming training courses, reports from FRI-meetings, what is coming up, what is going on and so on. The FRI-letters are a way to keep the users up to date with FRI and different activities.

Courses
Idavall offers different kind of courses like a basic course, which focus on the basic functionalities of FRI; the booking course, which has a specific focus on how to manage different kinds of bookings; the Web association course, which focuses on how to administrate and

create a nice catalogue of associations on the Internet. Another course is the system administrator course, which teaches how to administrate FRI. This course is more technical than the others. How to create reports, confirmations of bookings and so on by the way of templates is taught in the report generator course.

The courses last for a half or a whole day. The customer has to pay a fee to participate in the courses. Mostly, courses are held at Idavall, but it is possible to order a course at the customer's place as well.

Every employee, except Mari, at Idavall has the responsibility for at least one kind of course. In the responsibility is included planning and teaching, and developing tasks, course materials and manuals.

I attended a web-association course to watch and listen to how it was carried through. Janne who was in charge for this course, had prepared the course as described above. The course started at the local restaurant where we had lunch. All the employees of Idavall participated in the meal even if they not were involved in the course. It is a part of the business idea of Idavall's to know their users. Even if the course was a web-association course, all of the users use FRI as the basic application and it is possible that the users will call the support about problems related to FRI not only about problems related to web-association. In that case is it possible that anyone will be in contact to the customer in the future and that is why it is important to know the customers.

Eight users participated in the course, which is the maximum. Janne encouraged a dialogue between him and the users; he encouraged them to ask questions and in that way created a relaxed environment. He does not want to be a traditional teacher who tells the truth and right answers. He also tried to joke a little bit, even if it was a serious course, to make the participants to relax. During the course Janne went through Web-association talked about functionalities and showed how to do different set-ups to make the application fit to the user. Janne encouraged the users to call him if they run into problems after the course. During the course the participants came up with proposals, one wanted, for example, to have a text field to be able to fill in the phone number of the office in the register of the associations. Janne told the user that it was already implemented and showed it to the user. Another wanted a different formulation of text on the information page because the present formulation is a bit unclear. Janne discussed this proposal together with the other participants and agreed on a new formulation. This was a proposal he estimated he does not have to discuss with the others at Idavall.

Some of the users told me that they think it is important to participate in courses. It is not only because they learn a lot but even that they get to know other users of FRI. If they know other users is it easier to call and ask questions about things that they know they have solved before. They also think it is fun to come to Emmaboda and meet the staff at Idavall's. It becomes easier to speak in a phone when you know the face of a person, they said.

People at Idavall care for their customers and encourage them to contact Idavall as soon as they have problems or want to propose a change or improvements. Idavall is also committed to creating meeting places where users and representatives from Idavall have the possibility of meeting each other. These meeting places are important arenas where Idavall can acquire knowledge about the users. Gustaf talks about user driven development. It sounds like Participatory Design. But Participatory Design in the literature 'looks' different. The question is consequently; what is Participatory Design?

What is Participatory Design?

I want to in the following, among others things. I give an account for what is written about traditional PD in the literature. It is not meant to be an exhaustive description; it will just give the reader an overview over what is relevant for the future discussion.

PD researchers are concerned to explore conditions for user participation in the design and introduction of computer-based systems at work. A core concern is to understand how collaborative design processes can be based on participation of the people affected by the technology designed. (Kensing, Blomberg, 1998)

Why PD? The political background

PD was developed in Scandinavia in the mid 1970's as a response to employers' efforts in computerizing industrial work processes. Traditional PD was when it first was introduced meant for unionised traditional organisations. One of the aims was to prevent the introduction of computers at the workplaces, by providing the employees with structural possibilities and competence to influence their own work situation. Another aim was to influence what computer system should be introduced. The belief was that users' knowledge would improve the fit between the computer system and the work. Scandinavian research project in system development have traditionally put a strong emphasis on user participation as a strategy for increasing work life democracy. Increasing work life democracy means that those who are affected by a decision take part in the making of the decision as well as balancing claims from different stakeholders. Scandinavian approaches to system development have been characterized as user-oriented rather than management-oriented. (Bjerknes, Bratteteig, 1995)

From the very beginning, PD researchers have been explicit about their concern regarding the politics of system design and computer-based systems. It relates to the distribution of power at the workplace and possible negative effects on employees, for example de-skilling. The employees saw that much of their work was being de-skilled and decisions that once were under their control were either being automated or moved higher up the organisational hierarchy. They feared that ultimately this would lead to workforce reductions. PD researchers argued that computers were becoming yet another tool of management to control the employees and that these new technologies were not introduced to improve work conditions. Employees and their unions were concerned about the possibility that computers would take over the control of their work situation as well as the planning and administration of production. (Kensing, Blomberg, 1998)

In spite of successful projects where employees got an increased bargaining power employees continued to find it difficult to argue for alternative ways of using technology. Partly because management's goal and strategies were often built into the new systems and were reinforced by organisational distributions of power, making it difficult to alter the technology to fit employees' needs and interests. Researchers and employees became interested in determining if it would be possible to design, develop and implement technologies which took as their starting point the needs and interests of employees. (Kensing, Blomberg, 1998)

PD and usability

Since the 1970's computer based systems have become more and more integrated into people's work lives. Many design professionals and managers are realising the importance of taking the skills and experiences of employees into consideration. The employees need to take part in the design and organisational implementation of computer systems and the work

they support. Design professionals and managers argue that the participation will help to ensure a better fit between technology and the ways people wish to perform their work. (Kensing, Blomberg, 1998, Bjerknes, Bratteteig, 1995) The assumption is that if a computer system fits the work it would improve his/her work situation. The process of developing the system needs to be influenced by the employees in order to get a good and useful tool. It has been found that employees who are allowed to influence their own work situation are more efficient and take more responsibility. (Hallberg et al., 1998) The participation helps users to increase their skills and thereby increase the quality of the service they provide. Users who participate in the design process are also expected to be more willing to accept the final system once it is introduced. They have more indulgence towards possible failures and shortages. (Vimarlund, Timpka, 1998)

It assumes that employees themselves are in the best position to determine how to improve their work and their work life. In doing so, it turns the traditional designer-user relationship on its head, viewing the users as the experts – the ones with the most knowledge about what they do and what they need – and the designers as technical consultants. Another objective for participating in the process of PD is that it gives the designer new and better ways of gaining an understanding of the user's everyday work practices. (Shuler, Namioka, 1993) Design professionals need knowledge of the actual use context and employees need knowledge of possible technological options. These types of knowledge are developed most effectively through active cooperation between employees and designers. (Kensing, Blomberg, 1998) The computers should be understood in the context in which they are used and therefore be designed as instruments for work. The basis for design should be the knowledge needed to maintain daily work routines rather than production routines. The design process is closely tied to a concrete work situation and it is important to take into consideration the tacit knowledge of the employees and the implicit, shared understanding. (Bjerknes, Bratteteig, 1995)

Expertise from both the work place and designers' domain are needed in system development projects. Mutual learning is of great importance. Both users and designers need knowledge about each other to communicate and learn from each other in order to develop a useful application that fit the work and daily routines. (Bjerknes, Bratteteig, 1995)

Organisation of PD
The way that PD is organised is of very great importance. In many PD projects is it not possible for all who are affected to fully participate. In these cases the choice of user participants and the form of participation have to be carefully considered and negotiated with relevant organisational members, including management and the employees themselves. The design professionals would describe what kind of expertise that is needed, for example employees with particular skills or representatives from various groups. Alternatively the union may identify project participants. In making these choices is it important to be clear about the motivations for participation, the scope of participation, and the resources allocated for the project. (Kensing, Blomberg, 1998)

Most PD activities are time consuming and require the presence of resources - time and money, and negotiation structures to allocate these resources. Participants will only be motivated to contribute to the process if they receive compensation for their work and if their ordinary work will be done while they are away. To be able to meaningfully contribute to the process the participants 'might have to get qualifications', which also requires resources.

(Törpel et al., 2002) The relations between those who are taking an active part in the project and those who do not should be carefully considered and attended throughout the project. A risk is that those who are not directly involved in the project will not get a feeling of being affected. It is therefore important to establish appropriate relations with other members of the organisation during the project so that technological and organisational changes are communicated beyond the immediate project group and last after the researchers leave. (Kensing, Blomberg, 1998)

One way to get a sustainable PD practice after the research project is finished is to establish a PD project with internal design practitioners. Working with design practitioners, researchers strive to incorporate PD within the organisation by developing locally articulated design practices and guidelines. This will help the internal designers to integrate PD into their work practice. (Kensing, Blomberg, 1998)

PD methods

Researchers and practitioners developed a set of methods and tools to mediate participatory processes. Most of the methods have been developed in unionised contexts, like for example the graphical industry and hospitals. This section will briefly describe some common PD methods, which have been successfully applied in practice. Later in this thesis I will discuss why some of the traditional PD methods are hardly applicable, without modifications, in distributed environments and small companies, like these where the developing of FRI takes place.

