Master thesis in Informatics

# Improving IT projects: Can agile methodologies effectively support global IT projects?

IT University of Göteborg

CHALMERS | GÖTEBORGS UNIVERSITET

Sarah Svedberg

Göteborg, Sweden 2007

Department of Applied Information Technology

REPORT NO. 2007:2

# Improving IT-projects: Can agile methodologies effectively support global IT-projects?

- An investigative study at Volvo IT

SARAH SVEDBERG

Department of Applied Information Technology
IT UNIVERSITY OF GÖTEBORG
GÖTEBORG UNIVERSITY AND CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2007

Improving IT projects: Can agile methodologies effectively support global IT projects?
 - An investigative study at Volvo IT
SARAH SVEDBERG

Report no 2007:2
ISSN: 1651-4769
Department of Applied Information Technology
IT University of Göteborg
Göteborg University and Chalmers University of Technology
P O Box 8718
SE – 402 75 Göteborg
Sweden
Telephone + 46 (0)31-772 4895

Chalmers repro
Göteborg, Sweden 2007

Improving IT-projects: Can agile methodologies effectively support global IT-projects?

- An investigative study at Volvo IT

SARAH SVEDBERG
Department of Applied Information Technology
IT University of Göteborg
Göteborg University and Chalmers University of Technology

## ABSTRACT

Agile methodologies were developed to make small scale IT-projects more flexible in order to improve the project's ability to respond and adjust to the constantly changing environment surrounding software development projects. The question investigated in this thesis is whether the agile conceptual framework can be adopted by large companies with global IT-projects as well, and if so; how can it effectively be implemented. By finding out what problems there are in IT projects in large companies, through a number of interviews and a survey with IT professionals and customers of IT projects, is the agile methodology framework discussed based around those problems. The setting is Volvo Information Technology where 10 interviews with employees and costumers of Volvo IT were conducted on the Gothenburg site. The survey was sent out to international Volvo sites with 29 respondents representing 8 Volvo companies in 6 different countries. The thesis begins with an explanation on what agile methodologies are and an overview of Volvo IT and their work practices in IT projects. The interview material and the survey are presented in themes of problem areas which are categorised based on the collected material. The discussion focuses on the problem themes and aspects of agile methodologies. The conclusion is that an agile approach is likely to be beneficial to projects in large companies though the different approach to traditional project methods is making integration into existing work practices the biggest concern. Further research is recommended into which aspects of agile practices are most desirable and during which part of the processes in large IT projects these may be most efficient. Both costumer-supplier and employees on various sites and countries seem to be in agreement on where the problems arise in IT projects, no significant differences were noted in the material. The attempt with this thesis is not to bring additional work processes to the many project models that are already in use in the large matrix organisation, more so to provide material for a discussion on a new approach to work in IT projects.

The report is written in English.

Keywords: agile methodologies, IT projects, software development projects, global organisations

## Acknowledgements

I would like to thank those who have been involved in and supported the writing of my master thesis:

Lina Nykvist, Project Manager, dept. Purchasing Solutions, and Henrik Ott, Business Consultant, dept. Project Control & Finance, my industrial coaches and supervisors at Volvo IT, for their time, guidance, good advice and sharing of information. They have enlightened the “real-world” problems within IT-projects by generously sharing their experiences.

My university supervisor Magnus Bergquist Ph.D., associate professor, Dept. of Applied Information Technology at IT-university of Gothenburg, for guidance, inspiration, good advice and tons of patience throughout the lengthy process.

Everyone who took time out of their busy schedules and agreed to be interviewed; Thank you! Without that input the master thesis would have been considerably more lightweight in its study material.

Everyone who took time out of their busy schedules and responded to the survey and generously shared their views and experiences that way. This proved to be a more interesting source of information than we first anticipated.

Clara Pehrson, business analyst at Volvo IT, for useful advice and ever so helpful lessons on the functions of the Volvo “Violin” intranet.

Thanks again for all your help, input, time and sharing of experience and opinions!

Gothenburg, 10th of January, 2007

Sarah Svedberg

Göteborg, 2007

## TABLE OF CONTENT

**1 INTRODUCTION - 1 -**

1.1 PROBLEM AREA - 2 -

*1.1.1 Not knowing what is needed - 2 -*

*1.1.2 This thesis - 4 -*

1.2 PURPOSE AND OBJECTIVE OF THE THESIS - 4 -

1.3 SCOPE - 4 -

**2 METHODOLOGY - 5 -**

2.1 SCIENTIFIC VIEWPOINT - 5 -

2.2 QUALITATIVE DATA - 6 -

*2.2.1 Interviews - 6 -*

*2.2.2 Selection of interviewees - 7 -*

2.3 QUANTITATIVE DATA - 8 -

*2.3.1 Survey - 8 -*

*2.3.2 Selection of respondents - 9 -*

2.4 COURSE OF ACTION - 10 -

*2.4.1 Establishing the scope of the thesis - 11 -*

*2.4.2 Gathering material - 11 -*

2.5 CRITICISM ON THE COURSE OF ACTION - 12 -

*2.5.1 The interviews - 12 -*

*2.5.2 The survey - 13 -*

**3 THEORETICAL FRAMEWORK - 14 -**

3.1 AGILE METHODOLOGIES – THE VALUES - 16 -

3.2 AGILE METHODOLOGIES – THE PRINCIPLES - 18 -

3.3 AGILE PROJECT METHODS - 22 -

*3.3.1 Adaptive Software Development (ASD) - 23 -*

*3.3.2 Crystal methods - 24 -*

*3.3.3 Dynamic Systems Development Method (DSDM) - 24 -*

*3.3.4 Feature Driven Development (FDD) - 25 -*

*3.3.5 Extreme Programming (XP) - 26 -*

*3.3.6 Scrum - 27 -*

3.4 CRITICISM AND RISKS WITH AGILE METHODS - 28 -

3.5 CONCLUSIONS ON AGILE METHODOLOGIES - 33 -

**4 THE SETTING - 34 -**

4.1 THE VOLVO GROUP - 34 -

4.2 VOLVO INFORMATION TECHNOLOGY - 35 -

*4.2.1 History - 36 -*

*4.2.2 Project models - 38 -*

**5 RESULTS – EMPIRICAL STUDY - 41 -**

5.1 RESULTS FROM THE INTERVIEWS AND SURVEY - 41 -

*5.1.1 Interviews - 42 -*

*5.1.2 The survey - 42 -*

5.2 BEING GLOBAL - 43 -

5.3 ORGANISATIONAL LEGACY - 48 -

5.4 LINE VS. PROJECTS - 53 -

5.5 ROLE DEFINITION - 57 -

5.6 WORK ORGANISATION - 62 -

5.7 IT-PRODUCT VS. BUSINESS VALUE - 68 -

**6 DISCUSSION AND ANALYSIS - 73 -**

6.1 IDENTIFIED PROBLEM AREAS & AGILE METHODOLOGIES .......... - 74 -
*6.1.1 Being global* .......... *- 75 -*
*6.1.2 Organisational legacy* .......... *- 80 -*
*6.1.3 Line vs. Projects* .......... *- 83 -*
*6.1.4 Role definition* .......... *- 85 -*
*6.1.5 Work organisation* .......... *- 89 -*
*6.1.6 IT-product Vs. Business Value* .......... *- 94 -*
6.2 CONCLUSION ON THE DISCUSSION AND ANALYSIS CHAPTER .......... - 98 -

**7 CONCLUSION .......... - 99 -**

**8 AGILE WORK PRACTICES OF INTEREST FOR VOLVO IT .......... - 101 -**

**9 REFERENCES.......... - 1 -**

**10 APPENDIX.......... - 3 -**

10.1 INTERVIEW QUESTIONS .......... - 3 -
10.2 SURVEY INTRODUCTION .......... - 4 -
10.3 SURVEY TEMPLATE.......... - 4 -

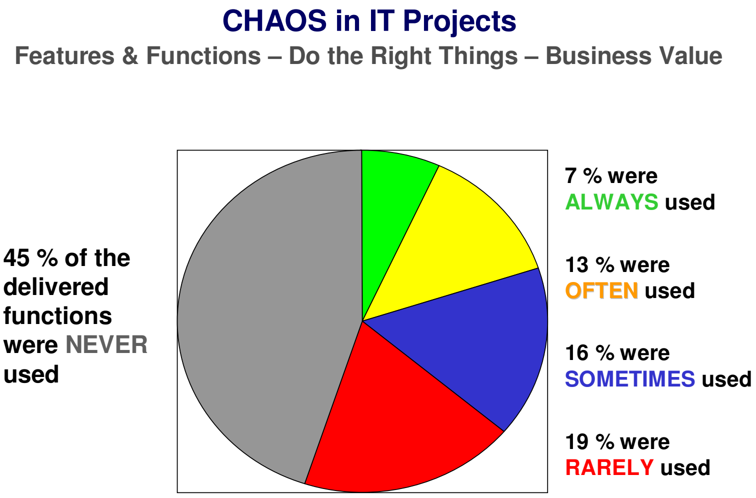
## TABLE OF FIGURES

*FIG 1; CHAOS IN IT-PROJECTS, THE CHAOS REPORT, 1994* .......... - 1 -
*FIG 2; THE VOLVO GROUP ORGANISATION, THE COMPANY STRUCTURE* .......... - 35 -
*FIG 3; LONG TERM FOCUS AREAS AT VOLVO IT* .......... - 36 -
*FIG 4; VOLVO IT - A GLOBAL COMPANY* .......... - 37 -
*FIG 5; VOLVO GROUP IT ORGANISATION – MORE THAN VOLVO IT.* .......... - 38 -
*FIG 6; HOW IS-GDP AND PCM PROJECT MODELS ARE MAPPED TOGETHER* .......... - 39 -
*FIG 7; PROJECT MODELS AT VOLVO IT - HOW THEY ALL CONNECT* .......... - 40 -

# 1 Introduction

*This chapter introduces this master thesis with some background information to give an understanding of the focal point for the research.*

Almost 13 years ago, in 1994, the CHAOS report was published exposing the overwhelming numbers of failures in IT application development projects[1]. The report was compiled from research involving 365 IT executive managers, representing 8 380 different applications. It was the first report of that magnitude and the first report to show such alarming rates of failure in the IT industry. Successful projects accounted for only 16,2 percent, more than half of all projects (52,7 percent) were completed with cost and time estimates overrun and features that did not match the initial scope for the project, almost as much as one third of IT projects (31 percent) were reported to be cancelled before completion. Even though the CHAOS report is now over a decade old it is still referred to in scientific articles and texts and is considered to be the classic report that was a real wake-up call on how much is actually spent on IT investments without being successful. One idea behind the CHAOS report was to find out the reality of IT projects since unlike product development projects - that has had thousands of years to develop and to be assessed – in the 'younger' computer industry it would often be covered up, ignored or rationalised if a projects was unsuccessful, with the results that the same mistakes would be made over and over again.



*CHAOS Report by The Standish Group, a study of 23.000+ IT projects since 1994*

*Fig 1; Chaos in IT-projects, the CHAOS Report, 1994* [2]

One reason to why the CHAOS report is referred to even today would be that the numbers are surprisingly similar in resent reports. There are still an alarming number of IT projects that are unsuccessful or even cancelled before they have reached their full

[1] The Standish Group, http://www1.standishgroup.com/sample_research/chaos_1994_1.php

[2] Image from The CHAOS Report, image from Volvo intranet Violin, 2006

development cycle. Additionally, the amount of projects that deliver features that is not satisfactorily to the customer and consequently never used are a staggering 45% in the 'old' CHAOS report and likewise does that seem to not have changed. An article on software development practices[3] mention that unsuccessful IT projects has as high a percentage as 80-90 percent for not meeting the projects' performance goals, 80 percent delivered late and/or over budget, about 40 percent of projects are failed or abandoned. One recent study showed that 72 percent of all IT projects in Sweden in 2005 was deemed as failed projects[4], this would translate into more than 1,5 billion EUR (14 billion SEK), since 2,15 billion EUR (20 billion SEK) investments were made into IT projects during 2005 in Sweden.

## 1.1 Problem area

There are several challenges presented to an IT application development project team that will affect the outcome of the project's success or failure. The many uncertainties of the surroundings are one and the reality of not knowing exactly what is being produced at the beginning.

### 1.1.1 Not knowing what is needed

One obvious difference between software development projects and product development projects is the difference of the outcome of the project; the product. In product development projects it is possible to know that four wheels and an engine is needed (and much more of course) if someone is going to build a car, also, cars have been built for a hundred years now which means there's the gathered experience from all those years that will help anyone attempting to build a car with knowledge and guidance, likewise building a house, or projects to build a road. Software is different. A company realises that they need *something*, perhaps some form of support in their ordering processes of tools for production purposes, they need to upgrade the existing system or change it completely. The need for a solution is understood, but exactly what the solution may look like or what features it should have is not clear, not for the customer and consequently not possible to know for the IT supplier either. There may also be a reality of not knowing what is possible to create in terms of a solution. It is not possible to manage IT projects in the same way as industrial engineering projects with detailed specifications and control, it is too complex and not feasible for large scale projects.

The reality of not knowing what it is that is needed or indeed is being produced puts additional pressure on a project team in terms of requiring effective communication and collaboration to reach a common goal, additionally, the product itself is evolving and changing throughout the process at a higher speed than any product made out of solid steel[5].

Every IT project team is also challenged with a rapidly changing environment, faced with and adapting to changing market forces, changing system requirements and

[3] Lycett et al, 2003, Migrating Agile Methods to Standardised Development Practice
[4] Exido/IT-barometern, research conducted on behalf of Projektplatsen, http://www.projektplatsen.se/press_arkiv/20051128.html
[5] Coram, M., Bohner, S., 2005, The Impact of Agile Methods on Software Project Management

changing implementation technologies at steadily increasing rate[6], the environmental factors surrounding a project team drives the need to be able to adapt swiftly to the evolving technologies and markets[7]. Every IT application development team is additionally face with the challenge of continuously increase its **productivity** while maintaining and/or improving **quality.** This is a key driver for most software development organisations to look for new ways to improve the success rates for IT projects[8]. There are differences however between the challenges faced by teams in smaller organisations, compared to those in larger organisations.

The larger an organisation is, the greater complexity it involves, and the greater the complexity for the individual project to comply with regarding external control functions and standardised processes that are pre-defined. Because of pre-set delivery dates, dictated by several external factors, a project may have to start before the requirements on the product have been fully specified[9]. In addition, if the requirements given to the development team are abstract the needs to understand the customer increases with the reality of not knowing the customers', equally complex, organisations very well or even have the ability to communicate frequently with the customer. Additionally will a detailed pre-study be time-consuming, even more so with a great complexity involved and by the time it is finalised it may already be outdated[10].

The many challenges that IT projects faces and the alarming reports of overwhelming amounts of unsuccessful, and costly, IT projects sparks the pressing question of what can be done about it. The need for increasing flexibility and ability to respond to changes quickly and inexpensively and also knowing the customer better to understand the requirements and being able to show early progress to the customer have lead large organisations to notice success stories from the emergent agile software development methods. Agile methodologies have the much needed characteristics mentioned above, but were initially developed for smaller software development projects[11] which tend to possess a greater amount of independence compared to those in larger and more complex organisations. This fundamental difference of the basic conditions between large and smaller projects is the main reason to question whether an agile approach is possible for large and complex projects. Several large companies, such as ABB, Daimler Chrysler, Motorola, Nokia[12], Volvo IT[13] and Stena Line IT[14], have during the last few years begun exploring agile methodologies in various ways with pilot projects and more. However the use of agile practices is only in its early stages in larger companies and not being used on a larger scale, the questions remain on whether it can be truly effective

[6] Cockburn, A., Highsmith, J., 2001, Agile Software development: The People Factor
[7] Lindvall et al, 2004, Agile Software development in Large Organizations
[8] Ibid.
[9] Ibid.
[10] Ibid.
[11] Although "IT projects" as a term may involve a greater variety of projects than "software development projects" will the terms be used interchangeably in this thesis, no difference will be made.
[12] Lindvall et al, 2004, Agile Software development in Large Organizations
[13] Volvo IT has its own version of an agile method called Lean-IT which are used in pilot projects at the moment (2006).
[14] Computer Sweden, nr 108, 1st Nov. 2006.

### 1.1.2 This thesis

This thesis address the subject of the possible outcome of implementing agile practices in large organisations and where it could be resulting in positive and/or negative effects. Without making too many assumptions beforehand the focus was to conduct an investigative study on what the problems in large IT projects actually are. By gaining a better understanding of the perceived problems in software development projects and the factors that causes those problems can the problems be structured and an overview of the problems be created. A comparison could then be made with the agile methodologies' strengths and weaknesses and discussed as to form a basis on whether agile methods are likely to be useful to counteract some of those problems.

The setting for this thesis is the very large and global organisation of Volvo Information Technology[15]; it serves as an example of a large IT application supplier, it is likely that other IT development project teams in other large companies experience similar or even identical problems. It can be argued that the fact that Volvo IT has shown an interest in addressing this issue of having a closer look into the perceived problems in IT projects within the own organisation shows a healthy openness to address and deal with the problems, to perhaps administering an alternative approach to the work practices.

## 1.2 Purpose and objective of the thesis

The purpose of this thesis is to investigate whether agile methodologies are likely to be effective if applied in large organisations. In addition, if an agile approach were chosen the question arises on what aspects of agile methodologies would be the most interesting and effective to apply in a large organisation.

**Can agile methodologies effectively support IT projects in large global organisations? - And if so; how?**

## 1.3 Scope

The focus is to bring forth *a general view on IT-projects* in large organisations. This thesis does not give detailed answers to the question whether or not, and if so – exactly how, Volvo IT should apply agile methodologies to their software engineering practices (even though some changes are suggested in the final chapter). More extensive research is needed to answers those questions. However, the aim is to provide material for a discussion on the subject which hopefully can be of interest for involved parties/stakeholders, i.e. project management and project team members, regarding agile methodologies and if /how these methodologies may effectively support the processes in large IT projects. In addition, the thesis will also provide views on actual experienced problems of IT-projects in large organisations.

[15] Volvo Information Technology will be abbreviated Volvo IT in this thesis.

# 2 Methodology

*This chapter will explain how the research in this study was performed and also give a short description on the various scientific methodologies and scientific viewpoint that is the starting point, and also why these were chosen for this study. The course of action for the research will be described here.*

For every research study should a description of the inquiry methodology that was used be given for **replication** purposes which mean that the method should be possible to repeat for someone under similar (identical) circumstances; the result should be possible to control for an outsider outside of the original research process. Also, for **evaluation** purposes to facilitate an evaluation of how the empirical data was collected, the choice of methodology, evaluation of the scientific validity and eventually evaluation of the conclusions in the report which are based on that material[16].

## 2.1 Scientific viewpoint

The objective of this thesis is to gain a deeper understanding of the working situations in IT project teams in large organisations and the problems that arise and may contribute to the large numbers of IT projects that are considered to be failures. First with an understanding of what the problems are in projects is it possible to discuss what causes them and compare them to theories that seem appropriate, in this case agile methodologies, in order to attempt to bring forth suggestions of possible solutions that may counteract those problems.

To gain an in-depth understanding of a phenomenon and to understand the experiences of the people involved is arguably a **qualitative research** approach the most appropriate where focus would be on a smaller selection of study objects (a smaller number of interviewees) that will represent experiences that are likely to be perceived in a wider perspective as well[17].