Major parts of the table below derive from Törpel, Wulf, Kahler, 2002.

Visiting workplaces: As a means of enhancing real-life knowledge about the organization, the workplaces, the work practices and the existing problems, interest, and conflicts, participants in PD processes spend time with members of potentially affected work environments, engaging in activities that seem suited, like observing, inquiring, interviewing, or scrutinizing relevant artefacts.

Elaborating work: Different typical work practices are reconstructed and integrated scenarios into an overall picture of the work tasks, the division of labour, and the cooperative structure to be supported.

The future workshop: A method to gather requirements in participatory processes for the design of computer applications at work. Future workshop is a method where the participants are able to discuss a specific and concrete problem. The aim is to help the participants to focus on the concrete reality without thinking of economic and technology. (Kensing, Halskov Madsen, 1991)

Using Mock-ups: Mock-ups are noncomputer artefacts, but as means for designing, computer-related artefacts they serve to imagine computer system functionality. They allow for hands-on experience, are easy to understand and inexpensive, enhance imagination of future solutions and are fun to work with.

Cooperative prototyping: As revisable technical realisations of the functionality represented in the mock-ups, prototypes can serve as means for cooperative experimentation and modification with the potential functionality and realisation of the prospective system.

Continued design of computer applications, work practice and organisation: Often, continued design is equated with tailoring as adapting or modifying technical solutions in use. Tailorable systems leave space for a variety of appropriations by providing the users with op-

tions to further design them in use. The ongoing improvement of means, processes, structures, rules, and so on can be described in terms of spiral-like change processes, often called evolutionary (Floyd, Reisin, Schmidt, 1989) The affected organisation members receive the opportunity to suggest and/or make changes in the form of feedback loops instead of being subject to linear processes (like the sequence of requirements, definition, design, implementation, test, introduction, use, maintenance).

PD in other contexts
Most of the methods described above are time consuming and thus expensive for the host organisation. Small organisations and companies often have not the opportunity to carry out a project where users and designers work together over a long period to develop a system uniquely suited to the tasks, practices and environments of its users'. Small organisations and companies often select the best-fit technology from what is available in the marketplace and adapting it to the local conditions. This has been recognised by researchers in software development as well. (Törpel, et al., 2002)

Törpel et al. 2002, consider user participation in small companies which do not have the resources and infrastructure to participate in the development of their own software. User participation in small companies focus on issues as shopping decisions, the compatibility, tailorability and reliability of off-the-shelf applications. Most of these small companies want to be as participatory as possible in the development, introduction and appropriation of processes in their computer applications. The principles of traditional PD approaches can guide participatory shopping and tailoring. In small companies is it possible for individual interests, perspectives, and work constellations to be taken into account and for everybody to have an impact on core decisions.

Fischer and Ostwald, introduce a notion, *informed participation* that can be regarded as an extension of PD and is not tied to one work place. Informed participants work in different work places but have problems which are comparable to those of users of FRI. Through discussions and mutual learning they construct new knowledge. The participation shifts from designing a system to using and evolving it.

> *"Informed participation involves a community of interest made up of people from several backgrounds, each having a unique stake in a common problem /.../ Informed participation shares many objectives with participatory design, which aims to involve users in the design of artefacts they will use /.../ Informed participation begins where traditional participatory design of an system leaves off, and extends into the system's lifecycle as the focus of participation shifts from designing a system to using and evolving it." (Fischer, Ostwald, 2002, p 136)*

If conditions are adapted to fit employees in small organisations and companies they will be able to take part in different PD activities. That means that they will gain more influence in the development of computer applications in their organisations.

User Driven Development at Idavall
In this part of the thesis I will discuss the way in which FRI is developed and relate it to traditional PD research and literature.

Historically, PD has focused on system development at design time by bringing users and developers together to co-operatively design or redesign the new or present system. But despite a nice working system in the beginning of use and the best efforts at design time, systems need to be improved at use time to fit new needs and the possibility of incorporating new technologies. (Fischer, Ostwald, 2002) Other articles describe the adaptation of PD in a non-traditional context like network organisations and highly flexible organisations like municipalities. (Dittrich, et al., 2002, Törpel, et al., 2002) How does the development practice of Idavall fit into this picture?

The first version of FRI was released in 1987, it was developed to fit a certain customer. Since the first version of FRI, FRI has been used by several customers and has been continuously developed on the basis of user feedback as I described earlier in this thesis. The users' proposals have had a strong impact on the development from the very beginning.

Even if users of FRI affect the development through FRI-meetings, support and courses, they never actually participate in the design process in a narrow sense. What is going to be implemented is decided by Idavall. You can claim this is not PD because the users are not present in the design process in the way that traditional PD suggests. I will claim, even so, that this is PD, but not the strictly traditional sense. The users drive the development of FRI as Gustaf expressed in a demonstration of FRI. User feedback is the key factor in the development process of FRI, because only the feedback of the users can ensure that FRI lives up to the users' needs and expectations. This is also recognised in PD literature. Developing software systems without listening to the users may be possible, but it is unlikely that such system will live up to the users' requirements. Only the feedback of the users can ensure that users get what they want. User feedback is also needed during the whole life cycle of a system. It makes sure that a system is continually improved and adapted to changing work practices. (Kensing, Blomberg, 1998)

Users of FRI come from several communities like municipalities, associations, museums, hospitals and so on. Each customer has often only few users, like one or two. FRI is used almost in the same way but is still individually adapted to fit each community. Users from different communities can be regarded as members of a community of interest (CoIs) as Fischer, (2001) calls it. (See 'PD in other contexts'). Basic challenges facing CoIs are found in building a shared understanding of the task at hand. FRI-meetings and courses are typically activities that bring users together and help to develop such a CoI. These meetings became an arena where it is possible to discuss common questions and problems. Design issues are highlighted as well.

Problems change rapidly due to new tasks or technologies being introduced into the user community. New needs arise during the use of FRI and are mainly reported individually to Idavall through the support service. The support service is the most important and most used forum where the users report their individual proposals. Idavall takes the users' proposals into account and implements, for example, new functionality in the up coming versions.

Since the users are located at different small work places all over Scandinavia it is not possible to organise future workshops, work with prototypes, scenarios or mock-ups in a traditional way as PD suggests. With a lot of users spread all over a big area results of cooperative use of mock-ups and prototypes would be outdated before they have even been produced. (Törpel, Wulf, Kahler, 2002) Neither is Idavall able to do field studies to gather work practice data which could be a base for further development. Field studies might even be unnecessary since Gustaf by his long term contacts to the users and their workplaces is aware of

their needs. Idavall's way of developing software can be seen as a variant of PD fitting the specific situation. Neither Idavall nor the users are aware of the research area of PD as far as I know. The development that takes place in cooperation to the users is grounded on common sense and takes place independently of what researchers think of or call 'cooperation'.

You could say that every version of FRI is like a prototype, which is the base for forthcoming versions. Versions are discussed at for example FRI-meetings, and are modified to meet users' requirements. The users themselves create scenarios in, for example, FRI-meetings when they put forward problems in the use of FRI. Gustaf or other users come up with solutions or proposals for new functionality to solve the problem. One example was when a user described a problem he has when he would like to fill in a number of a bank accounts in a form. It turned out that the text field that was reserved for this task was too short. After a discussion about this task and some explanations about how this text field was meant to work, Gustaf promised that this text field would be extended in forthcoming version.

Törpel, et al. (2002) describes a similar case of a networked organisation where traditional PD methods not are applicable. They recommend instead a continuous process of parallel experimentation and network-wide collection of experience, feedback, and integration into an overarching infrastructure consisting of a variety of local substructures when participants are not able to meet for bigger design sessions or when the infrastructure does not allow that. They think that an ongoing process of contributions and discussion in smaller groups would have a fruitful impact on the on-going development of a system. FRI-meetings are an attempt and a possibility to users in a region to meet each other and a representative from Idavall to discuss problems, news and requirements for the evolutionary development of FRI. FRI-meetings are as well a meeting-place where users with different backgrounds regarding, experience and knowledge about FRI are able to participate. Those who are not very experienced regarding FRI get an opportunity to learn more about FRI and how the development of FRI works. On the other hand they may have experience from other kinds of work environments that can be valuable in the evolutionary development. One important approach that is emphasised in informed participation is mutual learning where the sharing of the unique knowledge that each participant brings to the design helps to highlight and solve design problems. (Fischer, Ostwald, 2002) Even Törpel, et al. (2002) regard the mutual learning as an important issue in an on-going process.