The scientific viewpoint of **social constructivism** would be the most appropriate for this approach where the research approach focuses on meanings and gaining understanding of a phenomenon through the 'technique' of using conversation to create sense-making. In social constructivism is it not possible for the researcher to be objective since the researcher is indeed part of the process by the act of gathering information in the process. In an interview situation for example should always the aspect of the interviewer's influence on the interviewee and the results as well as the interpretation of the results be taken into account when the conclusions are assessed[18].

[16] Backman, J. (1998). *Rapporter och uppsatser.*
[17] Easterby-Smith, M., (2002). Management Research – An Introduction
[18] Ibid.

There are also aspects of an **ethnographic** approach to this thesis but this was not the primary approach although it was introduced at an early stage. In ethnography it is believed that the very best understanding of a situation is created not by viewing what somebody else is doing, but instead of the researcher taking active part in the everyday setting him or herself[19]. This aspect can be argued to be present in this thesis by my being located in the office area of one department at Volvo IT, seeing the organisation from 'the inside' though without taking part in any projects in the organisation.

This thesis primarily use a combination of the two research approaches for collecting data of **qualitative** and **quantitative** methods through qualitative **interviews** and quantitative data collected through a **survey**. Additionally the interviews were conducted **locally** on the Gothenburg site, and the quantitative survey was conducted **globally** including several international sites which brings the additional combination of local and global data. The research approaches of qualitative and quantitative methods will briefly be explained below.

## 2.2 Qualitative data

**Qualitative** research methods are flexible and dynamic, it allows for great variations in the material and it is used to create a *deeper understanding of the situation* through the collected data and it is the method that can contribute to develop new theories[20], one form of collecting data for a qualitative research study is through **interviews**, which was the primary form of collected material for this thesis. The disadvantages with a qualitative method are that it is time-consuming and it may be difficult to analyse and understand the many aspects in the collected data[21]. The researcher usually takes an active part in the collection of the data, by for example being an interviewer, and as such it is possible for the - although objective - researcher to influence the material, the numbers of sources that may influence the material are many to take into account when the researcher him-or herself is the instrument by which to gather the data[22].

### 2.2.1 Interviews

The fundamental idea in interviews is not to lead the respondent into a particular direction or affect his/her responses in any way[23] however the opinions differ as to whether it is possible for an interviewer to remain objective. In social constructivism it is believed that the researcher at all times will be a part of the phenomenon that is being studied[24].

There are several ways to conduct interviews in; the approach used for this thesis was that of **semi-structured interviews** where the interviewer has a set of questions to follow but the interview is somewhere between an interview and a discussion between the interviewer and the interviewee, this type of interview allows the interviewer to follow up on leads from the interviewee, to gain a deeper understanding of feelings and opinions of the interviewee, semi-structured interviews are used when the concern is

[19] Dourish, P., (2006). Implications for design.
[20] Ibid.
[21] Easterby-Smith, M., (2002). Management Research – An Introduction
[22] Backman, J. (1998). *Rapporter och uppsatser*.
[23] Easterby-Smith, M., (2002). Management Research – An Introduction
[24] Ibid.

what people think and how they understand what they do[25]. For this thesis the approach to use semi-structured interviews seemed most appropriate since the objective was to gather an understanding of the perceived problems in IT projects from the interviewees' collected experiences and opinions. With a few set open questions[26] the interviews were free to move in any direction that seemed more relevant.

### 2.2.2 Selection of interviewees

The criteria by which the interviewees were chosen were, somewhat bluntly put, "*various people with experience from having worked in IT-projects with Volvo IT, preferably with department Project Control & Finance and/or Purchasing Solutions*", which are the departments where the supervisors of this thesis are employed. The thesis should by that be of value for those departments and at the same time would those departments serve as case study for other departments of Volvo IT. The material should at the same time be of general research value for the university since, to the knowledge of those involved in making this report, no similar study has been performed at Volvo IT on the subject of addressing agile methodologies before and the thesis can by that contribute to the research on agile methods' usability in very large global software engineering development companies.

The attempt to have an objective viewpoint opens the door to allowing many subjects to come up in the process; the interviewees' backgrounds were varied and as such likely to give very different views on perceived problems. At the same time it would be interesting to see whether there would be similarities in the results or indeed big differences despite the open and objective starting point.

The fundamental idea in interviews is not to lead the respondent into a particular direction or affect his/her responses in any way[27]; therefore did the idea to base the interviews on a few open questions seem as an appropriate approach. With a few set open questions[28] the interviews were free to take on any direction the interviewee felt was more relevant. The interviews were also anonymous since the aspect of 'who said what' has no scientific value, it is the opinions and experiences that are more interesting of the professionals that are interviewed, and also will the anonymity allow the interviewees to perhaps speak more freely and honestly on the subject[29].

**Facts on the interviews**

- All the interviewees are Swedish and work on the Volvo site in Gothenburg.
- All interviews lasted around one hour in time.
- All interviews were conducted in Swedish and later translated into English.
- The totals of interviews are 10. The number of interviewees from Volvo IT is 7, and interviewees from the customer side are 3.

The working roles that were represented among the interviewees from Volvo IT are *Account Manager, Project Manager, Business Analyst, Maintenance Manager, Group*

[25] Easterby-Smith, M., (2002). Management Research – An Introduction
[26] See Appendix for the template of interview questions
[27] See previous passage on semi-structured interview.
[28] See Appendix for the template of interview questions
[29] Easterby-Smith, M., (2002). Management Research – An Introduction

*Manager* and *Business Consultant*. From the customer side was all interviewees *Project Managers* except for one *Global Process Manager* (with an employment history as Project Manager).

Because the interviews were anonymous and several of the working roles at Volvo IT represented in this study unique were no separation of the results made by separating into working roles/positions. The categorisation of the interview results that appeared to be the most relevant was that of two categories; that of Volvo IT (the supplier) and the customer side. The interviewees will be coded as V IT 1, V IT 2…from the Volvo IT side and C1, C2 …and so on for the customer side.

## 2.3 Quantitative data

**Quantitative** research methods are sometimes referred to as the traditional approach to research methods; it is structured and more concerned with frequency and may result in numeric calculations[30], examples of quantitative research methods are questionnaires, test, experiments, and **surveys**. The advantages of quantitative methods are their economic and less time-consuming characteristics (compared to qualitative research) but additionally they can also be criticised for being inflexible and unable to pick up on subtle nuances in the data[31]. This thesis uses quantitative data in the form a survey as an additional source of data to the qualitative interviews.

### 2.3.1 Survey

There are several opinions on the positive and negative aspects with surveys. Surveys are one way to reach a large number of respondents, but on the downside a survey can never cover all aspects of one respondent's opinions in the same way a face-to-face interview can. This was also pointed out by one respondent at the end of the survey where the respondents were asked to add their own comments;

> *"This survey was interesting but I guess the proposal for answers on some questions were a bit too restrictive. It is not only 1 parameter that makes a project turn to a success or a nightmare. It depends really on a complex context that can not be "explained" just with one "main" reason."*

The survey[32] was sent out to a total of 47 recipients, with both the Volvo IT and the customer side represented. An approximately 62 percent response rate gave 29 individual responses within two weeks (and one reminder). A result which was considered to be very good compared with response rates from similar surveys on the company intranet, according to the Volvo IT supervisors for this thesis, and since the respondents are generally very busy with fully booked schedules.

The survey was anonymous. Again; 'who says what' has no scientific value other than if related to a certain category of interest, for example a specific job category or decision level, company background, gender or as in this case where the different views from the Volvo IT versus the customer side. Having spent time at Volvo IT and discussed the

[30] Backman, J. (1998). *Rapporter och uppsatser.*
[31] Easterby-Smith, M., (2002). Management Research – An Introduction
[32] See Appendix for the complete survey template

subject with various people in various roles it seems to be the question which sparked most interest was; *what does the customer say*?

The Volvo intranet “violin” has a function which allows surveys to be set up and easily accessed by those added as “contributors”. The researcher’s role would be that of the “administrator” and also the only one to have access to all answers and ability to change the settings for the survey. After some initial tests and rewritings of the survey, it was sent out as a web link along with an explanatory/introductory email to all recipients[33]. After the first answer was received from one respondent no changes were made to the survey - in order to not affect the comparable data from the answers.

The questions in the survey was very much influenced by the first interviews, those results gave an idea about what to focus on and which areas that may be relevant.

### 2.3.2 Selection of respondents

The criteria by which the respondents were chosen was similar to how the interviewees were chosen; employees within the Volvo Group who has worked near or in IT-projects in one way or another, preferably with the department Project Control & Finance and/or the department Purchasing Solutions **-** for the data to be more applicable locally for the two departments that supervised this master thesis. The respondents were suggested by the Volvo IT supervisors for the thesis and a colleague from dept Purchasing Solutions. As mentioned above the main criteria were that the respondents were likely to have opinions on, and experiences from, IT-projects. Also, by focusing and limiting the survey to employees who have worked near or with those two departments the results may be more up to date and relevant for implementing possible changes for just those departments. It gives a high relevance to the thesis if the results are practically usable and not only theoretical. The general relevance of the study is not effected by this since the result is still of general interest regarding IT-projects. However the traditional approach to selecting respondents for a traditional quantitative research method, as a survey, would be a statistical selection method as not to bring a bias to the results and establish the aspect of generalisation to the result[34]. The selection process for the respondents in this survey is therefore not the ideal scientific process.

Out of the 29 respondents are 13 working for Volvo IT in Sweden, France, Brazil and more than half the total work for Volvo IT in the USA (8 respondents). Remaining 16 respondents can be viewed as customers of Volvo IT. The customers are predominantly based in France but also, Sweden, Belgium and Mexico. All are employees at various companies within the Volvo Group, for example Renault Trucks, Volvo Trucks, Mack Trucks, Volvo Powertrain and more. In total there are 6 different countries and 8 different Volvo Group companies represented among the respondents.

Only one respondent had worked for Volvo for less than one year, 11 respondents had worked for the Volvo Group for 1-5 years, 10 for 6-15 years and 7 for more than 16 years; this means almost 60 percent of the respondents had been Volvo employees for more than 6 years. There were no bias as to whether customers or Volvo IT employees had worked longer for Volvo, it shows an even spread of work experience all over.

[33] See Appendix for the email invitation to the survey
[34] Backman, J. (1998). *Rapporter och uppsatser.*

On the question "Have you worked in IT projects at Volvo?" did one of the respondents reply "No". This was explained by the respondent having sat in steering committees for IT-projects but had not personally been involved in work on an operational level in a project group. Nevertheless the respondent stated that in the role of being a steering committee member the respondent had got a good insight into the work and problems regarding IT-projects.

**Facts on the survey**

- The respondents work on various Volvo sites in Sweden, France, Belgium, USA, Brazil and Mexico.
- The survey was conducted in English.
- The survey was anonymous
- The total of respondents is 29. The number of respondents from Volvo IT is 13 and those from the customer side are 16.
- 8 different Volvo Group companies in 6 different countries are represented.

The working roles represented among the respondents are: *Project manager, Group manager, Department manager, Business Analyst, Global process manager*, *Business* (Customer) *project manager, Process manager, Quality and Process manager, Project executions solutions manager, Quality coordinator, Supervisor sales, and System analyst.*

The categorisation of the survey responses which appear to be the most relevant here is the same as with the interviews: two categories; that of Volvo IT and the customer side. The responses will be coded as Volvo IT = **[V IT]** and Customer = **[C]** in the results chapter.

## 2.4 Course of action

Volvo IT supplied information on project models and interviewees and supervised the work to secure its business value. Gothenburg University Dept. of Applied Information Technology and IT-university supervised and followed this master thesis to secure a general scientific value of interest in the research.

There were discussions at the beginning of the study about partaking in an actual on-going project at Volvo IT to see the problems 'from the inside', with the identified risk that the project may be put on hold or run into problems during the study period. However, it was decided that no appropriate projects were available at the time, at the relevant departments, when the research study for the thesis would start.

The approach for this research study became that of talking to as many various roles as possible from both Volvo IT as well as the customer side, to bring forward a palette of what the most common problems are, and where or when they do arise. Early on it was agreed that the most interesting and efficient approach to this work would be to make no prior assumptions on the problem areas. Instead the course of action would be to go out in to the company and pose the open question: *Where, in your opinion, do the problems occur in IT-projects?*

### 2.4.1 Establishing the scope of the thesis

The scope of the thesis was developed in collaboration with Volvo IT and Gothenburg University, the meetings included me – the author of this thesis, my supervisors at Volvo IT and my supervisor from IT-University/Gothenburg University, with the aim to find a problem area to focus on that would not only be of use for Volvo IT, but also had a general scientific research value for the university. It was a general agreement that the initial scope may come to alter and change somewhat during the working process.

### 2.4.2 Gathering material

After the scope for the thesis was agreed upon in meetings at Volvo IT, the work began with collecting research material and data. The data was gathered from these sources:

- **Literature studies.** Library databases and the internet were used to find scientific articles and books on the subject that may be relevant.

- **Information on the company intranet**. Access to the global Volvo intranet, “Violin”, with information on project models, work processes, company structures, company rules, guidelines and much more.

- **Interviews.** Interviews with employees at Volvo IT and their customers in other Volvo Group companies. Qualitative data. Locally.

- **Survey**. The creation of a survey which was sent out to Volvo sites worldwide via a function in the intranet “violin”. Quantitative data. Globally.

Interviews were decided early on to be one almost natural source for collecting material in this case. The work proceeded with contacting potential interviewees and booking them for interviews[35]. Parallel with the work with the interviews the idea came up to use the company intranet which has a template to make a survey[36]. The idea with this was to find out if the results would be similar or indeed different from the interviews with the survey being possible to send outside the national borders. The survey could reach respondents worldwide at various Volvo sites, and this would bring the aspect of quantitative data to the research.

Which theory to analyse and to view the results by were under discussion on several occasions though agile methods was mentioned already at the very beginning in this research and was continually reoccurring in discussions since. It was clear that it should be an area to focus on although the approach of the thesis was not obvious from the start[37].

The data was gathered and later grouped into identified problem areas; themes. Literature studies on agile methodologies and software engineering development projects at large companies were conducted continuously during the writing of the master thesis.

[35] See further chapter Interview
[36] See further information in the chapter Survey
[37] See further the Theory chapter

## 2.5 Criticism on the course of action

Even more material could have been collected from conducting more interviews, taking part in more meetings, participation in project meetings, and more. This is an issue in all research studies – when is the gathered material enough? When there is enough to answer the initial question satisfactorily - is the obvious answer. That however is not at finite answer. When “enough is enough” is in many cases more an opinion, rather than anything that is given. Given the parameters of; time (to start and finish the research during; one university semester), magnitude of the research (it is a master level thesis, not more or less) and the fact that it has one author (it is common to share this type of work between 2 or 3 students) has led me to the opinion that the collected material is ample for this purpose, even too great to be able to pick up on all the nuances in it during the time frame.

No material or interviews were made with the department at Volvo IT that already works with an in-house version of an agile approach, called *Lean-IT*. At the time for this study the practices were new and no studies had yet been published regarding success rates of those pilot studies. The Lean-IT department was also not directly related to the departments which this research would focus on.

With an even greater literature study preceding the collection of empirical data, from interviews and survey, the collection of the data may have been more focused and controlled from the start. The possible negative aspects from a greater literature study at the beginning could be that the focus could end up being too ‘narrow’ with an overly confident belief of ‘knowing’ what the problem areas are already beforehand, thus not keeping an open mind regarding the problem areas that would be brought up.

### 2.5.1 The interviews

The interviews may have strayed a little far from the main focus sometimes; *what is the cause for most problems in IT–projects*, to be more about certain project models and other specific details. At the beginning it was not clear what direction the thesis would take as the work progressed, I was therefore hesitant to rule out anything too early in the interviews, it could be important later on in the research but it was not possible to not know it at the time when the interviews were conducted.

Had the interviews been stricter and more focused on facts and figures it may most likely have resulted in a material that would have been more comparative. Instead the questions asked here and the focus of the study was that to collect and gather *opinions* and *experiences* from employees involved in IT-projects; in other words - a focus on softer values. Instead of numbers and technicalities the focus is on the people behind them, therefore is the results chapter a presentation of the interpreted opinion rather than hard facts.

The interviewees could have been chosen through a more careful selection process. This would have given a somewhat more scientifically justifiable result, by, for example, interviewing employees from only one work category or perhaps only Project Managers from both Volvo IT and the customer side. This may have brought forward a slightly different approach as to how the results could be compared and analysed.

### 2.5.2 The survey

The material from the survey has been used as an additional source of data to confirm, or challenge, the material from the interviews and has not been 'properly' evaluated as it would have been had it been the only source of empirical data.

The fact that the respondents were not chosen by some random statistical process can be argued to be "colouring" the results since the traditional scientific approach to quantitative research is to select respondents through a more statistical process[38], but it can also be argued that it meant that relevant respondents were approached with valid and interesting opinions for the subject.

[38] Backman, J. (1998). *Rapporter och uppsatser*.

# 3 Theoretical framework

*In this chapter the aim is to present necessary information on the theories that the thesis is based around in its Discussion and Analysis Chapter. Some historical background to agile methodologies and a brief description on six project methods that are labelled as agile is presented. Further the values and twelve principles that form a central part in agile methodologies are presented.*

> From February 11 - 13, 2001, at the Lodge at Snowbird ski resort in the Wasatch Mountains of Utah, 17 people met to talk, ski, relax and try to find common ground. What emerged was the **Agile Software Development movement**. Representatives from Extreme programming (XP), Scrum, the Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Crystal Methods, Feature-Driven Development (FDD), Pragmatic Programming, and others sympathetic to the need for an alternative to document-driven, rigorous software development processes convened. What this meeting produced – ***The Manifesto for Agile Software Development***, signed by all 17 participants – was symbolic. (Highsmith, 2002) [39]

The above is an extract from the preface in Jim Highsmith's *Agile Software Development Ecosystems* (2002) where he talks of the origins of what has since become known as the **Agile Software Development movement**. Preceding this meeting was a growing discontentment regarding *document-driven, rigorous software development processes*[40] in the 1990's, the Information Age. The wish to develop an alternative approach which would take in to account more of the human factors and be more flexible towards an ever changing environment contributed to several new and smaller software engineering processes –as those mentioned above- to evolve, more or less irrespective of each other. Many ideas were similar on how software engineering projects should be conducted, though all these lightweight methodologies had their own founders, leaders, communities and terminology. Similarities between these methodologies were their underlying values and ideas that they all considered to be of utmost importance for successful software development work.

A previous meeting with Extreme Programming leaders in 2000 invited "sympathisers", to what at the time was referred to as lightweight methodologies, to the meeting[41] (Highsmith, 2002). That meeting sparked the idea and wishes to start a form of collaborative work to aid development and foster a greater understanding of all those lightweight methodologies. This resulted in the workshop in Snowbird, Utah, in 2001 where seventeen programmers and methodologists gathered[42]. "A bigger gathering of organizational anarchists would be hard to find" (Fowler, Highsmith, 2001)[43]. At the meeting the name "agile" was decided upon "to act as an umbrella name for the various

[39] Highsmith, J., 2002, *Agile Software Development Ecosystems*, p.xvii
[40] Ibid., p.xvii

approaches"[44] since "lightweight" - which was the term mostly used at the time - was not considered to "accurately convey the essence of what these approaches were about"[45] by the workshops attendees.

> Agile = Eng. dictionary; *having the faculty to move quickly and gracefully; mentally quick*

What marked the Snowbird meeting in 2001 as significant was that, apart from the common term "agile" was agreed upon, all 17 participants from different methodologies and backgrounds agreed on what *values*[46] and *principles*[47] that was their common ground and what summons up the fundamental ideas of agile software engineering[48]. These values and principles were written down during the gathering and signed by all participants in what was named the *Agile Manifesto*. All 17 participants have since, somewhat jokingly, been referred to as 'parents' of the Agile Manifesto and the Agile Software Development movement.

The importance of having written down common values and having a common name can not be underestimated, none of the meeting's participants expected the kind of attention the "movement" gained afterwards, the meeting kick-started the Agile Software Development movement as a phenomenon in the software engineering development community.

> The effort clearly struck a nerve; I think we were all very surprised at the degree of attention and appreciation the manifesto got. Although the manifesto is hardly a rigorous definition of agile, it does provide a focusing statement that helps concentrate the ideas. (Fowler, 2005)[49]

The basic point of view that forms the agile development movement is that the environment in which software development projects are implemented in is dynamic and ever changing and instead of ignoring that fact – there should be methods that embrace and welcome change even late in the development process, instead of working against them[50].

[41] Highsmith, J., 2002, *Agile Software Development Ecosystems*,p. xix
[42] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#AgileManifesto
[43] Fowler, Highsmith, (2001), Dr.Dobb's Journal, article, http://www.ddj.com/dept/architect/184414755
[44] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#AgileManifesto
[45] Ibid.
[46] See chapter on Agile methodologies – the values
[47] See chapter on Agile methodologies – the principles
[48] The principles were started at the meeting but mostly developed on a 'wiki' afterwards, Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html (2005).
'Wiki' = [...] a website that allows the visitors themselves to easily add, remove, and otherwise edit and change some available content, [...]. This ease of interaction and operation makes a wiki an effective tool for collaborative authoring,
[49] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html (2005)
[50] http://www.agilesweden.org/metoder.htm

## 3.1 Agile methodologies – the values

The values in the Agile Manifesto emphasise the human and communicative aspects of software engineering rather than the technical aspects[51] and the values are the fundamental basis by which the agile software engineering work is designed. The values of agile methodology are expressed as follows in the Agile Manifesto[52]:

> *We are uncovering better ways of developing software*
> *by doing it and helping others do it.*
> *Through this work we have come to value:*

| | | |
|---|---|---|
| ***Individuals and interactions*** | *over* | *processes and tools* |
| ***Working software*** | *over* | *comprehensive documentation* |
| ***Customer collaboration*** | *over* | *contract negotiation* |
| ***Responding to change*** | *over* | *following a plan* |

> *That is, while there is value in the items on the right,*
> *we value the items on the left more*[53]*.*

It does not mean that there is no documentation done in agile projects but it does mean that if a customer were to chose between documentation and another feature in the software the customer would most likely chose the feature, and as such, agile methodologies say the features should be valued as more important than the documentation.[54]

The values may give rise to questions such as what does it mean and it can mean anything from how it is interpreted. Martin Fowler and Jim Highsmith (co-authors of the Agile Manifesto) commented on the values in an article in 2001[55]. They pointed out that it was as many as 17 experienced and recognized software development "gurus" that agreed to the statement in the first place which may be the first aspect that should be noted with it. The word *uncovering* was selected to indicate that the members of the Agile Alliance do not have all the answers, nor do they subscribe to the silver-bullet theory[56]. *By doing it* indicates that the members practice these methods themselves in their own work and *by helping others do it* show that the idea is to further the own knowledge through the helping of others.

[51] Migrating Agile Methods to Standardized Development Practice, Lycett et al, 2003
[52] http://agilemanifesto.org/
[53] Manifesto for Agile Software Development, http://agilemanifesto.org,
[54] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html
[55] Fowler, Highsmith, (July 16th, 2001) Dr.Dobb's Journal, article, http://www.ddj.com/dept/architect/184414755, also Highsmith, J., (2002) *Agile Software Development Ecosystems* , p.xvii
[56] Silver-bullet theory = *the metaphor of the* ***silver bullet*** *applies to any straightforward solution perceived to have extreme effectiveness. The phrase typically appears with an expectation that some new technology or practice will easily cure a major prevailing problem.* Brooks, F. (1975) *The Mythical Man Month*, chapt. 14.

> Ken Schwaber (a proponent of SCRUM) told of his days of selling tools to automate comprehensive "heavy" methodologies. Impressed by the responsiveness of Ken's company, Jeff Sutherland (SCRUM) asked him which of these heavy methodologies he used internally for development. "I still remember the look on Jeff's face," Ken remarked, "when I told him, 'None—if we used any of them, we'd be out of business!'". [57]

Each bullet point states a preference in the first segment and is followed by something that is of lesser importance. The distinction between them is where the heart in agility lies. The latter segment is however not without importance.

> *"Yes, we value planning, comprehensive documentation, processes and tools. That is easy to say. The hard thing is to ask 'what do you value* ***more****?!"* Roy Singham at ThoughtWorks, about that it is the edge cases, the hard choices, which interest him. [58]

It is recognised by the Alliance that **processes and tools** are important, but it is also recognised that the **interaction** between skilled individuals in a project is of even greater importance. **Comprehensive documentation** may be important in some projects but it can never be more important than the final product and delivering **working software**. Every team should decide for themselves which documentation is absolutely essential for their specific project.

Internal project charters or external legal contracts are not believed to be the best way to create an understanding of each other and to understand and deliver what the customer wants. Only through on-going **collaboration** can any real understanding be created and therefore is **contract negotiation** said to be insufficient for the purpose.

Even successful projects very rarely deliver what was planned in the beginning. Instead they may be considered to be successful because they were agile enough to respond to the **changing requirements** throughout the process. **A fixed plan may** even become counterproductive for a project if it is not allowed to change to respond to external changes. [59]

The values have since been altered numerous times and interpreted differently by various practitioners to suit needs of individual projects which is something agile methodologies encourage since an understanding that no two projects are exactly the same is essential[60]. The fundamental ideas however remain the same as those in the Agile Manifesto as the work continues to evolve in the software development community.

[57] Fowler, Highsmith, (July 16th, 2001) Dr.Dobb's Journal, article, http://www.ddj.com/dept/architect/184414755, also Highsmith, J., (2002) *Agile Software Development Ecosystems* , p.xvii

[58] Ibid., ibid.

[59] Ibid., ibid.

[60] *Agile & Iterative Development – A manager's guide,* Larman, C., 2005, p.29, also The Declaration of Interdependence for Modern Management, http://alistair.cockburn.us/index.php/The_declaration_of_interdependence_for_modern_management

## 3.2 Agile methodologies – the principles

The principles stated in the agile manifesto states somewhat more specific, though not in detail, the various areas that an agile work process affects. As with the values (above) the principles play an important role of reference in agile methodologies.

Principles of agile methodology according to the Agile Manifesto[61]:

**1.** Our highest priority is to satisfy the customer through early and *continuous delivery* of valuable software.

**2.** *Welcome changing requirements*, even *late in development*. Agile processes harness change for the customer's competitive advantage.

**3.** Deliver working software frequently, from *a couple of weeks* to *a couple of months*, with a preference to the shorter timescale.

**4.** Business people and developers must *work together daily* throughout the project

**5**. Build projects around *motivated individuals*. Give them the *environment* and *support* they need, and *trust* them to get the job done.

**6.** The most efficient and effective method of conveying information to and within a development team is *face-to-face* conversation.

**7.** *Working software* is the primary measure of progress. Agile processes promote sustainable development.

**8.** The sponsors, developers, and users should be able to *maintain a constant pace* indefinitely.

**9.** *Continuous* attention to technical excellence and good design enhances agility.

**10**. *Simplicity*--the art of maximizing the amount of work not done--is essential.

**11**. The best architectures, requirements, and designs emerge from *self-organizing teams*.

**12.** At regular intervals, the *team reflects* on how to become more effective, then tunes and *adjusts* its *behaviour* accordingly

**Comments on the principles of the Agile Manifesto**
Following are the principles of the Agile Manifesto with comments on their meanings.

**1. Our highest priority is to satisfy the customer through early and *continuous delivery* of valuable software.**
About seeing software development through the customer's perspective; by asking the question "what does the *customer* actually want?". The answer is believed to be "working software" and nothing else. In addition, it is believed that the best way to show the project's progress to the customer is through *continuous deliveries* throughout the development cycle so that the customer can actually see and fully understand what is

[61] Manifesto for Agile Software Development, http://agilemanifesto.org/principles.html/

being developed. The first of all these small deliveries should also be as early as possible, the sooner the better.

> "They [requirements specifications and architecture documents] are important, but we need to understand that customers don't care about documents, UML diagrams or legacy integration. Customers care about whether or not you're delivering working software to them every development cycle—some piece of business functionality that proves to them that the evolving software application serves their business needs." Jim Highsmith[62]

**2. *Welcome changing requirements*, even *late in development*. Agile processes harness change for the customer's competitive advantage**
Change and turbulence is a fact in business, every economy and technology. Instead of trying to prevent it from happening it is more productive to create working processes that welcome changes, even embrace it, and understand that it is either way inevitable. It is more effective to view change as an opportunity rather than a threat. In addition, this incorporates the aspects of welcoming feedback and understanding that a result of feedback being accepted – is change. [63]

**3. Deliver working software frequently, from *a couple of weeks* to *a couple of months*, with a preference to the shorter timescale.**
The ideas behind iterative and incremental software development are not new, but in agile it is heavily emphasised that every delivery cycle should be shortened as much as possible. The difference between delivery and release is pointed out saying that; multiple and rapid *deliveries* of a product with ever-growing functionality allows everyone to evaluate and learn from the growing product, *release* on the other hand is when valid functionality has been achieved regarding the whole product and the business can put the product into production. Part-deliveries allow everyone involved; customer *and* supplier; to understand, and see, the progress throughout the project's life-cycle. [64]

**4. Business people and developers must *work together daily* throughout the project**
This principle represents the radical change to the 'traditional' view of the requirements process. Work *with* the customer in the project, preferably on a daily basis with an active and committed customer, and share the responsibility *with* the customer for the software project. The background for this principle is the belief that software is most often *not possible* to buy in a similar way as one can by a car for example - with a set list of requirement and features.

Instead of a detailed list of requirements at the beginning of a project is a high-level view of requirements preferable. This high-level view of requirements will be subject to frequent change, which is one reason to not have it too detailed. Instead of relying on a detailed list of requirements to set the ground for design and coding is frequent

[62] Fowler, Highsmith, (July 16th, 2001), Dr.Dobb's Journal, article, http://www.ddj.com/dept/architect/184414755
[63] Ibid.
[64] Ibid.

interactions between the business people and the developers recommended in agile projects. [65]

**5. Build projects around *motivated individuals*. Give them the *environment* and *support* they need, and *trust* them to get the job done.**
"People trump processes". It is up to the people in a project to make the project work, regardless of how many processes, tools and technologies there may be; it is the people using them that make the difference. Agile heavily emphasise the importance of having the right and properly skilled people on in a project from the start.

For many people, trust is the hardest thing to give. Decisions must be made by the people who know the most about the situation. This means that managers must trust their staff to make the decisions about the things they are paid to know about[66].

**6. The most efficient and effective method of conveying information to and within a development team is *face-to-face* conversation.**
Agile do not preach that *no* documentation is efficient, but rather that documentation; as a form of conveying and sharing knowledge is far subordinate to face-to-face communication. The emphasis is the actual *understanding* of the facts and not whether there ought to be *documentation or not*. "Tacit knowledge cannot be transferred by getting it out of people's heads and onto paper," (Nancy Dixon, *Common Knowledge*, Harvard Business School Press, 2000). "Tacit knowledge can be transferred by moving the people who have the knowledge around. The reason is that tacit knowledge is not only the facts but the relationships among the facts—that is, how people might combine certain facts to deal with a specific situation."
The distinction between agile and document-centric methodologies is not extensive documentation versus no documentation; rather a differing concept of the blend of documentation and conversation required to elicit understanding.[67]

**7. *Working software* is the primary measure of progress. Agile processes promote sustainable development.**
"Working software is the measure of progress because there's no other way of capturing the subtleties of the requirements: Documents and diagrams are too abstract to let the user 'kick the tires,'" (Dave Thomas, co-author of *The Pragmatic Programmer*, Addison-Wesley, 1999) [68].

Several aspects of a project may be on time but it may still be unsuccessful if the final stages take too long or if several bugs are discovered late in the progress. With iterative development it is said that by continuously integrating working software into the product bugs and problems with integration are discovered continuously throughout the project instead of appearing as surprises in the end, when it may be too late to fix them without exceeding the deadline. In addition, working software is the best way for

[65] Fowler, Highsmith, (July 16th, 2001), Dr.Dobb's Journal, article, http://www.ddj.com/dept/architect/184414755
[66] Cockburn, A., Highsmith, J., 2001, *Agile Software Development: The People Factor,* , also Fowler, Highsmith, (July 16th, 2001), Dr.Dobb's Journal, article, http://www.ddj.com/dept/architect/184414755
[67] Fowler, Highsmith, (July 16th, 2001), Dr.Dobb's Journal, article, http://www.ddj.com/dept/architect/184414755
[68] Ibid.

everyone involved to see and understand what is being developed and the team and the customer will be able to see that the project is moving in the right (or wrong) direction. [69]

**8. The sponsors, developers, and users should be able to *maintain a constant pace* indefinitely.**
Long working hours do not lead to improved productivity. Errors occur when team members are tired and overworked and errors take additional time to correct. The software development industry has in many cases made overtime into the standard working procedure and it does not support agility for a team, it may even be counterproductive. An agile team relies on its team members, the most important component of a project, to be alert and creative and maintain that alertness and creativity for the full length of a software development project. Sustainable development means finding a working pace (i.e. 40 hours a week) that the team can sustain over time and remain healthy. [70]

**9. *Continuous* attention to technical excellence and good design enhances agility.**
To maintain agility and flexibility in a project is quality of the design emphasised. However, the design is at the same time expected and encouraged to change throughout the process in agile projects, the requirement are expected to change as the code is being written and this complicates the traditional view of creating a design first and code later. The various methodologies in agile has different ways of addressing this. FDD has a design phase leading in to each new iteration, usually graphically with the UML. XP emphasises refactoring[71] as the developers review the code continuously and this allows the design to evolve throughout the development of the product. Nevertheless, all of these processes borrow from each other: FDD uses refactoring as developers revisit earlier design decisions, and XP encourages short design sessions before coding tasks. In all cases, the project's design is enhanced continually throughout the project. [72]

**10. *Simplicity*--the art of maximizing the amount of work not done--is essential.**
Include only what everybody needs rather than what anybody needs. It is much easier to add or remove something to a simple process, than it is to change one that is complicated. To improve agility a minimalist approach to the development process and design is encouraged.

"Simple, clear purpose and principles give rise to complex, intelligent behaviour," (Dee Hock, former CEO of Visa International). "Complex rules and regulations give rise to

[69] Fowler, Highsmith, (July 16th, 2001), Dr.Dobb's Journal, article, http://www.ddj.com/dept/architect/184414755
[70] Ibid.
[71] Refactoring = rewriting programming code to improve its readability, maintainability, structure or performance *without* changing its functionality or behaviour. Informally referred to as "cleaning up the code". Each refactoring should be very small and accompanied by automated tests of the code to ensure that the behaviour of the code has not changed. The reason to perform refactoring is to improve understandability and make human maintenance easier in the future. *The impact of Agile Methods on Software Project Management*, Coram,M., Bohner,S., 2005, also; http://www.refactoring.com/, also; Larman, C., 2005, *Agile and Iterative Development – A Manager's guide* , p.149.
[72] Fowler, Highsmith, (July 16th, 2001), Dr.Dobb's Journal, article, http://www.ddj.com/dept/architect/184414755

simple, stupid behaviour." No methodology can ever address all the complexity of a modern software project. Giving people a simple set of rules and encouraging their creativity will produce far better outcomes than imposing complex, rigid regulations.[73]

**11. The best architectures, requirements, and designs emerge from *self-organizing teams*.**
The first point of this principle; the best designs (architectures, requirements) emerge from iterative development and use, rather than from early plans. Form does not follow function: Form follows failure; the "form follows function" idea has misled architects into believing that they could predict how buildings would actually be used. The second point of the principle is that emergent properties (*emergence,* a key property of complex systems, roughly translates to innovation and creativity in human organisations) are best generated from self-organizing teams in which the interactions are high and the process rules are few. [74]

**12. At regular intervals, the *team reflects* on how to become more effective, then tunes and *adjusts* its *behaviour* accordingly**
Trust in people, believing that individual capability and group interaction are key to success - extends to trusting teams to monitor and improve their own development processes. Agile methods should not be adopted and followed slavishly. It is preferable to start with one of these processes and be aware and recognise that it is not possible to come up with the one process that is right for every situation. Therefore, any agile team must refine and reflect as it goes along, constantly improving its practices in its local circumstances. [75]

Co-authors of the agile manifesto that all attended the Snowbird workshop in 2001[76]:

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

## *3.3 Agile project methods*

So how are the agile methodologies put in to practice? As with all methodologies there may be as many different versions of the methods as there are projects and this is also the main idea behind agile methodologies – proponents of the methodology do not claim to have all the answers that will guarantee success for any given project, modifications on the different approaches are encouraged[77].

[73] Fowler, Highsmith, (July 16th, 2001), Dr.Dobb's Journal, article, http://www.ddj.com/dept/architect/184414755
[74] Ibid.
[75] Ibid
[76] http://agilemanifesto.org/
[77] Highsmith, J., 2002, *Agile Software Development Ecosystems*, p. 336, also Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#ShouldYouGoAgile

The project methods that are mentioned in the literature on agile methodologies vary slightly but those that are mentioned most as the original agile methodologies will be presented here in a brief overview.

### *3.3.1 Adaptive Software Development (ASD)*

Created by James (Jim) Highsmith (co-author of the Agile Manifesto) the method contains practices for iterative development, feature-based planning, customer focus-group reviews and an agile management philosophy called Leadership-Collaboration management[78]. With a focus on working with prototypes is the method helpful in building large, complex systems with an iterative approach[79]. The method focuses on "outlining learning and collaborative techniques and theories for all project members" and brings in a role of a *collaboration facilitator* whose job it is to be "planning collaboration, rather than simply letting it happen"[80] and as such the method can be argued to have a focus on collaborative and learning based communication including between the customers and the developers. ASD has been criticised for not being detailed enough as a practical work method[81].

The ASD life cycle six main characteristics are:

- **Mission focused** - The detailed requirements in a project may not always be clear from the beginning but the mission is often well articulated from the start and with a constant refinement of the mission that gives a direction to the iterative processes and a direction to where the project should be headed and guidance for project decisions.
- **Feature-based** - The focus lies on results rather than tasks in the project. Results are application features which are useful for the customer. Features may be software features that are useful for the customer or documentation in the form of user manual's that has a direct result for the customer. Data models may be defined as a deliverable but is considered to be secondary to features of a direct use for the customer.
- **Iterative** and **Time-boxed** - Time-boxing is about deliver features to the customer in short time-frames and to make the right decisions to cut back on in order for the time-boxes to work. The idea with delivering work in short time-boxes comes from the fact that the environment is ever changing and to encourage change something is needed to force frequent deliveries.
- **Risk driven** - The critical risks are to be analyzed in all iterations.
- **Change tolerant** - Change is viewed as a competitive advantage when adapted to quickly instead of as a problem[82].

The ASD basic life cycle specific practices:
*Speculate*: Project initiation and Adaptive cycle planning
*Collaborate*: Concurrent feature development
*Learn*: Quality review and Final questions and answers and Release[83].