The participation of users in the development of FRI is not driven by requirements from any unions or users. Interested users who are encouraged by Idavall to participate and affect the functionality of FRI drive the development. An increasing functionality makes it easier to use FRI in the daily work.

I have recognised that Idavall is eager to create a close relationship to their users and between users. Idavall uses this relationship in the development of FRI. A mutual close relation creates a developing environment for further development of FRI. The way Idavall treats their users and strengthen the possibilities for users to participate in the development process reminds of the concept of Customer Relationship Management (CRM), which focuses on the relationship between supplier and customer. CRM in combination to PD might be a good way to further evolve user participation in development processes.

Using Customer Relationship Management to Enhance PD

In this passage I will describe in what way Gustaf and the other employees at Idavall communicate and cooperate to their customers and users. This communication is an important and valuable resource for users and customers in allowing them to impact on the development of FRI. As I have suggested in my thesis it is important to have a continuous and ongoing contact to the users to be able to develop a system that satisfies the requirements of the work activity and users' needs. It is important to create meeting places and environments where users and developers are able to meet each other and discuss.

Customer Relationship Management (CRM) is a concept used in the marketing area. The aim of CRM is to create a strong and lasting relation between supplier and customer, in this case even users. The objective is to build a strong relation that is based on mutual confidence. A similar relationship is the base for the possibility to jointly develop FRI and make it possible for users to participate in the development process. I will in the following relate CRM to the way Idavall is working and give examples that strengthen this statement.

Before the industrial revolution, craftsmen and apprentices worked in towns or villages. People who worked in the craftsmen's workshops became proficiency and forwarded their skills to the next generation, which guaranteed to keep the tradition. The craftsmen often controlled the whole process of production and had significant knowledge of their customers. They often knew their customers in a personal way and knew how they used the product and which their needs were. A close relationship, grounded in a mutual trust was a basic condition. (Storbacka, Lehtinen, 2000) To create a close customer relationship is still an important issue if one wishes to become a successful company. At Idavall, it works in a similar way today as it worked as described above. The objective for Idavall, as I mentioned earlier, is to know their user in a personal way, in what way they use FRI and their needs. Idavall puts a lot of effort into this challenge.

One way to get to know their users is to meet and talk to users in FRI-meetings, support calls and courses. Probably, the most important way to keep in touch with the users is the support where the users and employees at Idavall have a great opportunity to create and maintain their relationship. The persons who give support are the same who develop FRI, which means that they possess and can develop a deep knowledge about the application and their users. Storbacka and Lethinien argue that systems for support are a method to keep in touch with customers and further develop the relationship. Support is therefore essential to offer new customers especially. How the relationship will grow depends on the introductory relationship. Customers who feel that the supplier does not bother about their relationship tend to end the relationship after a short period of time.

A central notion within Customer Relationship Management (CRM) is the creation of customer value. The objective within CRM is not to maximize the profit of individual transactions, it is rather to create a permanent relationship to the customer. Another cornerstone in CRM is affiliated to the responsibility of the company. It is not enough to only satisfy the needs of the customer. The company is only able to create a strong relationship if they take the responsibility to develop these relations and offer their customer to create new values for themselves. (Storbacka, Lehtinen, 2000)

One way to create new values and permanent relationships with customers is, among other things, to allow for participation in the development process. When the users understand and notice their significance in the development process they get a feeling of importance. Another way can be to create networks between users through FRI-meetings and

courses. Those networks stimulate the users to create communities of interest and/or communities of practices where the members can teach each other and learn from different experiences. (Fischer, Ostwald, 2002, Lave, Wenger, 1997)

To be successful, the company has to gain the feelings of the customer/user. The feelings are of great importance, especially in the beginning of a customer relationship. Feelings come up as a result of interplay between people. Companies are able to encourage the origin and strengthen the feelings through conduct a dialogue to their customers. A good way to engender feelings is to discuss face to face. Feelings come up as a result of individual processes. People think that a face to face discussion is more reliable than written information. Face to face information is more effective as well. The more a company will give attention to the individual hallmark in the dialogue the more effective the intensity of the feelings will become. When the competition is tough is it of great importance that mutual confidences exist between the company and the customer. (Storbacka, Lehtinen, 2000)

The demonstration of FRI to potential customers is one of the first contacts in personal that occur between Idavall and the customer. It is mainly Gustaf who is in charge of these demonstrations. Before the demonstration Gustaf almost always has been in touch to the customer by phone and discussed the domain where FRI is going to be used. He has also discussed the requirements and expectations of the customer. When the demonstration takes place he is able to adapt it to the customers who are present. This adaptation gives a feeling of that Idavall is eager to satisfy the customer by demonstrating those functionalities that are of interest. The customer gets a feeling of importance, which is of net benefit both to Idavall and the customer.

To use CRM is a way to generate positive attitudes and relationship between customer and supplier. These positive attitudes and relationships are supposed to lead to pleased and satisfied customers that will become loyal to their supplier. Suppliers have to spend more resources to develop than to create these relationships. Long relationships are more valuable than short relationships are. CRM lacks concrete methods for attaining these values, which can be a deficiency. (Storbacka, Lehtinen, 2000)

The register of customers at Idavall is a kind of CRM-system where stored records are based on personal communication with users and customers. This register stores information about the customer, for example who is using FRI, which version is used, which parts are installed and so on. Computerised systems that manage customer relationship are available in the market. Mainly the marketing and sales departments use those systems. They are not as successful as expected. Today, companies think that CRM is more than marketing and sales and are looking for systems that integrate business systems and today's CRM systems.

CRM can act as a complement to PD in that way that CRM creates an environment where users, customers and the company are able to meet and work together in a relaxed and trustful way. If participants know and trust each other it is easier to participate in different PD-related methods.

I will claim that what I have discussed in this part can be thought of PD. What Idavall does can be useful for other companies. PD in such situations can gain inspirations from the lessons of CRM. But; the promises to users and customers have to be fulfilled. The commitments have to be kept. Traditional software engineering might not be so fitting because of long lead times, fixed requirements and long development cycles. How does Idavall manage to do up to 20 releases each year? How do they further develop their software?

The Development Process at Idavall

In this part I will describe the development process and the architectural structure of FRI. What characterizes the development process is in the highly flexible and informal way it is performed. The process has its focus in coding and the importance of user influences. The process is invisible to an outsider and does not follow any particular rules. The process does not deal with any particular documentation. The only documentation is the program. It is therefore difficult to study the process.

The fact that the development process focuses on coding, user involvement and does not make an effort to do a lot of documentation reminds me of Extreme Programming (XP). XP focuses equally on customer participation and involvement. In the view of this statement I will briefly describe XP and make some parallels to the development process of FRI.

Software Development at Idavall

To be employed at Idavall you have to be a social person who is able to communicate with the customers and users in their language. You have to be able to talk to a technician in a technical language and to the common daily user in a plain language. It is through this communication the developers at Idavall get to know what the shortcomings of FRI are and what the users needs consists of. It is thus not enough to be a good programmer. It can take a year or more to find the right person for a vacancy. New employees are taught by the old ones in a kind of apprenticeship, (see Lave, Wenger 1997), where Torbjörn and Gustaf act like masters. It takes about one year before you independently can give support on FRI in a proper way. Janne expressed it: 'There is like a well-used path between my room and Torbjörns' this path has been created during many times of walks when Janne was about to learn FRI. The fact that everyone is located in the same area and some of the staff has worked at Idavall for a long time means that a lot of knowledge is located 'in the walls'. The heart of Idavall is the coffee room, which is located in the middle of the area. The walls of the coffee room are decorated with postcards and Christmas cards from users and a map of the Nordic countries where all customers are marked with a needle. Previous Christmas cards with a photograph of the employees at Idavall as Santa Clauses decorate the walls as well.

Administration and ranking of proposals

Idavall has a valuable ongoing dialogue with users and customers through the support process, FRI-meetings, courses and similar activities. As I have mentioned earlier it is through these activities that users present their proposals and points of view. Everyone at Idavall takes part in the support of FRI and everyone get to hear proposals from the users. Everyone is aware of problems as well.

Every FRI-meeting has its own web site where, for among other things, all proposals from this meeting are noted. Proposals, which are expressed during support calls, are handled different depending on who is receiving them. Some are noted on a piece of paper, others are noted in a kind of register. It varies sometimes as well from one day to another and from person to person.

Before they start to implement new proposals all proposals are reviewed and ranked. Proposals are ranked by their quality like for example, if it is generic, if it is complicated to implement, if it affects other functionalities in a bad way, if it is useful for many users, if they have time to implement it, if they think it is fun to implement it and so on. Torbjörn says

that he prefers to implement many smaller improvements than one big one because many smaller changes make a lot of people happy.

Sometimes it can take a long time before a proposal is implemented at all. Sometimes it is implemented directly. Those proposals that are not implemented are saved anyhow in a document because a new technique might appear later, which will make it possible to implement the proposal later. Proposals which are highly specific for one customer are not often implemented, save only if they are very easy to implement and do not harm anything else. Most of the proposals are discussed at Idavall before they are implemented. Simple proposals as for example a new formulation of a text do not have to be discussed before implementation. Torbjörn, who is the chief developer, implements all that affect the original FRI, Janne is responsible for the implementation of changes to the web version of FRI.

Development cycles

I have noticed that the development process can be thought of as two different cycles. The smaller or faster cycle where bugs are fixed and minor improvements continually take place is highly flexible. In the bigger and slower cycle major improvements take place. These cycles run in parallel during the year. The implementation in the smaller cycles is more intensive during March – May and October – December when most of the programming takes place. During these periods the customers do their planning and become aware of the need for new functionalities.

Focus in the small cycle of development is on implementation of the users' proposals and bug fixes. In the bigger cycle focus is in the moment on the development of a new 32-bit version of FRI where the database will be changed as well as the version of Visual Basic.

The ongoing development of FRI is very improvised. Planned formal meetings where the development of FRI is discussed are very rare. Informal and spontaneous meetings are on the other hand very common. As everyone is located in the same place and the staff is not that big is it easy to meet and talk when a problem or question arises. Coffee- and lunch breaks where everyone meets serve as important opportunities for discussions as well as spontaneous meetings in front of the computers. Decisions are made very quickly. This informality and swiftness leads to a highly flexible development process.

If bigger changes are about to be developed, formal meetings are held where the development is discussed and sketches of the system are presented. Nowadays it is mostly refinements of existing functionalities, improvements of existing parts and the fixing of bugs (which has a high priority) that takes place in what I will call a small development cycle. When FRI for Windows was developed in the beginning of 1990's a lot of new development was carried out. In the bigger development cycle a new 32-bit version of FRI was comme3nced in 1999 and is still taking place, in which Visual Basic 6.0 and SQL will be used. These two development cycles run in parallel.

The smaller development cycle

The programming takes place mainly in the afternoons when the support is closed. Proposals that inspire the development are reviewed and discussed before decisions are made about what to implement. Torbjörn decides in the end what is going to be implemented to FRI and the database and in what way. Janne decides about issues that concern the user interface on the Internet and in what way the web-versions communicate to FRI. It usually takes one to five weeks before a new version is released on the web site. The development of FRI is an

ongoing process. FRI is continually improved to fit users' needs. You do not have to download and install every new version, FRI works anyway. Sometimes an upgrading is recommended and is in this case announced on the web site and e-mailed to every customer.

The bigger development cycle

As mentioned before the last major development took place in 1996 when the admission system part was developed. Today a new 32-bit version of FRI is under development by Torbjörn. The development of this version is mainly taking place when the smaller development cycles are relatively quiet. The new version of FRI is developed in Visual Basic 6.0, which is an object oriented programming language. Visual basic 6.0 offers more opportunities to develop new features in FRI. Automatically sending an e-mail to confirm a booking is one example. Today you have to copy the confirmation and paste it into an external e-mail programme before it can be sent. With the new version, the customer will also have the possibility of choosing another database, for example SQL-server, Access or Oracle. This is an advantage because most of the customers already have a database and want to avoid administrating more databases than necessary. One big change is that the subsidy-part, which is customer unique in the present FRI, will be generic in the new version as the other parts of FRI are today.

To be able to reuse parts of the present FRI in the new 32-bit version some duplication of work is necessary today, just to make a smoother change over. Torbjörn builds a kind of SQL-translator that translates SQL-questions to the language of FHS. In the new version where SQL-questions really are needed, these SQL-questions are already developed, so only smaller modifications are needed to suit the new version.

Torbjörn also has to create, for example, interface items that fit into the new version. Those pre-defined items in VB 6.0 are not enough and do not always have the proper characteristic. Some items have to be transparent, and have a special layout, to fit into the graphic user interface.

It is not clear when the new version will be released. It depends on how long it will take to get a reliable version. Idavall does not want to make promises if they are not sure. A small beta-version is already available to some customers who are interested to test. The old version FRI will run in parallel to the new version for an indefinite time. For example, to partially-sighted users some parts of the old version is better since it is possible to get bigger text in the old version than in the new one.

System Architecture

FRI is built of different parts. As I mentioned earlier the part called the 'register of customers' is the base of FRI and some systems that are needed for administration and maintenance of the system. The picture (Fig. 4) below shows a simplified view of different parts of FRI and the way they are connected to each other. The arrows show the flow of data between different parts of FRI. Some arrows should be double directed, for example between 'Schema' and 'Debitering' where 'Debitering' gets information from 'Schema'. The system is very complex and it is difficult to sketch a clearer picture of the system that describes all communication.
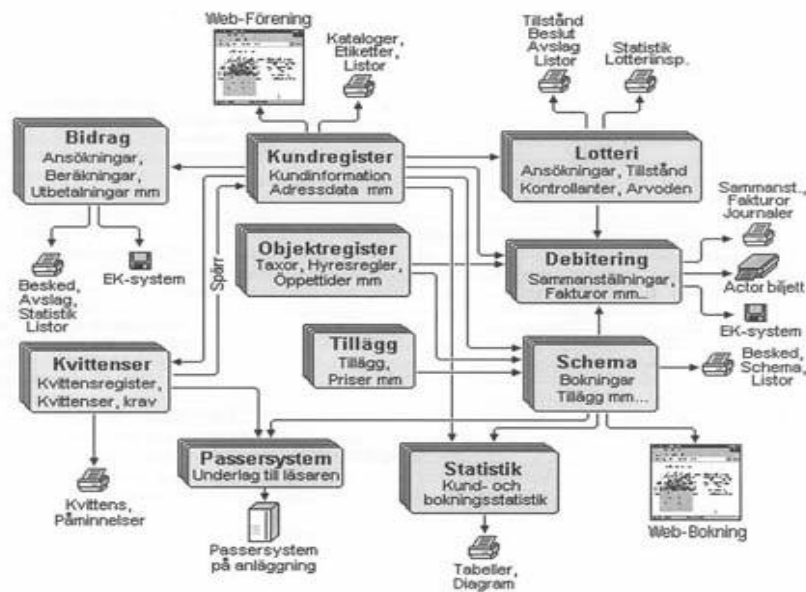
Fig. 4 Structure of FRI

The graphical user interface is built in Visual Basic 3.0, for example, see fig. 1, 2, and 3. Each window is built up by of a number of program files. Each module has a main window where several sub windows are connected. Some program files are generic and are used in several windows, for example the search procedure, and some are specific and are only used in one window. The customer window consists of 22 specific program files and 55 generic program files. In total, about 80 % are specific program files and 20 % are generic. Some program files only consist of functions and others define the graphical user interface.

The database that is used is called File Management System (FHS), which is a small Swedish database. Each module of FRI consists of a number of data files in the database. (Fig. 5)

Filnamn	Innehåll
FRI00.FHS	Kundregister
ADRESS.FHS	Kundens adresser
ANL.FHS	Kundens anläggningar
MEDL.FHS	Medlemsinformation till kundregister
VERK.FHS	Verksamheter

Fig.5 data files constituting the register of customers

A data file in FHS is comparable to a table in a relational database like SQL-server or MS Access. Each data file saves data records with a number of predefined fields. (Fig. 6 and 7) It is easy to change properties in a special field if it is needed; you just have to change values in the affected field. The values have to correspond to the values in the program files. If you want to change the length of the field of customer name you have change value of this field in the program file that built the user interface as well, otherwise it will not work.