[78] Highsmith, J., 2002, *Agile Software Development Ecosystems*, p. xxxii
[79] http://www.agilesweden.org/metoder.htm
[80] Martin, Biddle, Noble, 2003, *How do XP, Scrum and ASD build the right software?*
[81] http://www.agilesweden.org/metoder.htm
[82] Highsmith, J., 2002, *Agile Software Development Ecosystems*, p.312
[83] Ibid., p. 313

### *3.3.2 Crystal methods*

The family of Crystal methods created by Alistair Cockburn (co-author of the Agile Manifesto) with a focus on the people working in the projects, collaboration, good citizenship and cooperation are all important aspects[84]. There are various methods to choose from (named as crystals; Clear, Yellow…)[85] and the determining factor of which to chose is the size of the project, staff size, project priority and how critical the system is – priority and criticality should be decided upon and "maps" will recommend which version of Crystal methods is most suitable[86]. Cockburn has interviewed project teams worldwide on what people say *should work* and *what actually does work* and this influences his work.

> *Focusing on skills, communications, and community allows the project to be more effective and more agile than focusing on processes and plans.*
>
> Alistair Cockburn[87]

### *3.3.3 Dynamic Systems Development Method (DSDM)*

Developed in England in the mid 1990's DSDM is focused on customer solutions and business value[88]. The method in considered to have the best-supported training and documentation in Europe of all agile methodologies[89]. One fundamental idea is to have ***fixed resources but allow the content of the iterative processes be flexible***[90]. The method has nine basic principles which are similar to those of the Agile Manifesto, for example; active user involvement, frequent delivery, team decision making, integrated testing throughout the project life cycle and perhaps most notably; *reversible changes in development*. The last principle should be interpreted as because of consistent configuration management throughout the process all changes made to documents or code should be possible to change back into the state it was before the change – thus enabling a more allowing attitude towards mistakes - because everything is possible to reverse back to its previous state - this may bring an attitude that allows for more rapid decision makings as well[91].

The DSDM development process is divided into five major phases;

- **Feasibility Study** – the project is assessed, is the method right for the project? This phase takes a few weeks to create a *development plan*.
- **Business Study** – a *system architecture definition* and *prototyping plan* is outlined based on the main characteristics of the business and the technology. All these may change as the project continues.
- **Functional model iteration** – iteration in time-boxes with; *prioritised list of functions*, *functional prototypes review documents*, *non-functional requirements* and *risk analysis of further development*.

[84] Highsmith, J., 2002, *Agile Software Development Ecosystems*, p. xxxii
[85] Larman, C., 2005, *Agile and Iterative Development – A Manager's guide* , p.37
[86] Ibid. p.37, also http://www.agilesweden.org/metoder.htm
[87] Alistair Cockburn coauthor of the Agile Manifesto, *Agile Software Development Ecosystems*, Highsmith, J., 2002, p.261
[88] Highsmith, J., 2002, *Agile Software Development Ecosystems*, p.251
[89] Ibid., p. xxxii
[90] http://www.agilesweden.org/metoder.htm
[91] Highsmith, J., 2002, *Agile Software Development Ecosystems*, p.254

- **Design and build iteration** – iteration in time-boxes which produces the system to a minimum of customer requirements, through iterations and increments the system is systematically built and created alongside the customer's review after each iteration.
- **Implementation** – also iterative in its form, the formal phase where the system is transferred to the customer and the final product.[92]

### *3.3.4 Feature Driven Development (FDD)*

Created by Jeff De Luca and Peter Coad, FDD is a framework for processes for design and development with short iterations with an emphasis on quality[93]. Two key roles in FDD are *chief architect* and *chief programmer* and the work is based around *a list of processes* of *five steps*;

- **Develop an overall model** (should take 10% of the working hours initially and 4% ongoing), a high-level object model and notes, identify and understand the fundamentals of the domain that the system is addressing, and throughout the project it should be added to this model to reflect what is being built.
- **Build a features list** (should take 4% of the working hours initially and 1% ongoing), grouping the features in to sets and related areas.
- **Plan by feature** (should take 2% of the working hours initially and 2% ongoing), the identification of class owners and the identification of feature set owners.
- **Design by feature** and the following step: **Build by feature** (should take ca 75% of the working hours for design and build combined). The majority of the effort on an FDD project is placed here including tasks such as detailed modelling, programming, testing, and packaging of the system.[94]

The FDD process model asserts that:
A system and structure for building other systems is necessary in order to scale to larger projects; a simple, well-defined process works best; process steps should be logical and their worth immediately obvious to each team member; "process pride" can keep the real work from happening; good processes move to the background so team members can focus on results; short, iterative, feature-driven life cycles are best[95].

> As the name implies, features are an important aspect of FDD. A feature is a small, client-valued function expressed in the form <action><result><object>. For example, "Calculate the total of a sale", "Validate the password of a user", and "Authorize the sales transaction of a customer". Features are to FDD as *use cases* are to the Rational Unified Process (RUP) and *user stories* are to XP – they are a primary source of requirements and the primary input into your planning efforts. (Scott Ambler)[96]

[92] Coram, M., Bohner, S., 2005, *The Impact of Agile Methods on Software Project Management*,
[93] http://www.agilesweden.org/metoder.htm
[94] Ambler, S. at: http://www.agilemodeling.com/essays/fdd.htm, also
Highsmith, J., 2002, *Agile Software Development Ecosystems*, p.273,
[95] Highsmith, J., 2002, *Agile Software Development Ecosystems*, p.273
[96] Scott Ambler on FDD at; http://www.agilemodeling.com/essays/fdd.htm

### 3.3.5 Extreme Programming (XP)

Created by Kent Beck and Ron Jeffries (co-authors of the Agile Manifesto), XP is the most well known of all agile methodologies and the one most documented and therefore also the one which is the easiest to implement[97]. The term extreme programming was first used in 1997 but the ideas origins from the 1980's[98]. Its basic values are Communication, Feedback, Simplicity, Courage, and Respect[99] in order to effectively develop high quality programming code which in turn will provide improved flexibility[100].

> "…It then elaborates these [**values**] into fourteen **principles** and again into twenty-four **practices**. The idea is that practices are concrete things that a team can do day-to-day, while values are the fundamental knowledge and understanding that underpins the approach. Values without practices are hard to apply and can be applied in so many ways that it is hard to know where to start. Practices without values are rote activities without a purpose. Both values and practices are needed, but there's a big gap between them - the principles help bridge that gap".[101]

The XP practices include; the whole team working together in one room, constant refactoring[102], a strong emphasis on test-driven development and pair programming with two programmers at one computer to improve quality of code. The method is team-and communication-oriented with customers, developers, managers working together in the same area/room as a team with a focus on producing high quality code for software of high business value[103].

XP faces requirements difficulties by recommending that;
The customer is an integral part of the team and should be on-site with the rest of the project group; the customer writes user stories and discusses each requirement directly with the programmers; the customer is responsible for *all* business decisions including prioritising user story development; the small 2-3 week iterations allow the user to evolve their requirements based on concrete working software; the customer regularly tests the software to confirm it works as expected[104].

[97] Lindvall et al, 2004.*Agile Software Development in Large Organizations*,
[98] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#XpextremeProgramming
[99] Ibid.
[100] http://www.agilesweden.org/metoder.htm
[101] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#XpextremeProgramming
[102] Refactoring = rewriting programming code to improve its readability, maintainability, structure or performance *without* changing its functionality or behaviour. Informally referred to as "cleaning up the code". Each refactoring should be very small and accompanied by automated tests of the code to ensure that the behaviour of the code has not changed. The reason to perform refactoring is to improve understandability and make human maintenance easier in the future. Coram,M., Bohner,S., 2005, *The impact of Agile Methods on Software Project Management*, also; http://www.refactoring.com/, also; Larman, C., 2005, *Agile and Iterative Development – A Manager's guide* , p.149.
[103] Larman, C., 2005, *Agile and Iterative Development – A Manager's guide*, p.36, 140
[104] Martin, Biddle, Noble, 2003*How do XP, Scrum and ASD build the right software?,*

The life cycle in XP of six phases;

- **Exploration** – a few weeks to a few months; the phase where the team becomes familiar with the technology and practices at hand, and the customer gathers/provides requirements for the first release. Story card writing (features), estimating, and prototyping.
- **Planning** – several days, work with the customer to plan and prioritise for the first release, developers estimate the need for resources on their part, and team management plan a first release schedule. Story card writing and estimating.
- **Iterations to release** – each iteration takes one to four weeks and there are several iterations preparing for the first release and the next phase. Each iteration of programming is finalised with functional tests. Task writing and estimating.
- **Productionising** – further performance testing to meet the customer requirements. New changes may be added here or be recorded as changes to be added during the next phase. Training, marketing and documentation.
- **Maintenance** – corrective, perfective and adaptive changes and feature requests produced in new iterations alongside further testing. Building major releases. There should be fewer and fewer customer requests at the end of this phase to lead in to the next phase, otherwise return to the **Iterations to release** phase for the next iteration.
- **Death** – completion of all necessary documentation and the release of the product to the customer is planned. This phase occurs when there is no further incentive for additional changes; too expensive to change and low investment value[105].

### *3.3.6 Scrum*

Mostly developed by Ken Schwaber, Jeff Sutherland, and Mike Beedle in the 1980's and 90's[106] Scrum is a very 'light' **project management method** modelled on the game of rugby which it borrows its terminology from[107]. Daily meetings, short iterations, self-directed teams and constant estimations on remaining time in the project are practices to aid in this goal-oriented method. The project management methods of Scrum are sometimes combined with the more practical engineering techniques and practices of XP[108] which can work as a compliment to each other.

Example on key practices in Scrum are;

- Self-directed and self-organising teams
- No external addition of work to an iteration once chosen
- Daily (15-30 minutes) stand-up meetings with special questions (Example: *What have you done since the last meeting? Do you have any obstacles? What will you do before the next meeting?*)[109]
- 30-calender day iterations (called "sprints")

[105] Coram, M., Bohner, S., 2005, *The Impact of Agile Methods on Software Project Management*, also Larman, C., 2005, *Agile and Iterative Development – A Manager's guide*, p.142

[106] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#Scrum

[107] Coram, M., Bohner, S., 2005, *The Impact of Agile Methods on Software Project Management*, also http://www.agilesweden.org/metoder.htm

[108] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#Scrum

[109] Schwaber, K., at http://www.controlchaos.com/

- Demo to external stakeholders at the end of each iteration
- Client-driven adaptive planning in each iteration[110]

The life-cycle in Scrum consists of three phases with the first phase divided into two sub-phases;

- **Pre-game phase**
  - **Planning** – establish the vision – writing it down, set the expectations, secure funding in a budget, define the system in a *Product Backlog* (which is often updated with features and modifications including all known requirements - all which are prioritised and resource/effort estimated), exploratory design and prototypes. The *product backlog* is controlled by one person only, regardless of how many contribute to it; the *Product Owner*.
  - **Staging, Architecture/High level design** – more requirements are identified and the design and prototypes are elaborated based on the *Product Backlog*.
- **Development** – iterative development cycles; *sprints*, which span from one week to 30 days. Each sprint includes requirements, analysis, design, evolution and delivery, with daily sprint meetings and defining of the sprint backlog and sprint reviews. Three to eight sprints are executed in a development process before completion of the product/system.
- **Release, Post-game** – documentation, training, marketing & sales, delivers the system in one release with no additional modifications which concludes the project and effort.[111].

## 3.4 Criticism and risks with agile methods

Even the fiercest proponents of agile methodologies say that "using agile is not for everyone"[112] and that it is no silver-bullet theory. Various types of criticism from different directions targeting different aspects of the agile approach have emerged as the approach has been renowned and more and more widely used. The proponents of agile say that in many cases it is most often a lack of understanding agile that is creating the problems and that anyone contemplating to try an agile approach must understand that it is no quick fix that is appropriate in any type of situation[113].

### *Gain the support of upper management*

It is of utmost importance before embarking on any agile project methods to have the understanding and support of high level management. Without the support of higher management the likelihood of succeeding with a project is dramatically decreased. However, the agile approach represents a fundamental cultural change for high level, and executive, management with for example one or more of the following; less documentation, possibly no set release dates and no set features for every release as well as no predictive, detailed, planning - since this goes against the agile way of thinking – and that in turn goes against most management approaches, which makes it complicated.

[110] Larman, C., 2005, *Agile and Iterative Development – A Manager's guide*, p. 109

[111] Coram, M., Bohner, S., 2005, *The Impact of Agile Methods on Software Project Management*, also Larman, C., 2005, *Agile and Iterative Development – A Manager's guide*, p.113

[112] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#ShouldYouGoAgile

[113] Ibid.

Without the understanding and support from high level management it is not likely to succeed with any agile project[114]. The Standish Group, which produced the CHAOS Report, outlines the top criteria for successful projects and *executive support* is stated as the most important criteria to have[115].

The large companies ABB, Daimler Chrysler, Motorola and Nokia have through collaboration compared their experiences with agile pilot projects and this aspect of trust and support from upper management is one issue that was highlighted. Although all the companies knew of reports of many agile success stories they needed to know whether agile practices actually would work for them and whether a large company with established standard processes could at all adopt agile methods to develop large, complex, safety-critical systems that would be maintained for decades[116]. The studies did show similar success rates from all four companies; that agile practices could help to bring agility to large companies with traditional developments processes, there were reports that final products exhibited higher quality than it had previously, that flexibility improved in responding to changing requirements quicker, implementation was finished quicker alongside other positive aspects. However it was pointed out that it had to be done by assessing the situation and chosen projects well beforehand and well understand the different work practices and the changes they would bring. Surveys on the results among some of the developers showed that +80 percent of the respondents/team members thought the team morale had increased, that they felt more confident with the work from the XP-influenced pair programming (than had they done the same work alone), that the learning curve for new engineers were dramatically shortened, there was a general higher confidence in the quality of the design and the code. All those pilot projects were said to have succeeded in improving one or more of the following; customer satisfaction, quality, productivity, and cost. However, important observations were made that in several cases did agile methods indeed clash with the traditional established methods and values, which in some cases resulted in double work being done (for example double documentation to fulfil different level of management requests, double quality testing – both in the project's own and the standard quality management tests), the pilot projects also reported clashes with the traditional Change Control Board (CCB) who had decided to make a decision on a change request and by the time the decision had been made by the CCB the design of the product had already changed to the extent that more extensive changes had to be made in order to accommodate the change decided upon by the CCB (thus simultaneously decreasing the agility for the project), and conclusions from this were that more work had to be put into "*integrating the agile approach into the existing ones*" since the integration issue seemed to be more a cause for problems rather than the agile influenced approach itself [117].

The importance of trying out an agile approach on the right type of project is essential in order to gain success and also understanding from higher management. If not a suitable project is chosen for a pilot project it may not be a good idea to try a new agile approach

[114] Coram, M., Bohner, S., 2005, *The Impact of Agile Methods on Software Project Management*,
[115] The Standish Group, http://www.pm2go.com, also Cockburn, A., Highsmith, J., 2001, *Agile Software Development: The People Factor*
[116] Lindvall et al, 2004.*Agile Software Development in Large Organizations*,
[117] Ibid.

at all according to Martin Fowler[118]. He suggests a smaller project preferably of not a too high critical factor but still slightly more critical "than what is comfortable" or no one will care whether the project succeeds or not, which also eliminates the incentive to try and change existing processes for new agile methods.

### *Communication issues in the team*

XP and Scrum do not specifically emphasise the often difficult task of facing the on-site customer, prioritizing the requirements and features nor the gathering of requirements, other methods may be needed to assist and complement those aspects of possible problems.[119]

### *Communication between teams*

One aspect of an agile approach the pilot projects at ABB, Daimler Chrysler, Motorola and Nokia discussed above[120] noted was that larger companies per definition often have larger projects with more project members often based in various different physical locations. This was concluded as one problem that remained to be solved when working with agile project practices that emphasise real time, direct, face-to-face communication in preferably small teams. The way to solve this was partly by minimising the need for inter team communication as much as possible and partly by organising small workshops with representatives from all smaller teams at the beginning of the project, and later at regular intervals, to bring everyone up to speed with the iterative progress everyone (every small team) was at. This is also an approach Jim Highsmith[121] suggests for large projects separated into smaller teams and/or situated in multiple physical locations which often can be the case for large or global companies. However Highsmith writes that this should be an exception rather than a rule as face-to-face communication and working physically together in the same area can not be replaced with, however many, workshops[122]. As concluded in the pilot projects at ABB, Daimler Chrysler, Motorola and Nokia is this a problem, of inter team communication that remains to be solved[123] although there are agile projects that are reportedly effective in large projects of +100 team members over several continents (one example mentioned is ThoughtWorks)[124].

### *No detailed planning*

The workshop's (mentioned above) agendas should not be set beforehand in a predictive way but should only be addressing the iteration at hand and plan for the next iteration to follow, in order to supports an agile approach for every team. Again, this poses a cultural change to most management theories – to not have a set plan for every stage ahead in a project; it directly goes against the ideas of the traditional approach with detailed plans for every gate the project reaches. This can cause hesitation among various management levels to use an agile approach[125].

[118] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#ShouldYouGoAgile
[119] Martin, Biddle, Noble, 2003, *How do XP, Scrum and ASD build the right software?,* also Lindvall et al, 2004., *Agile Software Development in Large Organizations.*
[120] Lindvall et al, 2004., *Agile Software Development in Large Organizations*,
[121] Highsmith, J., 2002, *Agile Software Development Ecosystems*, p.249
[122] Ibid.
[123] Lindvall et al, 2004., *Agile Software Development in Large Organizations*,
[124] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#ShouldYouGoAgile
[125] Highsmith, J., 2002, *Agile Software Development Ecosystems*,

***Staff turnover***

In large companies with large projects that may run over a period of several years the staff turnover is inevitably a serious problem for agile projects. Agile methods rely on the involved individuals to form a project team with its collective knowledge and that the most effective way to spread information from one individual to another individual is to have those individuals based physically together, and the aim is to minimise ineffective work by drastically cutting back on the amount of unnecessary documentation that no one ever reads in order to rather spend that time more effectively improving the product[126]. Thus can the agile approach be counterproductive for itself in a large project in this sense, but there are ways to lessen the effects of the loss of an important project member although it can of course not be minimised completely. One way is to use team rotation which also generally increases knowledge and understanding throughout the team for all the various work assignments in the team, this also increases individual development and skills among the team members which makes it easier for someone to take over somebody else's job duties should it be necessary. Another way is to make sure the produced code is kept 'clean' and easy to comprehend and maintain which would make it easier for any new member in the team to quicker come up to speed with previous progress. However these approaches would not have an effect should there be a loss of customer or a customer representative that has been a team member. One approach can instead be to tailor the approach to the individual project (which agile encourages) and when needed keep some of the traditional practices of detailed documentation, though keeping in mind that it should be kept to a minimum if the agility of the processes are not to be too effected[127].

***Hostile towards problem solvers***

Skowronski, V. (2004)[128] speaks of how agile methods can "*marginalize problem solvers*" and that agile methods may be hostile towards the best programmers by encouraging continuous team-work, saying that such working practices does not allow for individual thinking processes – which requires time to consciously and sub-consciously work and *think* about a problem. Skowronski does the comparison with great thinkers of our time (Isaac Newton and more) that much of their best work came out of working and thinking in solitude and also pointing out that difficult problems may not be possible to solve through brainstorming sessions in groups but may need experience and knowledge from outside the appointed project group. The agile proponents on the other hand argues that people working together under conditions of good and effective communication and interaction can operate at higher levels and be more effective than they had had they only been using their talents individually, and that this is proven over and over in brainstorming and problem-solving sessions[129].