Namn	Typ	Längd	Förklaring
KOD	A	10	Kortnamn
MODERF	A	10	Moderförening
SAMDEB	B	2	Samdebiteras med moderförening
KNP	A	4	Kundnummerprefix
KN	B	4	Kundnummer
FTYP	A	20	Kundtyp
KGRUPP	A	20	Kundgrupp
NAMN	A	30	Kundnamn
CO	A	25	Co adress
ADR	A	25	Adress
PNR	A	10	Postnummer
ORT	A	20	Ort
TB	A	20	Tel bostad
TA	A	20	Tel arbete
TELFAX	A	20	Telefax
EPOST	T	4	E-post
HEMSIDA	T	4	Länk till hemsideadress
WEBPUBL	B	2	Godkänt Web-publicering ja/nej
PENR	A	14	Person/org-nr
VERK	T	4	Verksamheter i textformat
HVERK	B	2	Huvudsaklig verksamhet
RIKSORG	A	30	Riksorganisation
MT	B	4	Medlemsantal
BIDRMED	B	4	Antal bidragsberättigade medlemmar
MEDLÄNDR	B	4	Datum för ändring av medlemsantal
ANT	T	4	Anteckningar
Å	A	4	Bildad år
BIDR	B	2	Bidragsberättigad
BDAT1	B	4	-"- från dat
BDAT2	B	4	-"- till dat
KONTOTYP	B	2	0=PG,1=BG,2=Bank
PG	A	12	Postgiro
BG	A	12	Bankgiro
KONTO	A	20	Bankkontonummer
VILANDE	B	2	1=Vilande kund
KVITT	B	2	1 om ej bokningsbar pga kvittens
FAKT	B	2	1 om ej bokningsbar pga faktura
ANNAN	B	2	1 om ej bokningsbar pga annan orsak
OMR1	T	4	Område 1
OMR2	T	4	
OMR3	T	4	
OMR4	T	4	
OMR5	T	4	
OMF1	B	2	Fördelning område 1
OMF2	B	2	
OMF3	B	2	
OMF4	B	2	
OMF5	B	2	
TAXKLASS	A	1	Kundens taxeklass
RABATT	B	8	Rabatt i %
X1	T	4	Eget sökfält
X2	T	4	
X3	T	4	
X4	T	4	
X5	T	4	
X6	T	4	
X7	T	4	
X8	T	4	
X9	T	4	
X10	T	4	
MOTKONTO	T	4	Motkonto vid interndebitering

Fig. 6 A data file in the File Management System with different fields and its properties

Databasfält	Typ	Längd	Max värde	Sorteras	Anmärkning
Binärt tal	B	1	256	Enligt siffervärde	Endast tal kan anges
Binärt tal	B	2	−32,768–32,767	Enligt siffervärde	Endast tal kan anges
Binärt tal	B	4	−2,1–2,1 milj.	Enligt siffervärde	Endast tal kan anges
Binärt tal	B	8	15 + 4 dec.	Enligt siffervärde	Endast tal kan anges
Text	T	4	Fri textlängd upp till ca 64.000 tecken	Kan inte sorteras	Tar endast upp så mycket plats som använts
Alfanumerisk sträng	A		Maxlängd sätts i FRI	Enligt alfabetet	Bokstäver, siffror, övrigt
Numerisk sträng	N		Maxlängd sätts i FRI	Enligt siffervärde	

Fig. 7 Explanation of type and length in the fields

FRI is using a kind of 'help' program, which make it possible to communicate between FRI and FHS. One of these 'help' programs handles call of functions to FHS, it opens and closes files and navigates between different fields in the files of FRI. Another helping program creates SQL-questions, and translates them to suit the language of FHS. SQL-questions are used even if it is not necessary because it will later make an easier and smoother conversion to the new 32-bit version of FRI. A simplified cooperation between files inside FHS can for example look like shown below on the picture. (Fig. 8) The arrows show in which way the data is transferred. A lot of different files are involved in getting a proper answer to a question in FRI.
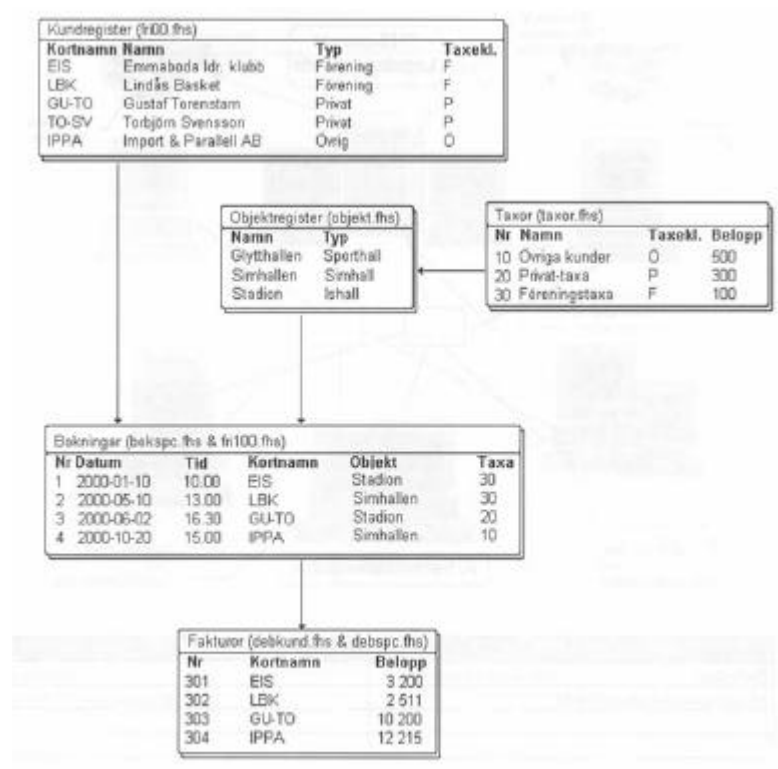
Fig.8 Cooperation inside FHS

Almost every user uses FRI in different ways, although it is broadly the same FRI everyone uses, albeit there could be different versions. This raises the requirements of FRI. To be able to fulfil most of the users' needs FRI has to have a lot of possible settings and thus be flexible. A lot of choices, settings and demands of flexibility lead to a lot of code.

FRI consists in total of approximately 56 000 rows of code plus 16 000 rows of code that build the web versions. It should be difficult to navigate and maintain the code but Torbjörn does not think it is difficult. He says he has a feeling for where to find a certain code. If he wants to improve the function of searching he naturally presses the button of 'Search' in the VB 3.0 development tool. In VB 3.0 one can use the user interface to navigate in the source code. Some code appears and all program files, functions and types that are connected to this button can be browsed from this view. From this view he navigates to the program files and the functions he wants to make a change to. It is important to name functions and program files, for example, in a logical way otherwise it could be hard to find the desirable function or program file. If a beginner would make changes it would not be easy to find the right code. Torbjörn says 'It would probably be a lot of searching in the beginning'. Documentation is rare. Some documentation is done in the code when Torbjörn or Janne thinks it could be a problem to understand later. The code itself is the documentation of FRI. Gustaf writes user manuals, which can serve as a kind of documentation as well.

Janne stores all files belonging to a running version on the server. He just gets the actual files which he codes to his local workstation. When he has made changes or/and improvements he tests all code before he saves the changed files to the server. This means that he always has a running version on the server.

These developing processes I have described are pretty informal but they work. One probable reason for success can be the small number of people working at Idavall. Another reason can be the short distance between decision and performance, which decreases the need of formality. This way of developing software is not very common in the traditional software engineering community. Discussions of developing software in an alternative way are present in the community of software engineering. One topic is the practice of Extreme Programming as an alternative to the traditional developing processes.

What is Extreme Programming?

I will in this part give a brief description of Extreme Programming (XP) and relate it to my field studies. My description of XP is based primarily on (Beck, 1999, Karlström, 2002) and http://www.extremeprogramming.org (020815)

XP is a lightweight software development methodology that aims to make software development more flexible and focus on highly flexible environments with quickly changing requirements. XP implements a highly user-centred approach where users play a central role during the design process. Users work together with system developers in a strongly iterative, prototype-based process. XP is based of a few fundamental *values, principles* and *activities*, which are implemented by 12 *practices*. Most of these cornerstones are not new, but the combination and packages of them are new. An XP project works best if each member of a team is assigned to a role so that everyone has different responsibilities regarding to the different practices. One role can be assumed by several people and conversely one person can assume several roles if needed. Roles and practices will be described later. The fundamental values are: *communication, simplicity, feedback* and *courage.*

From the four values a number of principles are derived to guide the style of XP. The five fundamental principles are *rapid feedback, assume simplicity, incremental change, embracing change* and *quality work.* Each principle embodies the values. A value may be vague while a principle is more concrete. I will describe some of the principles below.

To completely develop a program that is working properly you have to do some coding, but it is not enough. Something else is needed. The four basic activities in XP are *coding, testing, listening* and *designing.* XP stresses customer satisfaction and is designed to deliver the software your customers need when it is needed. Users/customers play a key role during the design process, specifying and designing the system in cooperation with system developers in a strongly iterative, prototyped-based process. XP empowers the developers to confidently respond to changing customer requirements, even late in the life cycle. Teamwork where managers, customers and developers are all part of a team enhances the chances to delivering software of high quality and in time.

I will claim that the development processes at Idavall have a lot of common with XP. Flexibility, adaptation to quickly changing requirements, stress on user satisfaction, short iterations and small releases are, among others, key words that fit into the concept of XP.

I will describe these main values, principles, practices and roles of the team that I found relevant in connection of Idavall and the development of FRI. I will encourage the interested reader to read more about XP in the literature.

Main values

C*ommunication* is emphasised within the development team and also between the team and the user. Communication within the team is achieved through frequent planning and design

meetings where the knowledge of the system and an awareness of the development process are shared. A stand-up meeting is held every morning and is used to communicate problems, solutions and promote team focus. Everyone stands up in a circle to avoid long discussions. A lot of smaller meetings can take place in front of a computer during the day where code can be discussed and ideas actually be tried out. This kind of meeting-culture makes it more efficient and saves time and money in the end.

Communication with the users is represented by the role of a customer in the team. The customer is the person who has to decide what is most important to implement and takes an active and major part in planning development. It is important that the customer has the authorization to make decisions about the software and has the knowledge of where and how the software will be used in the future.

Simplicity, XP thinks that it is better to do a simple thing at first and make subsequent changes when needed, rather than do complex work at the outset that may in fact never be used. Encourage simplicity means that the resulting software is easy to understand and reduces the time spent on extensions that may never be needed.

A major feature in XP is the continually *feedback* and evolution of the software by users. Having a customer representative as part of the development and planning process results in a dynamic environment. Testing starts at day one of the project, which means that the developers get feedback as well from the first day. Concrete feedbacks works together with communication and simplicity. The more feedback you have, the easier is it to communicate.

Principles

Rapid feedback is possible when developers, customers/users and managements work closely together. The customer who is present in the team notices and discovers very soon if something does not work as it should. He feeds back his points of view immediately instead of the next week, month or year. When the programmer gets the feedback he interprets it and put what was learned back into the system as quickly as possible. The programmers learn quickly how best to design, implement and test the system.

Assume simplicity means that every problem would be treated as if it can be solved with most possible simplicity. K Beck claims that assuming simplicity is the hardest principle for programmers to accept since they are traditionally told to design for reuse and plan for the future. XP encourage programmers to solve the problems of today and trust their ability to add complexity in the future if it is needed.

Incremental change is to make small changes often instead of make big changes that are made all at once. A problem would be changed with a series of small changes. Incremental changes are made in different ways in XP. The design changes a little at a time. The plan changes a little at a time and so on. Even the adoption of XP must be taken in little steps.

Practices

The practices can be introduced interdependently or as a whole depending on the situation in the development organisation. The practices are intended as a starting point for a team.

The planning game: At the heart of the XP planning process are user stories. A user story is a two or three sentence informal description of things that the system needs to do for the customer, is written by the customer. New stories are added continuously during the whole project. This means that new functionality is continuously added to the project. It also

means that the team members do not have a clear picture of the product at the start of the project. When the stories are written they are prioritised together with the development manager on the grounds that he has the best overview of the technical status of the product. These short stories form the basis for planning development.

The development of a project is broken up into a lot of small releases, which are divided into iterations. Iterative development adds flexibility to the development process. A release plan is established, which consists of the most important user stories. The less important user stories are implemented in the next release or later, sometimes a user story never will be implemented. The customer decides what is most important. An XP release cycle takes two to four months and each iteration takes about one to four weeks. This collaborative planning process, with roles for both customer and developer and rules to follow is called the *planning game.*

At the beginning of each iteration, the customer chooses a smaller subset of those stories that could be achieved in the time-scale of the iteration. These form the iteration plan. You would not schedule the programming tasks in advance and not implement something that is not scheduled. Just-in time planning is an easy way to in a flexible way change user requirements. If it turns out that all tasks will not be finished in time, you have to make a new iteration planning, re-estimate and remove some of the tasks. The high-level feature description in the user stories are broken down to specific engineering tasks. This is the point where most of the system design takes place. Iterations are kept small so that the customer has opportunities to evaluate and provide feedback to the programmers. When the completion of each iteration, the system is presented to the customer for evaluation and feedback.

Small Releases are needed. It is critical to getting valuable feedback in time to have an impact on the system's development. Every release should be as small as possible, containing the most valuable business requirements. It is the task of planning game to discover small units of functionality that can be released into the customer's environment early in the project and make him happy.

System Metaphor; the metaphor helps everyone on the project understand the basic elements and their relationships. One way is to name classes and methods consistently. What you name your objects is very important for understanding the overall design of the system and code reuse as well. By asking for a metaphor we are likely to get an architecture that is easy to communicate and elaborate.

Simple Design; a simple design always takes less time to finish than a complex one. Every piece of design in the system must be able to justify its existence. Always do the simplest thing that could possibly work It is faster and cheaper to replace complex code immediately before a lot of time is wasted on it. Do not add functionality before it is scheduled it might not be scheduled. Keeping a design simple is a hard work because it is sometimes tempting to add another feature that you think is valuable and you know exactly how to implement it.

One of the biggest criticisms in XP is the almost total lack of documentation. Documentation is written when it is necessary and when the customer asks for it. Documentation is not written because you just have to do it. Management do not need any special kind of documentation of the process since the running system shows how the process is proceeding. *Testing;* Traditionally, testing is a phase of development that is carried out after the main coding is finished. Test cases are designed to cover as much as possible of the logical functionality of the code. Testing in XP fills the same role as in other software engineering

processes. The difference is that in XP, programmers are requested to write the tests before the code. Every time new code is written, a corresponding test case must be written and implemented first. In this way the person who has written the code writes the tests in an iterative way in parallel with the code. This is an advantage since the tests are developed while the context is still fresh in the programmers mind, and there is constant feedback on the state of the code as tests can be run at any stage of development. All code has to be tested before it is implemented into the program.

Refactoring: XP does not think it is cost effective to use and reuse code that is no longer maintainable even if it still works in some way. Refactoring throughout the project entire project life cycle saves time and increase quality. Refactor continually to keep the design as simple as you possible can to avoid needless mess and complexity. Keep the code clean and concise so it becomes easier to understand, modify and extend.

Pair Programming: I believe that pair programming is the most associated and well-known concept when talking about XP. Two programmers at a single computer write all production code, i.e. code that is actually used in the product. The programmers sit side by side in front of the monitor. The one with the keyboard and mouse types and thinks tactically how to best implement this present method. The other one thinks more strategically about how the method fits into the class. In this way the code undergoes peer review as it is written. Pair programming increases software quality without impacting time to deliver. It may strike one that pair programming is a waste of time, but the time is compensated for by the improved code quality at the time for writing.

Pair programming is an excellent way to introduce new programmers to the code and project. Another benefit is that, as pairing is dynamic, pairs are swapped around, team members acquire knowledge of different parts of the system.

Collective Ownership encourages everyone to contribute new ideas to all parts of the project. Any programmer can change any line of code to add functionality, fix bugs or refactor. In XP everybody takes responsibility for the whole of the system. If programmers think of defects as a group issue, rather than someone else's 'private' defect a lot of irritation can be reduced and create a constructive atmosphere. Collective ownership is more reliable if a programmer leaves the project it is less vulnerable.

Continuous Integration: Code should be integrated and released into the code repository every few hours, whenever it is possible. Continuous integration often avoid diverging or fragmented development efforts, where developers are not communicating with each other about what can be reused, or what could be shared. Everyone needs to work with the latest version.

Each development pair is responsible for integrating their own code. When a planned functionality is finished they have to integrate. Almost continuous integration avoids or detects compatibility problems early. If you continuously integrate you always have a working system that you can show to your customer, you do not have to do all integrating the last weeks before deadline.

40-Hour week: Overtime is a symptom of a serious problem on the project. Working overtime sucks the motivation out of a team. If the project requires overtime you have to re-plan and maybe change the scope of the project or timing.

*On-Site Customer: O*ne of the few requirements of XP is to have a'real customer available'. A 'real customer' is someone who will really use the system when it is in production. The customer does not only help the development team- he is a part of it as well.

All phases of an XP project require communication with the customer, preferably face to face, on site. Initially the customer writes user stories that are used in planning. Customers select the user stories, which they wish to have implemented for the next system release, assign priority, negotiate and estimate time. Everything is done in collaboration with the developers. The customer must make decisions that affect their business goals.

After the specified stories have been implemented, customers take part in the acceptance testing to make sure that the requirements are fulfilled. The customer writes the acceptance test in collaboration with the tester. The program is tested at the end of each iteration. If the customer is not satisfied the story goes back into the pile of stories to be implemented in the next release. Thanks to the prospect that the customer can continuously try the system the customers are able to give the developers feedback in real time.

Coding Standards: Code must be formatted to agreed coding standards. Coding standards keep the code consistent and are easy for the entire team to read and refactor. The standard should emphasize communication and must be adopted voluntary by the whole team. The name of variables and methods should be transparent to make it easy to understand their functionality.

Roles of the team

The *programmer* is the heart of XP. An XP-programmer has to have the faculty to communicate, coordinate and negotiate close to other team members in order to be successful.