[126] Highsmith, J., 2002, *Agile Software Development Ecosystems*,
[127] Ibid., also Fowler, M. at
http://www.martinfowler.com/articles/newMethodology.html#ShouldYouGoAgile
[128] Skowronski, V., 2004, *Do Agile Methods Marginalize Problem Solvers?,*
[129] Cockburn, A., Highsmith, J., 2001, *Agile Software Development: The People Factor,*

***Risks with direct feedback***
Skowronski (2004)[130] also speak of the risk with direct feedback - something agile methods encourage between project members, developers and customers; that it needs to be carefully controlled, and even when it is controlled it may still pose a risk for killing new ideas prematurely if not supported correctly.

***Developers "should not be 'people-people'"***
Skowronski[131] comments on the 'people-centred' aspects of agile methods saying that software engineering people primarily solve technical problems and not problems related to people, and because of that a focus on the *technical* aspects is much more efficient. Also saying that too much interaction with other people may hamper or interfere with these people's (the developers) focus on 'things' and technicalities[132], thus arguing against the fundamental values of agile. The agile values states that software development is far from a solely technical activity, and also does Cockburn and Highsmith conclude that agile individuals have a difficult time to function well in a rigid organisation with traditional methodologies, and vice versa. The agile approach tends to grow and span teams, organisations and other forms of working relationships as well[133].

***Costumer involvement***
Agile principles and values dictate bringing the customer out of the traditional customer role where he/she/they are only present at the beginning of a project to set up the requirements, and later when acceptance testing and release are planned. The agile approach says that it is not possible to understand each other fully from the beginning, even more so when systems become increasingly complex, and the traditional methodology of meeting up, producing a product, deliver product, and then after that go back to implement change requests; is time-consuming and inefficient. In addition, the requirements are changing as the environment for a company/system is constantly changing and thus new requirements will be applied. The fundamental idea behind agile is to use work practices that welcome change and deals with it effectively and flexibly, and this, it is argued, requires a customer to be present and preferably involved in the work as a team member throughout the project. Additionally, it should not be just any customer representative but a customer present who is; committed, knowledgeable, collaborative, representative, empowered, and know and understand what is required for the end user. If the customer has no power to make decisions on features in the product development it may still cause delays and hamper agility for the project. However, that type of customer may not be available or even willing to take the 'agile' role of involvement in a software project; hence the agile approach may not be appropriate or desirable for the project at hand[134].

> "If customers do not collaborate, then you will not see the full advantages of an adaptive process. Having said that we have found on several occasions that we

[130] Skowronski, V., 2004, *Do Agile Methods Marginalize Problem Solvers?,*
[131] Skowronski, V., 2004, *Do Agile Methods Marginalize Problem Solvers?,*
[132] Ibid.
[133] Cockburn, A., Highsmith, J., 2001, *Agile Software Development: The People Factor,*
[134] Coram, M., Bohner, S., 2005, *The Impact of Agile Methods on Software Project Management*, also Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#ShouldYouGoAgile, and also Highsmith, J., 2002, *Agile Software Development Ecosystems,*

> have worked with customers who did not want to collaborate, but changed their mind over the first few months as they begun to understand the agile approach". Martin Fowler[135]

## 3.5 Conclusions on agile methodologies

There are many ways through which an agile approach may be adopted into the work processes of an IT project, there are several project methods to draw ideas and influences from, with extreme programming (XP) being the one which is most widely used and also most tested today. What the different methods have in common are a set of **values** that they all emphasise through their various work practices, however the methods were originally developed independently with a different terminology for the functions in the various methods as a result, which may be argued to make it slightly confusing when referred to as *one* methodology. Several factors are critical for the success of an agile project and the most important may be the one to understand that everyone must make their own evaluation of the process to make it work in the right way for the specific project in a specific organisation[136]. One fundamental idea in agile development is that the team can be more effective in responding to change if it can; **reduce the cost of moving information between people** (by placing people physically closer/together, replace documents with talking in person and at whiteboards, improve the team's amicability – a sense of community and morale – to make people more inclined to share valuable information quickly) and **reduce the time between making a decision and seeing the consequences** of the decision (by making user experts available to the team or even better; apart of the team and work iteratively and incrementally) [137]. The idea is to both be able to **create change** and to be able to **rapidly respond to change** in the turbulent, high speed and uncertain business and technology world[138].

[135] Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#ShouldYouGoAgile
[136] Cockburn, A., Highsmith, J., 2001, *Agile Software development: The People Factor*
[137] Ibid.
[138] Ibid.

# 4 The setting

*This chapter intends to give necessary information of the setting for this study; The Volvo Group and more specifically Volvo Information Technology. Company structure will briefly be explained and project models will be given an overview.*

Volvo Information Technology is the setting for this study, a global IT application supplier for the Volvo Group and some of the group's external customers. The study object serves as the example from where experiences and phenomena can be expected to be related to other organisations with similar attributes. To further the understanding of the setting is the Volvo Group structure briefly presented and following that is Volvo IT given a brief presentation.

## 4.1 The Volvo Group

Volvo may for the individual consumer be mostly associated with safe family cars but that was only one part of the organisations' productions which today no longer is owned by the Volvo Group (Volvo Car Corporation is today a Ford-owned organisation). The Volvo Group is one of the worlds' leading producers of **trucks**, **buses**, **construction equipment**, and **drive systems for marine and industrial applications** as well as **components for aircraft engines**. The Volvo Group also provides financial services to its customers[139]. Founded in 1927 the Volvo Group today employs approx. 82 000 worldwide with production in 18 countries and sales in some 185 countries. The Volvo Group has focused exclusively on transport equipment for commercial use since 1999.

> The company vision:
>
> ***"The Volvo Group's vision is to be valued as the world's leading supplier of commercial transport solutions."***

Most of the Volvo Group's sales are for markets in Western Europe and North America. An important element of the Volvo Group's strategy is to strengthen its positions throughout the world in markets that show high growth potential.[140]

**The Volvo Group is organised in to <u>business AREAS</u> which are:**

Mack (trucks)
Volvo Trucks
Volvo Construction Equipment (CE)
Volvo Financial Services
Renault Trucks
Volvo Buses
Volvo Penta (marine)
Volvo Aero

[139] Volvo annual report 2005, www.volvo.com

[140] Volvo annual report 2005, www.volvo.com

**The business areas are *supported* by <u>business UNITS</u> which are:**

Volvo 3P
Volvo Parts
**Volvo Information Technology**
Volvo Business Services
Volvo Powertrain
Volvo Logistics
Volvo Technology Transfer
Volvo Technology

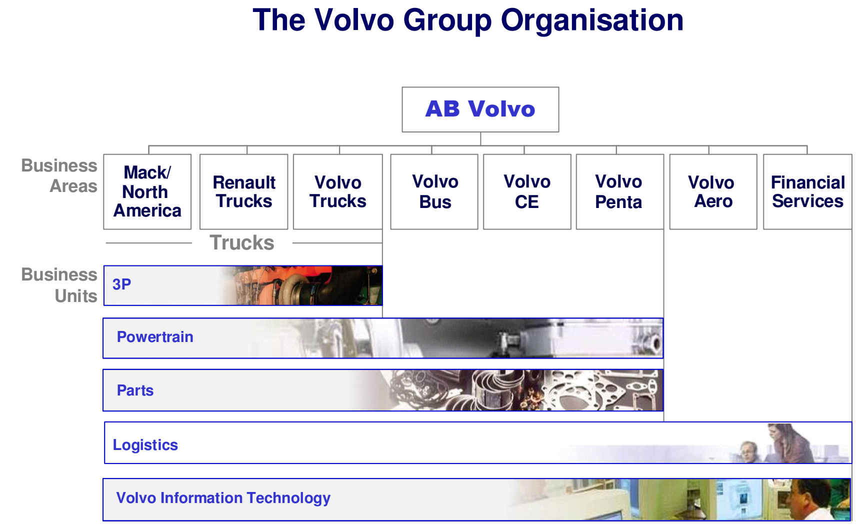


*Fig 2; The Volvo Group Organisation, the company structure* [141]

## *4.2 Volvo Information Technology*

Volvo IT is a company – within the company - dedicated to support the other business areas and business units in the entire Volvo Group with IT-solutions, maintenance and support. Volvo IT AB is a wholly owned subsidiary of the Volvo Group.

Volvo IT **provides IT solutions** for **the whole industrial process**, from product development and manufacturing to sales, the aftermarket and administration, including IT operations and infrastructure. Clients include the Volvo Group, Ford-owned Volvo Cars, and other large industrial companies. The largest customers are in the auto industry but the number of customers in other industries has increased in recent years.[142]

Volvo IT states their long term focus areas through which the company aims to improve its business. The long terms focus areas are: *Profitability & Growth* – meaning that the company will strive to increase its business in the Volvo Group and with its customers; *Customer Satisfaction* - to improve customer satisfaction by being professional, enterprising and be easy to be doing business with, also through new business opportunities, shorter lead-times, improved delivery precision, quality and cost-efficiency to deliver better business value for its customers; *Global Provider* – to build a business that supports the global expansion of the Volvo Group and also utilise any synergy effects that may bring within the own organisation; *Attractive Workplace* – to continue to be regarded as an attractive workplace as Volvo IT has received awards for

[141] Volvo intranet Violin, 2006
[142] Volvo annual report 2005, www.volvo.com

being "Sweden's best workplace". There shall also be a focus on *industrial IT* with a focus on closer working relationships with the customers, to provide business- and IT-change and even operational strategies for the customer and its key processes. [143]

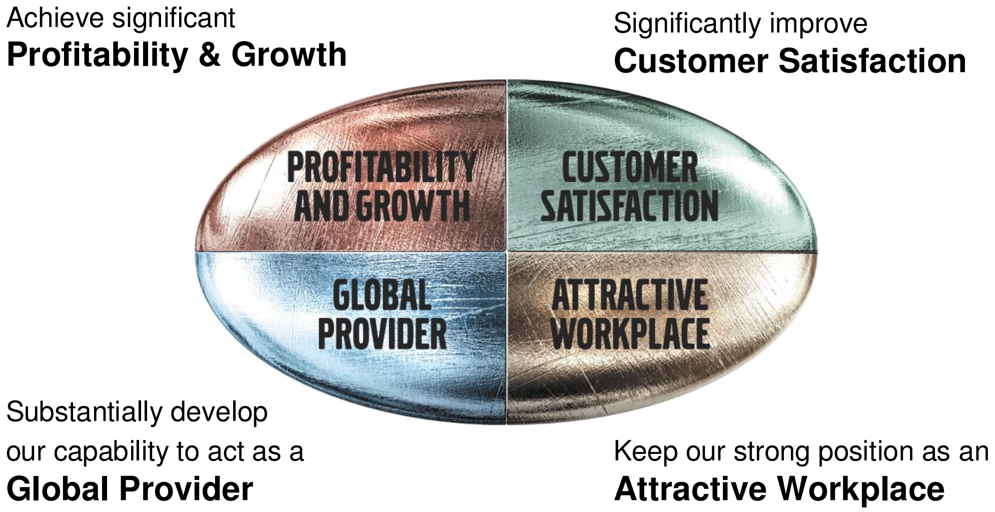


*Fig 3; Long term focus areas at Volvo IT* [144]

### *4.2.1 History*

The first computers at Volvo went into operation in 1961. Developments have since continued at a very fast pace.

- In **1967**, the Volvo Group gathered its IT operations together in a separate company for the first time: Volvo Data.
- In **1998**, the current global Volvo Information Technology (Volvo IT) was created.
- In 2001, the IT staff at Renault Trucks and Mack Trucks was integrated with Volvo IT. [145]

**Volvo Data becomes Volvo IT**

Before 1998 was Volvo IT all a part of The Volvo Group as one company and it had been so for a good 30 years. This fact is something that makes Volvo IT unique as an IT consultancy firm; its history as previous colleagues to what is their customers today. Many of the staff is the same but their relationship with their former colleagues is now that of being customer and supplier.

For good and for bad this means, among many things, that there is an understanding of each others' businesses that may be unusual in other customer – supplier relationships, and at the same time it may cause problems since there may be *expected* that there is an understanding for each others processes when there in fact is not. In addition, a friendly atmosphere can be good meaning; open-minded communication and sharing of information about problems, and bad; resulting in a lack of documentation and change

[143] Volvo intranet, Violin, 2006
[144] Volvo intranet, Violin, 2006
[145] Volvo annual report 2005, www.volvo.com

requests not being documented correctly. All of these opinions and experiences have been brought forth by the interviewees and survey respondents.[146]

## Volvo IT – a global company

The Volvo IT head office, with 3600+ employees, is situated on the Volvo Gothenburg site in Sweden and there are several other units in other parts of Europe, North & South America, Asia and Australia, all units combined employs more than 5600 employees including contractors[147].

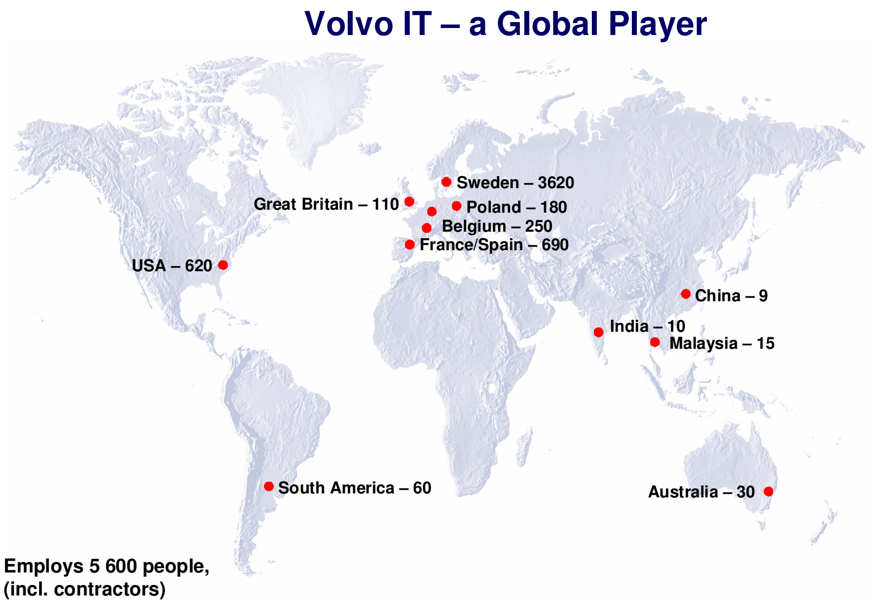


*Fig 4; Volvo IT - a global company* [148]

Volvo IT supports the Volvo Group organisation with IT support worldwide, the various areas that are supported include:

**Product development**
**Sales and market**
**After market**
**Manufacturing**
**Business administration**
**IT infrastructure**

These may include aftermarket solutions, business consulting, business intelligence, CRM solutions, dealer infrastructure solutions, IP telephony, manufacturing solutions, product development solutions and training solutions. All these areas emphasise even further the versatility of Volvo IT, and by that also, emphasise the complexity that the company deals with.

[146] See the Results Chapter
[147] Volvo intranet, Violin, 2006
[148] Volvo intranet, Violin, 2006

There is more IT to the Volvo Group than Volvo IT. The overall structuring function of *IT Governance* is directing the IT processes throughout the Volvo Group through directives and dictation of practices to follow for all companies in the company group[149].

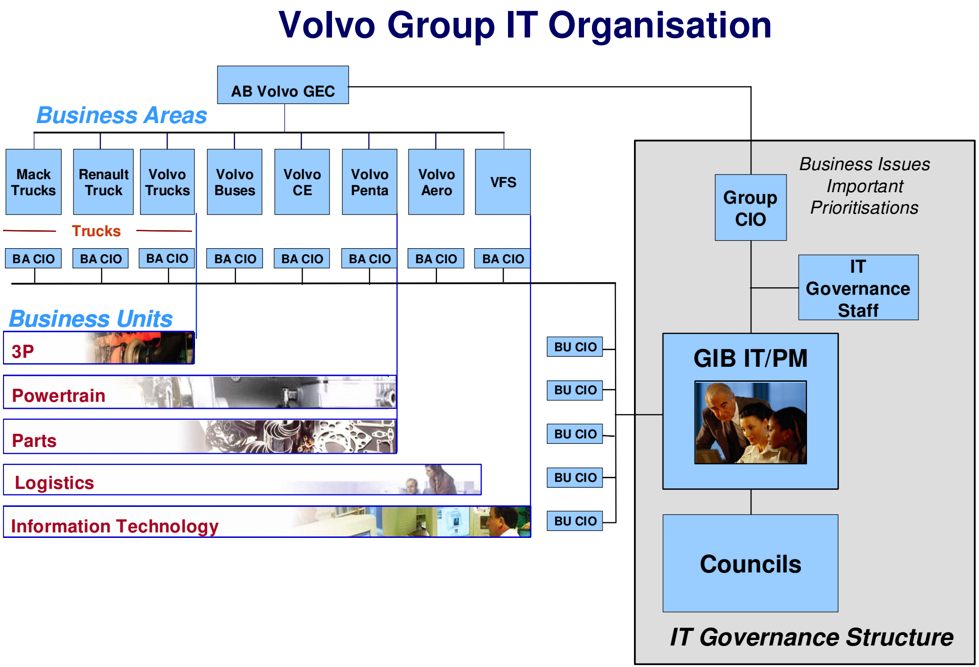


*Fig 5; Volvo Group IT organisation – more than Volvo IT*[150].

**Four main groups**

Volvo IT is divided into four main groups:

- Infrastructure & Organisation (I&O)
- **Industrial solutions**
- Application & Techniques
- Commercial

This study is mainly focused around the Industrial Solutions area with the Volvo IT supervisors working at the Purchasing Solutions and Project Control & Finance departments in the Industrial Solutions group.

### 4.2.2 Project models

There are several project models used at Volvo IT. The common feature for the models are that they consist of a certain number of "gates" that are viewed as closed at the beginning of a project, as a project approaches a gate there are specific conditions that has to be fulfilled for the project to be allowed to proceed forward. At each gate one or more meetings are held with representatives from both the customer and the supplier side, this will include steering committees from the IT side and the customer side, Project Manager from Volvo IT and Project Manager from the customer side for example.

[149] Volvo intranet, Violin, 2006
[150] Volvo intranet, Violin, 2006

The main model for IT-projects, which should be used throughout the Volvo Group, is called *Information System Global Development Process* (IS-GDP) and it is an extension of the Volvo Group project model *Global Development Process* (GDP) which is applied for product development projects. The IS-GDP model can be seen as a classic waterfall project model[151] where the preparation and pre-study phase (to the left in the image, see image below) are the beginning stages of a project where material is gathered to define the scope and list of requirements from the customer. As the project moves forward the stricter and more set the variables become. At the Final Development Contract Gate (the first narrow part in the model, see image below) all the necessary materiel needed for the project should have been gathered and from that point the project should be all about technical development. The IS-GDP waterfall model is the official project model for IT projects for both the customer in the Volvo Group and Volvo IT. Every IT project is run as two parallel projects in one; one project team on the supplier side developing the IT application, and one project running on the customer side addressing the change in the customer's business organisation that the IT application will affect.