The *customer* is another important part of the team. The programmer knows who to program and the customer knows what to program. As a customer you have to learn among other things how to write good user stories and functional tests to ensure that the product does what it is supposed to do. Another important task is to communicate with other team members and give feedback to programmers and testers. A customer must be able to make decisions and become comfortable influencing a project without to control it.

The *coach's* role is to be responsible for the process as a whole. The coach notices when people are deviating from the team's process and bring this to the team's attention. The coach always remains calm when a problem arises. The coach has to know the ideas behind XP and how to use them in the current project.

Comparisons between XP and the Development Process of FRI

As I mentioned in the introduction to this part of the thesis I have noticed that the development process of FRI has a lot in common to XP. I will further compare some of the ideas behind XP to the development of FRI.

One of the most important and striking strengths of FRI is that the users drive the development. The users are not on site physically but through different kinds of meeting-places, as I have described earlier. The users convey their proposals and have in that way a significant impact on the system. It is true that the developers at Idavall rank and priorities the proposals but it still the user-proposals that are implemented. A risk is that the developers rank proposals in a wrong way from the users' point of view. The prioritizing depends on different circumstances as I have described. This is a difference compared to XP where the customer ranks the user stories and decides which user stories that are most important to start with and which ones that can wait until later in the process.

Another important and striking strength is in the way the communication takes place between the developers. Informal meetings often take place during the day. It is like a kind of

stand-up meeting. People walk in and out of their rooms and discuss how to solve problems or decide what to implement. Even coffee and lunch breaks provide important possibilities for discussion. Since everybody is continuously updated there in no need for formal meetings. The user communication is not as frequent as recommended in XP, but still they have daily discussions with users through the support process. Users give rapid feedback when a new version is released, propose improvements and report bugs. It would be hard to get an on-site customer all the time and should not be desirable either since the customers are so different. Since the development process is very informal and all developers are located in the same place they have the ability to very quickly adapt to changed requirements and be highly flexible. Short iterations makes small releases, about 20 new working versions are released every year. Each release often consists of some new functionality and small changes or bugs that are fixed. The code increases continuously since the functionality of FRI growth. Therefore, Torbjörn and Janne have to refactor their code regularly. Refactoring makes it easier to maintain the code and add new functionality. It is easier to find a certain piece of code as well. Refactoring saves time and money in the end as Torbjörn expressed.

Janne tests all new code locally at his computer before he integrate it into the running version that is on the server, that means that the version on the server always is the latest running version, this way of testing before integrating reminds of the tests and continuous integration in XP.

Documentation has a low prioritizing in the development of FRI as in XP. The code is the documentation of the program. Sometimes some comments are included into the code if Janne or Torbjörn think it would be difficult to understand the code later. Gustaf writes manuals, which could be thought of as a kind of documentation for the users. The lack of documentation might be a problem, especially if someone quit his job or get sick for a long time. Today, everything is in the head of the developers. The 'telelogg' is a kind of documentation that is used in the development of FRI where a lot of information is stored about users and their problems.

Pair programming that is one of the cornerstones in XP does not exist at Idavall since just two of the developers write code. I do not think it would make any sense to encourage pair programming at the moment because they are coding totally different parts of FRI and in different programming languages. Maybe it could be an alternative if, for example, one more web-developer was hired.

XP advocates, as I mentioned earlier, collective ownership of code and a coding standard to make it possible for everyone to change and understand the code. Because the programmers code in different programming languages it is difficult or even impossible to have collective ownership of the code. As it is today the code belongs to its programmer, and nobody else is able to make changes to the code. It is the same when it comes to the coding standards. Even if everyone owns its code, everyone is encouraged to contribute new ideas to all parts of FRI.

As I have discussed and reflected over in this paragraph is it apparent that Idavall unconsciously practice most of the 12 practices that XP is built upon. Of course they are adapted to fit the circumstances at Idavall. Even the main values of XP are present in the development process of FRI. I do not think it is surprising that the development process of FRI has a lot of common with XP. XP fits especially small development teams and is a highly flexible development process in the same way as the development process of FRI.

XP as a fairly new software engineering approach has a lot of common with PD. XP shares a number of similarities with PD approaches in general. It implements an iterative, prototype-based approach, user representatives describe their requirements in a non-formal manner (user stories), integrating users on different levels of the design process, participate in testing etc. The whole process is performed in a strongly iterative manner implementing rapid proto-typing and continuous user involvement. (Rittenbruch, et al. 2002) After all XP has some shortcomings in the context of PD, which I will discuss below.

Shortcomings of XP in the context of Participatory Design
Even if XP implements a highly user-centred approach and the process is strongly iterative some shortcomings of XP in the context of PD are noticed. Rittenbruch, et al. 2002 have noticed this in a research project where an XP approach was used. I have as well noticed that in my study of the development of FRI where an XP -inspired approach was used. In this passage I will discuss some of these shortcomings of XP in this context.

Originally users only participate in a representative manner, as the customer role. XP projects use to have just one user/customer who represents all users. This user probably has a general view, not a detailed view of a particular working context and is selected by the company. In a PD process users often are represented by more than one user, for example in workshops where users from different department participate. If workplace studies are performed the observer may choose work situations that are suitable to represent a wide range of the working context where more than one user is represented. (Rittenbruch et al.2002)

In the development process of FRI users represent many users in a distributed manner. There is no user present at Idavall having the customer role. The requirements are formulated by different kind of users who become aware of shortcomings in their daily working context. These shortcomings are delivered to Idavall as proposals for improvements booth in an electronic way, face to face or by telephone. Rittenbruch et al. provided an *electronic feedback* form, in their project, which enabled users to write user stories whenever they encountered a problem or had a specific requirement. Electronic methods for gathering user feedback become increasingly important in environments where works happens in an increasingly flexible and distributed manner. (Rittenbruch et al.2002)

Rittenbruch, et al. argue that users might choose to describe working situations and their work context, but they might also be quite focused on pure functionality. XP has no means to ensure that the working context is taken into to account. Mostly new functionality is proposed as Rittenbruch, et al have noticed in their project. XP lacks PD methods that concern the understanding of the context users work in. A way to integrate information about working situations and users that are not represented is proposed by Rittenbruch, et al. 2002. They propose a new role into the XP team. Beside the electronic feedback a *user-evaluation customer* should be added to the team to ensure that the working situation is taken into consideration in the design process. This person has studied users in their work environment. The user-evaluation customer has the same rights as other customer during the planning game. This role is meant to represent the user-community based on the user-studies within the planning game.

Workplace studies do not exist in the development process of FRI. Anyway I would say that the developers at Idavall have a very good understanding of their users' working situation. Because of frequent discussions through support, FRI-meetings and courses the developers have developed a picture of the different places of work and special needs of their us-

ers'. Gustaf travels around and meets users as well, which means that he gains a deep knowledge of different places of work. Gustaf, is the person in the first hand at Idavall who has the role as a user-evaluation customer in the development process of FRI. He has a deep knowledge of the customer and their working situations as I mentioned earlier. He expresses the users' interest in discussions and ranking of proposals before changes and improvements of FRI are conducted.

A possible problem would be the large amount of proposals/user stories that are sent in electronically or by telephone. Who is ranking them when the users not are present in the development process? Who is taken the users interests into consideration? As I mentioned earlier it is the staffs at Idavall who do the entire ranking based on several criteria. Rittenbruch, et al. propose an additional role, the gardener, to the XP team whose task is to maintain the user stories. One task is to keep the user stories current. Since the prototype continuously evolve, several of the user stories become out of date. The overall aim for the gardener is to reduce the amount of user stories, to keep them well structured, current and intelligible.

All developers at Idavall act as a kind of gardeners since they have an ongoing discussion of what to implement and they know exactly if some proposals/user stories are out of date. The problem is still who is taken the users interests into consideration. Even if Gustaf in a way acts as the user representative he is not a user representative he is a company representative.

This part has had its focus on the development processes and the architectural structure of FRI. The development processes are highly flexible and has a lot of common with XP. XP stresses, for among other things, user participation in the development process, small releases and short iterations. Idavall is a company that works in a XP-inspired way with its main emphasis on fulfilling user requirements. The architectural structure permits an ongoing development, which facilitates new functionality to continuously be implemented.

In the end of this part I have pointed out some shortcomings of XP in the context of PD. I came to almost the same conclusions as a research group in Australia did. XP has a limited conception of user participation. Although users play an important role during the design process.

In the last part of this thesis I would tie up the previous parts and point out some important issues that are valuable to think of in software development processes in small software companies.

Conclusions

This master thesis has discussed how successful use oriented software development in close cooperation with users can take place in a small software company with small recourses. The success is built, I would claim, on at least three cornerstones; User driven development including Participatory Design (PD), Customer Relationship Management (CRM) and a highly flexible development process with similarities to Extreme Programming (XP).