The IS-GDP model is 'met' with Volvo IT's own *Project Control Model* (PCM) within the Volvo IT organisation. The PCM project model has gates that corresponds to the gates in IS-GDP but they are scheduled roughly a week in advance to the IS-GDP gates as check-points for the Volvo IT project team to make sure all conditions and tasks have been achieved in order to properly be prepared for the IS-GDP gate. The PCM model is similar to IS-GDP in its waterfall model characteristics in that it has less fixed variables in the beginning which later becomes 'set' as the project progresses and gates are passed.

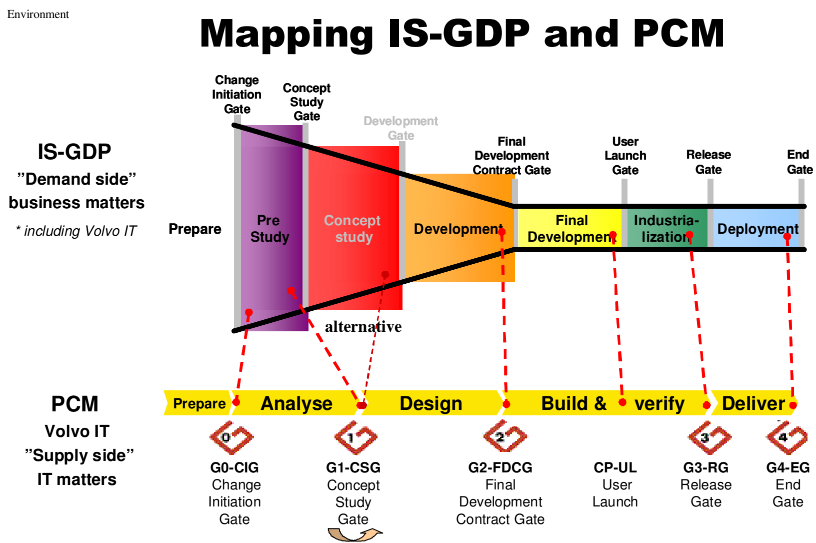
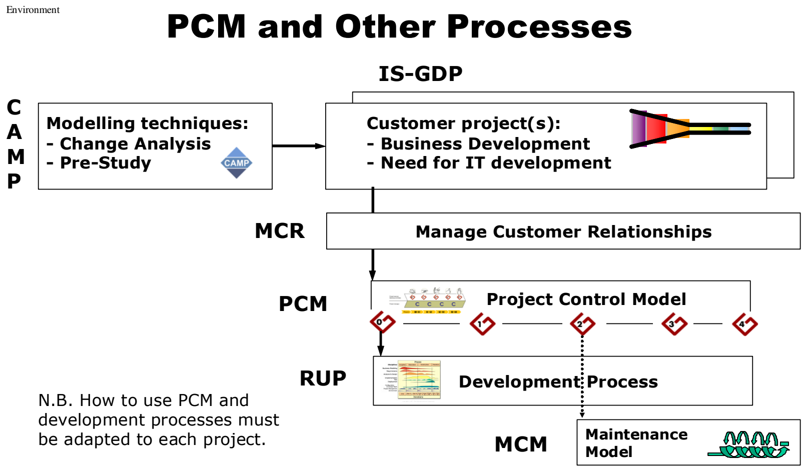


*Fig 6; How IS-GDP and PCM project models are mapped together*[152]

[151] The waterfall project model was developed during the 1960's as a response to the 'code-and-fix'-development at that time, it was an approach simple to explain by: "do the requirements, then design, and then implement", it has been argued that the article that most has been ascribed to the model's fame ("*Managing the Development of Large Software Systems*", Royce, W., 1970) was indeed misunderstood and the author's suggestion in reality was an evolutionary approach. The model has for the last 20-30 years been considered to be the 'traditional' project model for software development projects. Larman, C., 2005, *Agile and Iterative Development – A Manager's guide*, p.103-106

[152] Volvo intranet, Violin, 2006

To show the multitude of project models used at Volvo IT can project models be mentioned such as: *Change Analysis Methods for Pre-studies* (CAMP) – on how to perform the pre-study; *Rational Unified Process* (RUP) – for the development process alongside PCM; *Maintenance & Enhancement Control Model* (MCM) – on how to perform maintenance and follow up after a project has had its final release. These are only a selection of project models, there are several more used for various purposes and projects. Virtually any and every process has an accompanied project model to set a standard for the large organisation.



N.B. How to use PCM and development processes must be adapted to each project.

*Fig 7; Project models at Volvo IT - how they all connect* [153]

[153] Volvo intranet, Violin, 2006

# 5 Results – empirical study

*This chapter presents the results from the empirical study, the interviews and the results from the survey are grouped in to six problem areas.*

## *5.1 Results from the interviews and survey*

The material from the interviews and the survey will be presented in themes of problem areas. Although some problem areas/themes, may be more dominant than others may be there is no internal grading system;

**Being global** – about the problems a global company faces, i.e. in terms of project members being based in different parts of the world, the challenges in communication and collaborating efficiently.

**Organisational legacy –** every company, that is more than a couple of years old, has a history; *the way we have always done it*, which may clash with new directives of *this is how we do it* ***now***. Changes in work practice.

**Line Vs. Projects** – how to balance **the general responsibilities** of the department and working in **several projects** at the same time. The employees are viewed as resources and booked by percentage of time in projects.

**Role definition** – who should be doing what? About the confusion surrounding the different working roles in IT-projects.

**Work organisation**– the problem of having a lot to do and not enough time. What happens when someone have to start 'cut corners' and loose the big picture in all the stress when deadlines are too close and too many.

**IT-product Vs. Business Value** – is the focus right? The *product* or the *change* the product will bring to the customer? The focus of the supplier and the customer differ in a fundamental way.

The problem areas were categorised intuitively following the subjects and themes of problems that were brought up by several, or sometimes all, interviewees and what intuitively seemed to be causing most problems in the IT project processes. The names of the problem areas attempt to reflect the focus of a specific theme in this master thesis and are not related to any specific organisational function at Volvo IT.

### *5.1.1 Interviews*

The comments from the interviewees will be “coded” as stated below. The interviews are anonymous and only where it is relevant in the context will the interviewees’ working role be mentioned. The working roles represented among the interviewees are: *Project manager, Group manager, Business Analyst, Business consultant, Maintenance manager, Global process manager* and *Business (Customer) project manager*.

| | | | |
|---|---|---|---|
| Volvo IT: | **[V IT 1]** | Customers: | **[C 1]** |
| | **[V IT 2]** | | **[C 2]** |
| | **[V IT 3]** | | **[C 3]** |
| | **[V IT 4]** | | |
| | **[V IT 5]** | | |
| | **[V IT 6]** | | |
| | **[V IT 7]** | | |

### *5.1.2 The survey*

The survey was done anonymously and the respondents’ replies will be coded as *Volvo IT employee* or *customer*:

Volvo IT = **[V IT]**
Customer = **[C]**

## 5.2 Being global

*About the problems a global company faces, for example when project members are being based in different parts of the world.*

> *"… as an example from a project I worked on previously, we had project members in Lyon (France), Hagerstown (USA), Curitiba (Brazil), the Project manager in Brussels… this is the everyday reality in a global company today…"*
> **[C 1]**
>
> *"…there are projects that have run for 3 years and the project members have never met because they are based in France, U.S. and other countries …"*
> **[V IT 4]**

Globalisation is a reality for many companies today; Volvo IT has offices in several countries and accommodates Volvo Group customers worldwide. The majority of the software engineering projects at Volvo IT are distributed globally, including the software engineering development teams. This complicates collaboration in various forms regarding communication; between the team members within the project group when they may be based in different physical locations and are working together, as well as towards the customer when the customer also may be based in another/or several other country/-ies. Perhaps the obvious differences to think of are those in language and culture. The official language within the Volvo Group is English and according to some of the interviewees the skills and comfortable use of it varies with all employees in all countries, and that comment included self criticism as well, as one interviewee pointed out. One respondent of the survey says regarding on what it is that causes problems in IT projects; *"Lack of communication skills. Language is an issue when talking about global projects. "* **[V IT].** The language differences are however not mentioned as the most troublesome factor regarding being a global company. The problems seem more to be that of not being able to physically meet very often, or at all, face to face with other team members. This view is brought forth as one if the main reasons for causing misunderstandings in communication by both the interviewees and the respondents of the survey with no notable differences in opinion between Volvo IT or the customer side's respondents. In addition, that there should be company standards communicated to all sites on which project models to use and what work directives to follow are also mentioned as one problem, interviewees explaining that there are still differences in work practice between the sites, although the main idea is that they should all be the same throughout the Volvo Group. The sizes of the projects are also mentioned as a significant contributing factor to why the overview and big picture is lost.

**[Survey]** *"Communication in global projects should definitely be improved."* **[V IT]**

**Face to face**
When communications is not done face to face the alternatives are emails, telephone or the quite regularly used Net Meetings. In Net Meetings can for example a couple or

more conference rooms in two or more different countries be linked via the internet to view the same computer screen and have a speaker system in common in real time. Telephone conferences are similar to Net Meetings but without having a computer screen in common and everyone are linked up via the same telephone link. In addition, there are video conferences where it is possible to see the other persons' face on the video screen but these do not seem to be used very often at Volvo IT where this material was collected, none of the interviewees mentioned it as one of the communication forms that are used regularly. Nevertheless, video conferences can not truly be compared to physically being situated in the same room with the other person, nothing can really be compared to that - seems to be the general consensus among the interviewees.

Time zones may be an issue in all of these cases of communication since the communication is in real time though not in the same geographical location. One business analyst at Volvo IT spoke of conference calls after working hours at home in the evening only to fit in with office hours in USA. The interviewee did not think of it as too much of an intrusion of privacy since it was really preferred to speak directly in any circumstance - it would make the issues in the related work in the project easier to solve which made up for it - rather than emailing. However, telephone does, as mentioned, not show the other person's face.

> *"One important factor is that everything is more or less global today, which means more meetings over the telephone... The language barrier aside – which in itself is not always that serious in my opinion – it is so much more difficult to really understand someone if you can not see the other person's facial expressions as you speak to one another…"* **[V IT 4]**

Everyone in the interviews seem to agree and be very aware that this is a problem - that project groups are not located geographically together and that this brings forth serious communication problems, perhaps even more so when the language of communication is not every team member's mother tongue.

**Teambuilding**
One group manager mentions the importance of having some form of team building every time new people are brought in to a project and that this is the project managers job to organise, but concludes that also the whole idea of teambuilding is dependent on the project group being a working group that is based geographically and physically together. *"..Otherwise does the whole idea of teambuilding fall apart..."* **[V IT 6]** Most often is this something the working role of the project manager has no influence over; it is instead on a company organisational level that the projects are decidedly global and very large. However, on smaller projects teambuilding activities are viable and important to emphasise says the interviewee.

The point of organising teambuilding for a project group is to bring the project group together in the beginning, allow everyone to get to know each other better, allow everyone to find their roles comfortably, and to later on discuss any problems or arguments that have occurred and by doing this improve communication and team feeling. Problems later on in the project can be avoided by doing this and the sheer quality of the work may also improve when everyone are able to communicate better as one interviewee points out, another interviewee says;

> *"…there is sometimes talk about different personalities that are needed to complete a project group but I have never seen it work like that in reality, you have to work with whoever is available at the moment…teambuilding is something to work on and develop, everyone take on a certain role and it is important to come up with something to do together in order to get to know each other so we can work better together"* **[V IT 3]**

Without any possibility to organise teambuilding activities in a project with team members located around the world the communication and team feeling will not surprisingly be effected.

> *"The idea is to have a chance to meet up, discuss any problems we have had, see what we have achieved so far, try to wrap it up with doing something fun together, maybe eating a nice dinner and socialise with each other in a relaxed way… not everyone thinks this is important when you are pressed for time in a project but that is often when it is needed the most, you are sure to make up for all that time later on in the project when problems are dealt with easier and quicker for example if you know the people you are working with…"* **[V IT 6]**

**Organisational culture differ between Volvo sites**
For every large organisation appears the problem of having global standards or local versions, or both, of project models a balancing act. For global projects there need to be global working standards and policies, but local variation may appear even though they may not be evident right away, according to one interviewee. This can be viewed as a managerial question where global directives should be agreed upon and be followed up in order to facilitate global projects and collaborations. However, as one customer project manager also points out; it is also dependent on the individual.

> *"We use the IS-GDP project model but it is new, not everyone does use it, it varies both between Volvo sites and countries… some are a little more reluctant to change the way they work. If everyone has reached different levels of how to use the model - that makes it so much more difficult to work with. For example, France has not come as far as USA in implementing and using IS-GDP which makes it easier to work with USA…In many cases it is foremost dependent on the single individual's choice though…"* **[C 3]**

**[Survey]** This is mentioned in the survey in some places, for example one respondent points out these local differences as one factor to what causes problems in IT projects.

> *"Some companies have local systems and local processes. The wish to work globally must take into account these specificities "* **[C]**

One respondent in Mexico also points out that Volvo IT is not present in Mexico and their site have not yet access to several project planning tools and software which are

commonly used across the globe in IT projects at the Volvo Group. The respondent writes that that problem should be solved by 2008.
One respondent mentions the cost factor regarding global/local differences that may arise when they are overlooked.

> *"When talking about global projects, the specific needs for small sites are often overlooked. Solutions are designed for the larger sites and many times are those inapplicable/not manageable for small sites. This is often seen too late and causes considerable costs."* **[V IT]**

**[Survey]** On the question **"***What can be done TODAY to avoid or solve some of these problems?"*- are the differences between the Volvo sites brought up:

> *"Many things. Example: Formal IS-GDP training on every site"*

Any large company is likely to face similar problems with co-ordinating various sites in various countries, this is likely not an exclusive problem for Volvo IT.

**Extremely large projects**
For someone coming in from the outside it may at first be hard to comprehend the scale of some "projects" at Volvo IT. There may be software development projects running for 5-10 years, involving all important and all very large global Volvo companies as Volvo Parts, Powertrain, Volvo Penta, Volvo buses and more; all important and powerful global players. The software may effect functions in the production line on all various sites in various countries and representatives from all companies and all sites are involved, alongside Volvo IT employees –also- from all over the world. Needless to say, some projects are very 'large' and indeed worldwide and they may affect several different levels within the various complex organisations in several aspects, and every project is sure to have multiple interdependencies to other projects throughout all involved organisations, that may come to affect the project planning and execution. It is not possible for any human to keep an overview and see the big picture and to comprehend how all pieces and projects are linked together nor to be able to foresee how changes can come to have knock-on effects on other projects. To take it to a philosophical level; we humans build things bigger than we ourselves can comprehend. How do we deal with the complexity of magnitude? Some interviewees say the problem is that the very large projects need to be broken down into smaller part-projects, small enough to allow an overview, that there really need to be a view over "the big picture" in order for correct decisions and prioritise to be made.

> *"The more complexity a project has - the more difficult it is to operate and that should be avoided, as much as possible. Global projects may not be a problem in itself if they can be broken down in to smaller projects…"* **[V IT 6]**

Some interviewees mention that to make the projects even possible to comprehend they should be made smaller in time and scale. Instead of adding and adding to one large project; create smaller projects that are more flexible and manageable.

**[Survey]** What can be done today to avoid or solve some of these problems was one question in order to find out if the respondents of the survey had ideas of their own about how to address these problems directly, today.

> *"Avoid giant projects, or break very large projects into much smaller, manageable releases. Have weekly meetings with customers where progress is shown and demonstrated. Discuss all potential and realized project risks the minute they are identified. "* **[V IT]**

**[Survey]** On the question on what or who causes the problems summarises one respondent several problem themes, of which being global and not being coordinated at the same time, is one.

> *"No formal methodologies and policies. Unclear or not existing processes prior to automation attempts. Global systems using old technologies, not flexible, not integrated, expensive, slow response times, and scarce resources scheduled for other priorities"* **[V IT]**

### Conclusions on Being global

Perhaps somewhat surprising is it not the cultural differences and language issues that are stated as the most complicating factors in global projects but more so the problems in communication caused from people not being able to meet and see each other face to face. In addition, having different directives to follow on different sites may be a cause for problems. The size of the projects is another important factor. An incredibly large project is difficult to operate and offers no real overview which complicates planning and risk management.

## *5.3 Organisational legacy*

*Every company has a history of work practice; the way we have always done it, which may clash with new directives of this is how we do it* ***now****.*

Every company that is more than a couple of years old has a history. The history can be of a more technical nature, for example with a dependence on old systems and old code that has not been documented as is done today, as well as of a more “softer” nature with a change in the work practice. Although the problem with a technical history is very “real” at Volvo IT perhaps more important is the change of work practice when Volvo Data went from being an IT-service department within the Volvo Corporation into being a separate company on its own called Volvo Information Technology[154]. This meant that; the staff remained the same, the offices remained the same, but the official relationship had changed between the former colleagues; they were now in a customer-supplier relation with each other. That which used to be one company now became separate companies, independent of each other, though closely related. For example did this significantly change the demands on documentation in projects with the new customer-supplier situation. It became more important to document all changes and new requests and to be able to do estimates on time and cost before change requests were implemented, all to another level than it had to have been done before. It should no longer be possible for a customer - who used to be a colleague within the same company - to pick up the phone and ask for an alteration in a system, it would need a documented formal decision for that from now on, an interviewee explains. Questions had to be formally answered such as; is the alteration the right one, is it a priority, does the customer want to pay what it will cost in the end, etc. Even though this change took effect several years ago it is important to point out that many employees within the Volvo Group have worked within the same, or various Volvo companies, for many *more* years. The history lives on in many ways on all levels within the Volvo Group and this is not surprisingly evident in the factual work practices.

**Assuming that everything is understood**

There can be a great advantage in knowing the customer well, to be ‘old friends’ with the customer; it may be possible to know what the customer needs even better than the customer itself does. Similarly for the customer; it can be of great help in knowing the software supplier well by being familiar with each other’s processes and ways of communication. At the same time it poses a risk to *assume* that someone knows the other part better than they perhaps actually do. The mutual opinion may be that both are in agreement with the customer/supplier but that may result in a lack of detailed documentation of the demands on the system and fundamental criteria that needs to be fulfilled in order for the system to work efficiently.

[154] See chapter on Volvo IT history

> *"…the customer is counting on us to have a full insight and understanding of their organisation and their processes, we often expect the customer to have a good understanding of how we work…there is a lack of competence on both sides…we don't know as much as we think we do…"* **[V IT 4]**

This is a problem in all customer–supplier situations; to understand each other and to understand that there may not be an understanding of each other as well as it is assumed on both sides. This is not exclusive to Volvo IT and their customers but the long and mutual history they share with their customers today is somewhat unique and cause problems, as well as being an advantage, in negotiating requests with customers.

**Scope creep**

One result of not having discussed in detail what it is the project should result in is the troublesome situation of having a customer say later on that functions are missing that were included in the scope from the beginning, and the supplier side saying it was not, an interviewee explains. It can turn into a tug-of-war where the customer claims functions are missing and the IT side is saying it was not in the scope, and the customer saying that of course it is in the scope – the system could not work without it, and the IT side saying that it was not specified from the beginning and so on, one interviewee saying adamantly that this is linked to the history of working with former colleagues and assuming that everyone knows what the other one is talking about. This is also linked to the theme *"IT-product vs. Business value"* [155] where the different view-points of the customer and the supplier are discussed.

**Unwanted customer service**

**[Survey]** On the same subject as above but in "reverse"; instead of the *customer* assuming that Volvo IT would automatically 'know' their need, *Volvo IT* can assume that they *do* know and add functions that they assume that the customer wants. The case of the supplier assuming they know what the customer wants and believe they are doing the customer a bigger service, but the customer unfortunately does not agree and accuses the supplier of taking longer than necessary to finish a product. One interviewee (from the customer side) told a story on this and said that this did indeed happen and concluded that neither of the scenarios is cost effective and can cause delays to the project and cause frustration. This subject was therefore included in the survey. In the survey the respondents were asked to answer the statement: *"Volvo IT may add functions in systems without informing the customer first"*. - Given a five grade scale ranging from *I completely agree, Average* to *I completely disagree*[156]. Looking at the results there is a clear emphasis towards the opinion that this is not a common problem, although as many as 7 respondents, out of 29, which translate to ca 25%, do *"agree"* or *"completely agree"* with the statement. It should be preferred not to have anyone agreeing at all on this question.

[155] See IT-product vs. Business Value chapter

[156] For more information read the full-length survey in the Appendix section

### Professionalism between former colleagues

The confusion on whether we are *old colleagues and friends*, or *customer – supplier* does sometimes result in a feeling of unprofessional attitudes, several interviewees mention this in various ways.

> *"There can be a lack of professional correctness when it is Volvo IT's old friends that call up and place an order, external consultants can be much more specific when it comes down to what it is that shall be done and be specified…"* **[V IT 4]**

The need to be, and act, professional may perhaps be more important and emphasised when someone is dealing with an external player who they do not expect to know anything about the organisation, or whose help and business they are depending on.

> *"Volvo Data becomes Volvo IT…there are demands now on being more professional, that is lacking in some cases, the old way of thinking and acting is still in practice. There is a remarkable difference in dealing with … [external consultancy company of software engineering] and my own company – they will take care of everything and organise all practical aspects, but try calling someone in my own company and ask for help and you are more likely to be met by an unhelpful arrogance …"* **[V IT 3]**

### Professionalism

**[Survey]** "*The communication between Volvo IT and the customer is always professional*" was a statement with multiple choice answers since interviewees had pointed out  this problem of 'everyone being friends and suddenly there is a customer – supplier relationship since Volvo changed from being different departments into different companies'. The view that this may bring about a friendlier and sometimes less professional attitudes was brought up by several interviewees and also that it may result in documentation of new requests not being done properly. The respondents were given a five grade scale ranging from *I completely agree, Average* to *I completely disagree*[157]. There is a leaning towards "*I completely agree*" and "*Average*" here with a total of 27 respondents, ca 93%, ranging from average to completely agreeing (7 respondents completely agreed the communication was professional); saying the communication is indeed professional but notably two respondents on the customer side completely disagreed.

### Internal administrative functions

The internal administrative processes seem not to have been adjusted to suit the needs of a modern consultancy company, instead of a large 'service' department one interviewee concludes. This may be an effect of a rapidly growing company with a history of not having to accommodate its customers as external professional clients. One project manager talks of the problems of organising an office chair for an external consultant in a project. Initially, it was complicated to find out how to order a chair and from which department within the whole organisation, on to finding out that it would take as long as a couple of weeks to get the chair after placing an order for it. Similarly did it demand a

[157] For more information see the survey in the Appendix section

certain amount of detective work to find out daily costs from another department in order to make cost estimations for a project, the interviewee explains that the reluctant reply in the end was that it would take three months to receive those figures from the relevant in-house department within the same company. All additional time a project manager need to spend on lengthy additional administrative work is a cost for the project and eventually for the customer, the interviewee explains, and says that this is an issue that is not really highlighted and understood by everyone.

### Technical history

Technically, for a large company such as Volvo, are there plentiful large systems that have been developed, altered, added on, and changed over time and may have "lived" through several changes of work practices. Documentation and follow up's may not always have been done by the standards that are dictated today and the attitude over the years have been to accommodate the costumers every request and say "yes" rather than "no" to new requests to meet the customer's needs. This has in some cases resulted in an overgrown system and today investigative work may be needed to determine the effect new code and changes may bring, an interviewee explains after having worked with maintenance. There may be code that is no longer in use but can nevertheless surprise and affect changes that are made in one end today – it can have an effect in a completely different place that may not logically be linked. This is understandably something that may cause unforeseen problems but at the same time it is a hard problem to address effectively, cleaning up the whole system is not a feasible task because of its sheer size.

Today the processes to handle new requests from customers are defined, how to prioritise and how to document it, but they have not always had an effect on the day to day routines, it takes time for changes to set in.

> *"…One problem about having to work with a system that is quite old and the philosophy has always been to accommodate the customer as far as ever possible - makes for a quite "wild and overgrown" system… a very difficult to keep clean… there can be old code lying around in there which is not in use, you loose the possibility to keep an overview and to see beforehand what the results of your changes may be… this can create a lot more work later on when you may need to investigate why certain problems appears in unexpected places – that kind of work is completely unnecessary if you know what it is you are doing when you change or add something to a system that is structured from the start…"* **[V IT 5]**

### Conclusions on Organisational legacy

The history of having been colleagues with the customer may be helpful as well as being a cause for problems. An insight and understanding of the customers' business processes may be an advantage but can prove a problem when accompanied with an overly optimistic notion that all have agreed on something, without making absolute sure that everyone has the same idea of what it is that shall be done. We think we know, but do we really? Going from being colleagues to being customer-supplier changes the fundamental demands on collaboration regarding documentation and attitudes, a lack of professionalism may be one result of not having the roles and new work practices laid down into every day to day factual practice's within the organisation. Additionally a technical history which is the result of a philosophy of complying with the costumers every request through the years can be causing problems when the work practice changes and new technique is applied, it can also cause frustration when changes may take longer to be implemented then previously made requests.

## 5.4 Line vs. Projects

*On problems about balancing the responsibilities of the own department and working in several different projects at the same time.*

Every Volvo Group Company is organised into business units, every business unit is organised into departments, every department is organised into groups. Every employee at Volvo IT is working in a group and has day-to-day responsibilities towards that group and group manager. In addition to that does several projects run across groups and departments and countries which employees are involved in as IT project members. Contrary to the work in some external consulting firms that book a *person* for a project are employees at Volvo IT booked as resources by *percentage of time*. There is no limit to how many different projects one employee can be involved in at the same time.

> *"I am involved in three different projects at the moment but there is no higher limit to how many there can be... Efficiently it shouldn't be more than 2-3 small projects at the same time, if it is larger projects the best would be to only work in one project at the time..."* **[V IT 4]**

One weeks' work is 100%. To be booked on one project for 50 percent means that an employee should spend half of his or her working week working on that project, variations exist but i.e. 50 percent equals roughly 20 hours of a working week. No one can spend 100 percent of their time on one project or assignment since there will always be additional duties towards the own group and/or department during a working week.

> "…everyone has other, additional job duties already so no one can dedicate 100 percent of their time on one assignment, which is a difference compared to external consulting firms where you can hire someone full time to focus on one problem..." **[V IT 7]**

Volvo IT sells *complete solutions* to its customers including new development and continuous maintenance. This is also one fundamental reason to why it is not possible to book one person for one project to 100 percent, it is in some cases the same employee whose responsibility it is to perform maintenance on one system who is also booked for taking active part in other separate projects. When discussing this issue regarding projects a couple of interviewees point out the advantage of having people from maintenance involved in a project from the start throughout the project's various life cycle, to have that knowledge throughout the project, though this would unavoidably mean taking time from other maintenance work.

Projects can run for several years and be dependent on, and effected by, many different complex factors and it is not efficient to book one resource on only one project for several years in a company that performs multi level services on its customers systems. It is also an advantage to have the insight and knowledge of maintenance people being brought in to new development projects and combine that with the thinking of project people. Nevertheless this is brought up as a problem with the allocation of resources when it is combined with a shortage in resources and a high changing-and development rate in the company; who/what is prioritised, who/which project is not, and who can have time to *do it all* when resources are scarce. One maintenance manager explains:

> *"…The maintenance team should support maintenance, perform maintenance work, do enhancement work on systems, work as general support on various issues sometimes AND have time to be members in various IT-projects for new development… this means that the workload grows faster than we can work it off…"* **[V IT 5]**

**Management level - Prioritising the work and various projects**
The human factor plays an important role in the allocation of resources; it is imperative to receive attention from others both within the project team, from outside the own team and from higher management. One interviewee says that it would be best not to even take on a project unless it has the attention from management on a high level because those projects are the only ones with a good chance of being given appropriate resources and not become ignored later on.

> *"… It is important to have the attention from everyone else regarding a project, to have more people think it is important and interesting, it is important to have the customer think it is important as well as your own colleagues so that no one starts to book your resources on other assignments…"* **[V IT 3]**

It is smaller assignments and projects that loose out when priorities have to be made, one interviewee explains, those projects can in turn have a negative effect on other projects that depend on their time plan and so forth. It becomes a domino effect in the matrix organisation and all interdependencies are difficult or humanly impossible, to keep an overview over and understand, it is all interconnected from organisational to operational level.

**Operational level - Who/What decides on who is doing what?**
The project manager has a responsibility towards a project and its execution but has no power over the allocation of resources. The project manager hands in requests for resources to the project's steering committee and the steering committee decide on whether to approve of the project manager's requests and additionally make sure that these resources are available for the project. However, the result may be that the project manager is approved fewer resources than requested and that the approved resources may also not be readily available, or become allocated away from the project to focus on other assignments later on. An effect of this may come as a surprise when new priorities have been made during a week outside of the project manager's knowledge. *"One week I had a resource booked for 60 percent and when the week had passed I got 2 hours work reported from it"* as one project manager recalls. As some interviewees points out; in the end it is the group manager who pays everyone's salary and as such will the group manager always have everyone's attention when it comes down to it, and that may mean that projects will be put on hold when activities for its members collide with other responsibilities. The group manager has an area responsibility and is not responsible for

any single project themselves, but may be a member in several steering committees for several projects involving employees in the own area group.

It is firstly on a managerial level in the steering committee that the priorities are made on how resources are allocated and secondly with the individual "resource". The individual performs a balancing act between several different assignment and projects and may be facing contradictory directives.

> *"…perhaps it does not have to be about a new system or a new project model, it could be something else, perhaps just a clear managerial decision on that it should not be possible to book a resource for more than 2 projects, if I request an already booked resource then I should receive a "No" right away… it is frustrating amongst project managers to be given resources of 10 percent, the small percentages are always overlooked by other assignments or activities anyway when it comes down to it, say 50 percent or nothing at all, that would be a great improvement…"* **[V IT 2]**

It is difficult for a resource to split a working week into 10, 20, 30 percent and make that work efficient, explains one project manager, the personal priorities play a big role in that and many factors can influence it outside the employee's control or power.

**A focus on project management**
**[Survey]** On the question on what or who causes the problems in IT-projects at Volvo IT does one respondent in the survey talk of the problem with management not realising or emphasising the importance of project management, also project managers not being able to concentrate solely on project management because of line responsibilities.

> *"Executive/Upper management does not take Project Management seriously enough and assigning inappropriate individuals to the role as PM. Furthermore, Project Manager's often have other line responsibilities which do not allow them to focus on Project Management."* **[V IT]**

On the same question does a respondent from the customer side also conclude that there are problems that arise from having project members involved in several projects at the same time.

> *"People who are working on several projects are not always well coordinated in terms of workload per period."* **[C]**

Another respondent speak of the impossible task of managing to 'do it all' with both project and line responsibilities and being able to deliver 100 percent to all ends when asked if there is a phenomenon of some sort that causes problems.

> *"Not having enough people dedicated to the project; the people that are assigned to work on projects are still working their normal day to day job and still devote 100% to the project. The workload is overwhelming. "* **[C]**

### Conclusions on Line vs. Projects

With a working week split into percentages of time to spend on different projects and assignments is one result that it may only be the project manager who is able to focus and give full attention on one single project, and sometimes is not even he or she “free” to do that. The project manager has additionally no power over the allocation of resources in a project, that is the steering committee and group manager’s responsibility and they have in turn several other projects to decide on, as well as to take responsibility over their own area or unit. There may be conflicts of interest. Priorities are always made that affect the factual work of resources both on a managerial level and on individual operational level. This will not surprisingly effect the working situation in a project with regards to the dedication to it, the time plan and eventually costs.

## 5.5 Role definition

*Who should be doing what - about the confusion around the different working roles in IT projects and the problems this may cause.*

> *"…it is a serious problem if people in our projects do not do what they are supposed to do, this goes for both the customer side as well as the IT side… it is a great risk with having people on that do not know what is expected of them or that do not know how to do it… it can lead to a lack of faith in the project members - and then we have serious problems…"* **[V IT 1]**

The different working roles in a project are explained in the literature on the project models at Volvo IT. There are many different roles and it is not always easy to find a straight answer to questions about who is responsible for a specific function. The roles vary slightly in its assignments from project to project as well, since no two projects are exactly the same with the same group of people involved, interviewees explains, but there is no official document included in the project models which should be updated regarding every project member's role in each individual project.

> *"…**Who is the decision maker in a project?***
> *Well that is not always clear… you have a sponsor on the IT side and an IT steering committee, a project manager… on the business side there's a similar sponsor, and a business steering committee, a business project manager…" (the interviewee begin writing an overview on the white board)* **[C 2]**

Interviewees do indeed speak of the confusion surrounding the different working roles in projects. The roles are not always communicated to everyone when a new project starts, alas, there may be misunderstandings when what the project member *think* is expected of him or her is not equivalent to what other project members assume.

**[Survey]** After the statement "*The working roles are well understood in projects, everyone knows what to do, on the customer side as well as the IT side*" the respondents were asked to indicate on a five-grade scale ranging from *I completely agree, Average* to *I completely disagree*[158] what they felt was most appropriate according to their experiences in IT-projects. Most respondents chose the "*Average*" column but as many as 8 disagreed and 3 completely disagreed out of 29 respondents. That means that as many as almost 40 percent of the respondents say that the working roles are not well understood in IT-projects.

**Managing uncertainty**

> *"The roles need to be specified for each project… every role has to feel a responsibility for their working role and the assignments of that role, everyone need to understand what their role means, for example the business analysts*

[158] For more information read the full-length survey in the Appendix section

> *must feel a responsibility that what they talk to the customer about is feasible to execute in reality, they must understand that it matters..."* **[V IT 7]**

Confusion on who owns a function may also lead to functions being overlooked and forgotten about if it is not viewed as important enough to delegate at an early stage in a project. Such functions can for example be of how to deal with complaints in a project that is not running well. In the beginning of a project are everyone's intentions usually that everyone get along really well, one interviewee says, it may be overlooked that it is important to discuss how the project group may deal with potential problems that may arise if no working role is delegated the responsibility to deal with those issues and bring it up on the agenda.

> *"…there is not really a correct way to deal with complaints…a project manager may turn to his or her group manager, that manager may take it to the steering committee for the project, but there is no official right way to go…"* **[V IT 2]**

**Role definitions may affect the formation of a project group**
The potential confusion surrounding working roles in projects may also affect the appearance of the project organisation as one interviewee points out. The planning of the work in a project as well as the formation of members in the project group may be affected should there be confusion on which working role is doing what. The result can be a project group with too many employees being booked up, or more likely a project group that is under-dimensioned because not all working functions have been clearly specified from the beginning.

> *"You HAVE TO understand what is expected from you and that needs to be communicated out to everyone…discuss the various roles and rules for each project, often it is far too generally explained about the roles - it needs to be specific in details and understood what it means for this specific project – this could even effect the look of a project, effect the number of people that are needed for it and so on..."* **[V IT 4]**

**No official project document on role definitions**
There is today no specified function or document that follows a unique project where all working roles and responsibilities are detailed. The problems that follow this is a certain confusion on who is supposed to do what, confusion on who owns a function in a project, who is making sure it gets done properly and who feels a responsibility to make it right.

> *"…I was booked for a project where I was responsible for a specific gate, I did however realise that the customer thought that my role was that of a project manager, quite a serious misunderstanding… It is not clear about which working role is supposed to do what and it is not communicated to everyone, there is always the time issue - everything is rushed and most often do people just assume that they know what to do… There are different opinions about the correct assignments for each working role and the interpretations differ in general, the way it works in reality is not always how the literature on the project models explains it…"* **[V IT 4]**

Interviewees talk about the frustration of sometimes not finding someone responsible for a function in the company and the same goes for individual project organisations.

**Having the right people in the right positions**
Interviewees speak of the frustration of people "ending up" in positions that are not suited for them. This can be the case of having the wrong person in the wrong place but it can also be linked with the fact that an employee does not know what is expected from him or her. Linked to this may be the problem of people not 'stepping up to their plate' as well as people 'stepping outside of their own plate'.

The problems surrounding the confusion on roles in projects involve managerial level as well as the operational level in projects. The equally important roles of the steering committee seem to not always be understood by everyone. Some of the interviewees pointed this out as an issue for concern that also the steering committee members need to understand and respect their own importance in a project.

> *"…some people do not seem to think that they have to take part in a whole project, instead they can step up when it suits them and start making demands and change decisions, this can happen in steering committees, I do not know why they believe that that is appropriate behaviour for them, if they were not informed about the project beforehand or what the reason is? …but it does create a lot of problem when someone suddenly turns up and wants to change everything that has already been decided on…"* **[V IT 4]**

This is also linked to *Line vs. Project theme*[159] because the steering committee members with decision power in the projects may be in 10-20 different steering committees at the same time, as well as having other responsibilities to other departments, areas and more.

**[Survey]** On the question on "*Why do you think these problems in IT-projects occur?"* does one respondent in the survey mention the problem of not having the right people on the right positions which may be linked to the problem of not everyone understanding what a certain role involves which leads to 'wrong' employees being appointed certain roles. It is also likely to be referring to the fact that there are no formal resources as to how the communication with the customer should be performed.

> *"No dedicated resources for communicating and understanding customer needs. Many times, resources available are too technical and lack communications skills"* **[V IT]**
>
> *"Stick to a time plan and have the right people on in the project from the beginning."* **[C]**

**[Survey]** On the question *"How can the communication between the customer and Volvo IT be improved?"* do one respondent suggest that there should be project managers who are only working as project managers and nothing else, because they

[159] See chapter on theme Line v. Project

have the necessary skills for it instead of having other responsibilities and occasionally be appointed to the role of project managers.

> *"Use Project Managers whose sole responsibility is Project Management, we too often put technical leads in the position of project managers who generally lack the proper communication skills to be successful. "* **[V IT]**

**[Survey]** On the question "*What can be done TODAY to avoid or solve some of these problems??"* is this opinion expressed again by a respondent from the Volvo IT side.

> *"Have a knowledgeable Project Manager function strictly as project managers in order to better develop and utilize their skills. Although it is a currently over used buzz-term, a global PMO (*Project Manager Organisation*) could be a good option. "* **[V IT]**

**The list of requirements – whose responsibility is it?**
Out of all interviewees was only one adamantly sure about whose responsibility the list of requirements was in the beginning of a project. The list of requirements were however brought up by many as something that needed to be improved, but the opinion on whose responsibility it actually was to compose it varied. Many agree that an improvement is needed but no one really knows whose job it is to do it; a telling example of the confusion surrounding the roles. The correct answer according to the one who claimed to be sure about it said it is the business analyst's role that should compose the list of requirements, together with the customer and with an IT-architect, if there is an architect in the project.

**Comparison with product development projects**
Something that makes IT-projects fundamentally different from product development projects is the fact that it is not possible to be entirely sure about what the end result will be in the end. The uncertainty around it all is much greater; IT systems can be very varied. As one interviewee compared it; if someone is going to produce a car - they will know what they will end up with in the end and they know what parts they have to have in order for it to work, and without it they would not even consider starting a project. With IT there are no straight answers in the same way but some interviewees say that the software project organisations and the product development organisations perhaps should be compared with each other to a greater extent. Perhaps IT-projects could learn more from the much more established product development project organisation, was one interviewee's idea.