I would like to highlight and discuss some issues that I think are of great importance when small companies develop software.

Creation of a lasting confidence-based mutual relationship between supplier and customer/user and should be the fundamental objective to a leader of a business enterprise. The use of CRM can help to develop a positive attitude towards the company and its software. If users have a feeling of importance and participation in decisions, concerning the present software they are more willing and inclined to participate in the development process. I am convinced that successful software is dependent of user participation throughout the whole process.

If developers talk to users as experts they gain a respect that is valuable further in the process. This relationship should be built on personal communication for example face-to face communication and/or ear-to-ear communication. I think the reason why several CRM-systems of today are unsuccessful is because of the lack of personal communication. (Computer Sweden 020902) I would say that CRM lacks concrete methods to attain and maintain lasting relationships. However, the register of customers at Idavall is an example of a kind of CRM-system that works. The reason why it works is probably because of the personal knowledge of their users gained through personal communication.

PD and CRM might be a successful combination where CRM enhances PD. I believe CRM can be valuable in the introductory phase as well in the continuous development process. Resources to maintain user participation are at least important in the ongoing process as it is in the introductory phase.

Traditionally PD methods might be a problem to use when users are located in a distributed manner. One way to partially solve this problem is to arrange routines where users are able to give feedback on the software. This feedback can be delivered either electronically, face-to-face or ear-to-ear. A well working support is one opportunity to receive feedback in a personal way. The call taker is able to clarify proposals and requirements. Another way is to arrange distributed meetings where users can meet each other and discuss the software, as a suggestion together with a representative from the company. These meetings serve as breeding grounds for communities of interests as well as opportunities for mutual learning.

There is an obvious difficulty in a distributed environment to perform workplace studies where an ethnographer studies the ongoing activity in a workplace. These workplace studies has the potential to provide designers with new ways of gaining deeper understanding of user work practices and provide a context for designers to collaborate with users. (Blomberg, et al., 1993) It is not just enough and reliable to listen to the users description of their own work because what people say and what they do are not the same. There are many activities that are so much a part of our everyday lives that we are unable to talk about them, sometimes we do not even have the vocabulary that describe our work. (Blomberg, et al., 1993) A way to minimize these disadvantages could be to have a company representative who travels around and visits different workplaces and in that way gain knowledge about different workplaces and activities taken place there.

A flexible development process and a flexible architectural structure of the software are basic conditions that are required in cases like the development of FRI. Extreme Programming is a kind of software development methodology that aims to make software development more flexible and focus on highly flexible environments with quickly changing requirements. The XP approach is highly iterative and strongly influenced and driven by user decisions, which could be of interest in small companies like Idavall. A highly iterative approach makes it easier to quickly adapt the software to new conditions and requirements. XP does not define the use of PD methods such as mock-ups, scenarios or field studies. (Rittenbruch, et al. 2002)

The customer role in the XP-team is not clear I think. Who has the customer role and who selected him/her? Is it a person who is selected by management or by the union? Is it a person who has an overview or a specialised knowledge of the workplace? It is preferable to choose a person in co-operation who has a specialised knowledge I think. A specialised customer role has deeper knowledge of the real work situation where the new software will be used. Has the person authority to decide about bigger improvements and changes that will cost a big amount of money? The customer role ought to have the authority to decide about even bigger changes independently of the management. I think it is not enough to have only one person who represents all customers/users. Rittenbruch et al. propose some additional roles to the team to take care of the users that are not present in the development process. (Rittenbruch, et al. 2002)

Even if the employees at Idavall practice a kind of PD, CRM and XP in the development process of FRI do I think that, after all, most of the success of Idavall and FRI is due to the fact that Idavall is a small company with only five employees. The coffee room is a central place located in the middle of the house where the employees meet several times a day as well as in their offices. Decisions are taken immediately when it is needed in an informal way. All employees at Idavall think that the users are the most important parts of the development process of FRI. They treat them as experts and listen attentively to them regarding improvements of FRI.

A striking utterance from Gustaf was "The development is driven by our users not by ourselves" an expression that in a nutshell describes the relationship between Idavall and their users. I would like to finish this master thesis with that expression. Remaining issues might be a focus for other forums in the future.

References

Beck, K. *Extreme Programming Explained: Embrace Change,* Addison Wesley Pub. Co, 1999

Bjerknesn, G., Bratteteig, T. User participation and Democracy: A Discussion of Scandinavian Research on System Development. In: *Scandinavian Journal of Information Systems,* nr 7(1): 73-79, 1995

Blomberg, J., Giacomi, J., Mosher, A., Swenton-Wall, P. Ethnographic Field Methods and Their Relation to Design. In: Shuler, D., Namioka, A. (eds.), *Participatory Design: Principles and Practices.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1993

Dittrich, Y., Eriksén, S., Hansson, C. PD in the Wild: Evolving Practices of Design in Use. In: T. Binder, J. Gregory, I. Wagner: *Proceedings of the PDC 2002,* Malmö, Sweden.

Ely, M. et al. *Kvalitativ forskningsmetodik i praktiken - cirklar inom cirklar*, Studentlitteratur, Lund, 1991

Eriksén, S. *Knowing and the Art of IT Management – An inquiry into work practices in one-stop shops,* PhD dissertation, Lunds Universitet, Lund, 1998

Fischer, G., Communities of Interests: Learning through the Interaction of Multiple Knowledge Systems, *24th Annual Information System Research Seminar in Scandinavia (IRIS'24),* Ulvik, Norway, 2001

Fischer, G., Ostwald, J. Seeding, Evolutionary Growth, and Reseeding: Enriching Participatory Design with Informed Participation. In: *PDC 2002 Proceedings of the Participatory Design Conference,* CPSR, USA, 2002

Floyd, Ch. Reisen, F M. Schmidt, G. *STEPS to Software Development with Users*. In: Ghezzi G, Mc Dermid J. A. (eds), ESEC '89' Berlin, 1989

Greenbaum, J. A. Design of Ones's Own: Towards Participatory Design in the United States. In: Shuler, D., Namioka, A. (eds.), *Participatory Design: Principles and Practices.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1993

Hallberg, N., Pilemalm, S. Timpka, T. Participatory Design of Inter-organizational Systems: a Method Approach. In: *PDC 98 Proceedings of the Participatory Design Conference,* CPSR, USA, 1998

Henderson, A., Kyng, M. There´s No Place Like Home: Continuing Design in Use. In: Greenbaum, J., Kyng, M. (eds), *Design at Work Cooperative design of computer systems,* Lawrence Earlbaum, Hillsdale, 1991

Jordan, B., Henderson, A. *Interaction Analysis: Foundations and Practice,* 1994

Karlström, D. *Introducing Extreme Programming -An Experience Report,* Third International Conference on eXtreme Programming and Agile Processes in Software Engineering, May 26-29, Sardinia, Italy, 2002

Kensing, F., Blomberg, J. Participatory Design: Issues and Concerns. In: *Computer Supported Cooperative Work 7:* 167-185, Kluwer Academic Publishers, Netherlands, 1998

Kensing, F., Halskov Madsen, K. Generating Visions in Future Workshops and Metaphorical Design. In: Greenbaum, J., Kyng, M. (eds) *Design at Work Cooperative Design of Computer Systems,* Lawrence Earlbaum, Hillsdale, 1991

Lave, J., Wenger, E. *Situated learning, Legitimate peripheral participation,* Cambridge University press, USA, 1997

Rittenbruch, M., McEvan, G. Ward, N. Mansfiels T. Bartenstein, D., Extreme Participation – Moving Extreme Programming Towards Participatory Design. In: Binder, T., Gregory, J., Wagner, I.(eds) *Proceedings of the PDC 2002,* Malmö, Sweden, 2002.

Shuler, D., Namioka, A. (eds.), *Participatory Design: Principles and Practices.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1993

Storbacka, K., Lehtinen J R. *CRM Customer Relationship Management - Leder du dina kunder eller leder dina kunder dig?,* Liber, Malmö, 2000

Törpel, B., Wulf, V., Kahler, H. Participatory Organizational and Technological Innovation in Fragmented Work Environments. In Dittrich, Y., Floyd, C., Klichewski, R. (eds) *Social Thinking – Software Practice,* MIT Press, USA, 2002

Vimarlund, V., Timpka, T. Participatory Design in Economic Terms. In: *PDC 98 Proceedings of the Participatory Design Conference,* CPSR, USA,1998

Not published material
Ekstrand, S., Hansson C. *Design och utveckling av IT-verktyg – ger ringar på vattnet i en organisation,* Bachelor Thesis, Department of Human Work Science and Media Technology, Blekinge Institute of Technology, 2001

Extreme Programming: A gentle introduction http://www.extremeprogramming.org (020815)