> *"Perhaps it would be possible to look at product development projects to another extent - they have a more established project organisation… it is nothing new about it in the same way as software engineering projects has a shorter history, after all it is more or less the same components… programmers = construction workers, someone to overlook it all = project manager..."*
> **[V IT 7]**

### Conclusions on Role definition

The negative aspects of having a project group where the working roles are not clearly defined and communicated to everyone are many; for example that there may be functions in a project that no one feels responsible for, that are overlooked or perhaps even done twice in unnecessary double work. It may affect both the time plan and the costs in the end. Misunderstandings if two people believe they are to do the same job can be unnecessarily time consuming and costly for the customer in the end. The working roles in a project need to be clearly defined, explained and continuously updated.

> *"…**What can be done to eliminate this confusion?***
> *Specify the assignments for each working role! And keep updating that information…"* **[V IT 3]**

## 5.6 Work organisation

*On the problems created from having "too much to do" and deadlines that are set too early to be able to have all preparatory work done properly.*

When discussing problems in IT-project and what causes them is a frequent comment that there is just not enough time to do everything right, that everyone has an overwhelming workload that prevents them from performing at 100 percent all the time. In the interviews there are many different views expressed on what the work 'overload' results in. It does however seem to be the general opinion of both interviewees and survey respondents that the overwhelming workload does cause problems.

> *"…Usually there is just not a time frame that allows for reflection on whether even the fundamental conditions are fulfilled for a project to be successful… often a project is given the 'Go ahead' to start before someone has had the possibility to see if it is even possible to succeed…"* **[V IT 2]**

**[Survey]** On the question on "*What causes these problems?*":

> *"Customer's workload is too big, Volvo IT Project Manager, team member's work load too big "* **[V IT]**

> *"Often there are not enough resources available to do the work, which pushes back the project scope and in turn makes the customer unhappy. "* **[V IT]**

**The pre-study**

The pre-study is the gathered research that is done before a project starts and this is where all specifications for the project and its prerequisites and requirements are specified. The pre-study is the first part of the project model and it sets the basis on how the remaining project processes should be developed and executed. The 'infamous' pre-study for a project is commented as a frequent source of problems in IT projects by the interviewees. Everyone seems to agree that this is a problem that should be addressed and that the pre-study needs to be put into focus and be allowed more time in the project process since mistakes made in an early stage inevitably means greater problems later on. The main reason to why the importance of the pre-study is overlooked is part because *the understanding of its importance* is lacking and also part because the lack of that understanding the appropriate time-frame for it to be produced in is underestimated as a result. It is not allowed to take the necessary time because everything has to be done quickly and is rushed.

> *"…**How come the pre-study is not given sufficient time to be properly defined?** Everything is always rushed! It is really important to communicate well with the customer in the beginning, when or if, the project is about to start and allow time for the pre-study, instead of having to 'put out fires' later on…"* **[V IT 2]**

> *"…Many people feel that it is more important to get going in a project then to have everything clear and defined first…it is never good if something requires more time- and that is a big problem… If there is a lot that has to be done and it is supposed be done with significant speed, a lot of resources are needed in order for it to work, … you can not expect it to work unless the function is re-evaluated with greater importance… At Volvo IT we need to be better at informing the customer that the pre-study have to take more time, be more extensive… this should be an important signal to communicate to the customer "* **[V IT 7]**

**[Survey]** The respondents were asked to indicate whether he or she agreed or disagreed with a statement and one statement was: **"**The initial scope of projects is detailed and well defined". Only three respondents from the customer side *"agree"* or *"completely agree"* that the scope is detailed and well defined out of 29 respondents. All other respondents' answers ranged from *"average"* to *"completely disagree"*. This means that almost 90% of the respondents disagreed in some form.

**[Survey]** After the question of what causes MOST problems in IT projects are the respondents of the survey asked: *"What can be done today to avoid or solve some of these problems?"*

> *"Less focus in the development phase in projects, and more focus on the pre-study. What do we need to analyze before we can start to develop. "* **[V IT]**

**Keep up the structure**
If there is not enough time to organise the workload and to keep some form of individual control and overview of the individual workload, this will be a contributing factor to worsen the situation an overwhelming workload will bring.

> *"…so much has a tendency to lack some form of structure… People have too much to do to have the time to be structured in their work – that is the reason to why people become stressed and burnt out – because of all uncertainty that is surrounding everything when it is all very complex and crazy to a greater extent, the uncertainty is what gets to people…"* **[V IT 4]**

**The intense changing pace**
With an intense workload there is the increased possibility of the human factor that may cause problems knowingly, but not willingly. With the pressure of getting a lot of work done in a short time-frame, it may be the human psychological factor that influences an employee to 'want to' get more done than the situation actually allows for; the human unwillingness to realise that the situation is not feasible to be effective and productive. When there is 'too much to do' sometimes the most productive approach is to take a step back and to properly review the situation, this however is not possible if the pressure of the overwhelming workload is constant – that does not allow for a constant 'stepping back and reviewing of the situation', it would be counterproductive. The problem, alas, begins before it reaches that stage.

> *"…the whole organisation has an intense changing pace, the more requests there are for changes the more it is that has to be done, it creates a bottleneck-effect, there can be 200-300 different requests at the same time in various decision groups related to our system…and all of them should be prepared appropriately and decided on…it is so many requests that it creates a pressure to let some of them through even though they may not be adequately defined and prepared – and the deliveries are a results from that…we have to make interpretations and adjustments along the way that all take unnecessary extra time later on…"* **[V IT 5]**

**The need for sharing information and documentation**
There are several views on documentation in project organisations expressed among the interviewees, for example that the burden of documentation is too big, that it in some cases the wrong things are being documented –there is criticism that there is more talk about 'to do it' rather than 'how to do it', some say that no documentation is unnecessary but it is important to keep it simple and easy since people are so busy and if it is not quick and easy to read and easy to understand it may have the reverse effect on the receiver of the information; perhaps not reading it at all, because of lack of time. At the same time there are also views that there is documentation that is missing and that the documentation that is done is not extensive enough, for example regarding customer's new requests during a project.

> *"…In a perfect world would everyone read everything and question everything, but that is not what actually happens, you can write down something and send it off and months later someone comes back to you and it turns out they have misunderstood it all and they have only read half of what you wrote, if at all…"* **[V IT 4]**

With an overwhelming, workload perhaps the first thing that comes to mind may not be to question, and respond to, information if not expressively asked to do so. In addition, if a response is expected but experience has taught someone that there will be no reply in return or no reaction to his or her response then it is more likely to react with some form of indifference to the information given.

> *"…people do not prioritise to read documentation when they are stressed or too busy, no documentation is unnecessary but it can not be too complex or extensive if it is to have the desired effect-which is to have the receiver to read it…"* **[V IT 4]**

Another view on the documentation is also that the documentation and information that is produced should be the right one and relevant for the employee to read. With an increasing workload does also the demands increase to have access to the correct information at the right time, therefore puts additional pressure on the sources of information to be accurate and updated in order to not cause even more work and confusion.

> *"…since the changing pace in the organisation is intense the need for forwarding and sharing information is incredibly big in the whole organisation,*

> *this is something we are not successful with, not on any level, not in top or middle management, not at operational level, not even inside working teams and groups – this is a serious problem… we have not got the time to share our information with everyone and we have not got the time to take it all in ourselves…"*
> **[V IT 5]**

> *"…it is really important that you can trust the sources of information presented to you, you can not always run to the customer to ask about everything, you have to be able to trust the contacts and structures to work appropriately…"* **[V IT 4]**

**Lack of resources**
Several interviewees speak of the importance of having the right people in the right positions if everything ever is supposed to run successfully, and it seems to refer back to the fact that there are not enough resources available.

> *"…people are just too busy; it does not create a good foundation for collaborations…everyone is just trying to catch up on things all the time, it is easy to say that you have to have people that are interested in their jobs; but it is more to it than that, most often I find that it is really a question about having resources at all…"* **[V IT 3]**

**[Survey]** After the question of what causes MOST problems in IT projects are the respondents of the survey asked *"Why do you think these problems occur?"*.

> *"Not enough people that have enough knowledge about the project up front and the time to devote to the project."* **[C]**

**[Survey]** On the question on **"***What can be done TODAY to avoid or solve some of these problems?"* does one respondent in the survey conclude:

> *"Not much without adding more resources "* **[V IT]**

**Personal development**
One great strategic risk with having a continuously overwhelming workload is that individual development and growth may come in second place when other aspects are prioritised higher.

> *"…all the company goals and objectives are broken down all the way through to individual levels, this is followed up three times per year… the interest for this is not always what one would like it to be… because of the workload do people 'forget' about their own personal development and make other priorities… "*
> **[V IT 6]**

> *"… the organisation is under-dimensioned in many places and can not act on everything that is expected from it…there are important competences among the employees that would be great to work with and develop further but that requires time, for example develop an understanding of business changes linked*

> *with system knowledge which is needed to improve the pre-study, there are only a few that can manage that confidently, that would be great to be able to work more on with everyone, that is a competence we need more of…"* **[V IT 5]**

**The big picture**

Interviewees speak of 'being able to see the big picture' as wishful thinking since the systems, technically and organisationally, are so very complex and large combined with an intense changing pace.

> *"…the overview and big picture is needed on the customer side as well, there can be 20-30 different business development projects that are run in parallel and they all affect our IT-system as well as many of them effect each other at the same time… I have said a few times that 'Why don't we draw a map covering all projects to see how it is all connected…?' and everyone laughs at the idea since there is no one that possibly can have that overview… I would like to have that … why not cut back on the changing pace, close down enough projects to allow us to have an overview, and to keep that in order for us to be able to see the actual consequences of what we are doing... To do that would take a lot of courage… it is really a managerial question, but it would certainly benefit us all…"* **[V IT 5]**

**The Steering Committee - Are the right decisions made?**

In a change intensive organisation increases the number of decisions made and the pressure to make the *right* decisions. There tend to be different people working in projects and sitting on steering committees but the overwhelming workload seem to be constant for everyone, listening to the interviewees. The project members are booked in many different projects that are all rushed for time alongside their own line responsibilities. The steering committee members that make decisions for many various projects are generally booked on several different project meetings – is it humanly possible to be well informed and updated with information on more than 5, even 10, different projects? Many interviewees, who represent both project organisations and members of steering committees, speak of the problem of steering committee members making decisions for projects that they are not fully informed on. The information is available to all steering committee members but whether they have the time or even the possibility to be involved with all various projects they are expected to make decisions on is not a given.

> *"…I do get called to go on a lot of meetings which I do not really need to attend, that is really unnecessary and time consuming but you have to respect everyone else and show up…I sit in on 10-17 different groups and make decisions at the moment…"* **[V IT 1]**

**[Survey]** On the question on "*Do you have access to enough information about your IT-projects, to allow you to do a good job? Or what would you like to change?"* does one respondent in the survey write:

> *"The information is available but the Steering Committee should allocate more time to perform their functions "* **[V IT]**

### Conclusions on Work organisation

With an overwhelming workload for project members is the work in a project organisation inevitably be affected, most notably perhaps is the fact that too much to do often translates into inaccurate decisions and the quality of the work done being lower than it normally would should there instead have been sufficient time to prepare appropriately. The rapid changing pace for the organisation is one contributing factor, the lack of sufficient resources another.

## 5.7 IT-product Vs. Business value

*The problems created from the supplier and the customer having different perspectives in projects.*

> *"We [at Volvo IT] should work with and change the customer's processes instead of just adding tools to them, we need to step up from being technicians and start working with business value instead, this is Volvo IT's responsibility"* **[V IT 7]**
>
> *"…the customer does not always have knowledge of IT…and the IT people do not always understand the customer, it is something everyone has to learn …"* **[V IT 6]**

There is an inevitable fundamental difference between the customer and the supplier's view-point in software engineering projects. The customer's focus is on *the business value for their organisation*; the *change* in their organisation that the new software may bring and they will 'live with the product'. As such the customers may not necessarily see an end to the project with a release since it is in fact on-going thereafter in their own organisation. The supplier on the other hand has a focus on the *product*, to be able to *meet the time and cost* for the project and focus on achieving a satisfactorily *delivery* of the product. The supplier view a project with a start and a finish and the customer see a change to their processes that will be continuous. Several interviewees point out the importance of understanding this difference regarding the problems that arise in collaboration in projects.

> *"…The customer's steering committee has people with knowledge and responsibilities over the production line in the customer's organisation, their interest are <u>the result</u> of the project; the product…The Volvo IT steering committee has people in it with responsibility over and knowledge about IT processes. There is a constant conflict of knowledge between these two; there can even be a conflict of interest between them. It may look good on paper but in reality it may be much more complex… One side has its focus on <u>what</u> is being delivered and the value that may generate; the other side has a focus on <u>when</u>- the actual delivery – to deliver on time and to the right cost etc…"*
> *"…It is an ongoing conflict…the customer is a line organisation that will live with the result from a project for a long time after the software has been delivered, they may be reluctant to 'let go' of a project because of that …"*
> **[C 2]**

This may be linked to the theme Organisational legacy[160] as well, 'assuming too much'. It is the customer that realises that there is a need for a new product or processes in their organisation and they then need to explain this specified need for the supplier in order

[160] See chapter on Organisational legacy

for the supplier to understand this need and to create a solution to the problem with an understanding of the customer's business. This is where Volvo IT sometimes falters according to the interviewees. Problems such as having products that are not successfully implemented in the customer's organisation after a project may be a result from this – that the reason is a lack of understanding of the customer's business and fundamental needs on the supplier's side.

> *"…A successful project is when you manage to really implement the system into the customer's organisation and the system is subsequently put into actual use the way it was intended, developing the software is the easy part, making it work in the organisation is the difficult bit… there need to be clearly stated in the scope of the project that the system should be implemented in the customer's organisation in the end…"* **[C 1]**

**The understanding of the scope**

What was included in the initial scope seems to be a reoccurring topic up for discussion. The customer may say something "obviously" was included in the scope and Volvo IT says it was not specified and is therefore not included. Without the same view-point for a project may the number of misunderstandings increase regarding what is actually agreed.

> *"…that is OFTEN the discussion… The customer says it was included in the scope all along and Volvo IT says it is an additional function… What is really important is that we learn how to make really good specifications in the beginning of a project as to learn what the underlying needs and requirements are for the customer…"* **[V IT 6]**

**Shift of focus**

If the supplier could create a "shift of focus", to be more that of the customer may a greater understanding be possible overall which could then lead to increased precision in delivery of projects and also a happier customer alongside increased success rates for IT-projects, similar thoughts are expressed among the interviewees.

> *"… we need to understand that it may not be the system for the customer that is the most important but instead the actual change in the customer's business. There need to be a focus on the actual purpose with the project throughout the whole project organisation with everyone involved, including the customer and all steering committees, this is not done today and there is a need for it…"*
> *"IT-projects are change projects…You can not only develop tools and products and think you have achieved some kind of business value, the IT tool is perhaps 20% of the change, 80% is a change in the processes and to change peoples behaviour… it can not be complicated if people are to understand it and embrace it… it has to be easy for the steering committees or they will not be able to understand what they decide on in the steering committee meetings, it should be viewed as an iterative process on all levels to emphasise and keep emphasising the focus and purpose of the project…"* **[V IT 7]**

### Sell the product

Interviewees point out the implementation of the product as something which is too often overlooked and not properly performed. A comparison is made with the old way of thinking regarding the new; implementing change; that it is almost expected to be function *automatically* because people are being paid to do their job and should therefore accept the changes, and that that way of thinking is outdated and not very efficient in today's complex organisations. More emphasise should be put into 'selling the product' to the customer, to the end-user, and this should not be ignored if higher success rates are desired in IT-projects, arguing that the supplier should take on a conscious greater responsibility regarding this.

> *"…when you have more complex matrix organisational structures as there is today it may not always be that easy, you may have to actually <u>sell the solution</u>, sell the fact that there are going to be changes in the organisation in order for this tool/system/software to be implemented successfully in the organisation …"*
> *"…the classic scenario was that you had a line, one system and changes were introduced the militant way from top management, these days there are systems that are run right across organisations through several business units and that brings a new situation… often this is not calculated for, in order to keep the costs down this is 'forgotten about' and the line is supposed to just be able to implement the changes themselves, people are being paid to work so they should just do as they are told… that kind of thinking does not work anymore… Volvo IT have to be able to sell that kind of understanding and knowledge to the customer… it is important to give the business project manager the right arguments for him or her to take with him or her into the organisation and to help them sell the solution all the way to the end-user, cost arguments are not effective in this case …"* **[C 2]**

### The time aspect

The longer a project runs the more likely it is that the requirements of the product changes. Changes in society, financial, practical needs, and add to that there are several customers in a project that all must agree on the main purpose for the project. Interviewees explain that the longer a project runs the more increases the probabilities that the requirements for the project and product changes. If a project runs over a time period of 3-5 years for example, or longer, there will inevitable be additional requests from the customer that are not going to be found in the initial document for the scope or could have been foreseen in the pre-study no matter how extensive it may have been. With larger projects the need increases for a fundamental understanding of the business value for the customer in order to be able to accommodate additional request without being taken by surprise by them, or perhaps even foresee them before the customer realises the need for a change themselves.

> *"…the customer changes his or her mind sometimes during a project, there are so many different parties involved with different interests in a project …"*
> **[V IT 1]**

**Comments from the survey related to IT-product vs. Business value**
**[Survey]** On the question *"How can the communication between the customer and Volvo IT be improved?"* do some of the respondents talk of the different point of views the both sides have, what the problems are and what can be done to improve it.

> *"It depends on the projects and the people involved. Mainly I think a better understanding from Volvo IT of the user's business would help understand the requests made and therefore tune more accurately the proposals made"* **[C]**

**[Survey]** Other comments on the question on what causes most problems in IT-projects and why do they think these problems occur:

> *"Not enough communications to help the customer analyze what they really need and then to clearly define the IT deliverables. "* **[V IT]**

> *"Customers don't see the big picture of software impact"* **[V IT]**

> *"Because most of the people in Volvo IT have no idea about the IS-GDP. Because Volvo IT can be defending its products instead of taking care of the customers' needs. Little capacity from Volvo IT to add resources to meet customer requests. Because the split between Volvo IT and the customer is not always clear. There are resources that sit on both sides of the fence and do interfere on the priorities."* **[C]**

**[Survey]** After the question of what causes MOST problems in IT projects are the respondents of the survey asked: *"Who/What causes these problems? (A specific work title? A phenomenon like, everyone's workload is too big?)?"*

> *"When a report on a project is requested, I see too many COST related problems and less focus on the "content"."* **[C]**

> *"The customer's expectations are often unreasonable, and there is a tendency on the IT side to propose unreasonable solutions to anticipate customer's reaction."* **[C]**

**[Survey]** What can be done today to avoid or solve some of these problems was one question in order to find out if the respondents of the survey had own ideas about how to address these problems directly, today.

> *"Understand the processes of Volvo IT and the customer. Try to communicate more within the project, from the beginning of the project, to the end, including after for follow up. "* **[V IT]**

> *"- Assist the customer in gathering requirements*
> *- focus more on "deliverables": what do we want to achieve with this project and so what do we have to deliver for that, not only concepts but also practical things."* **[C]**

> *"Identify more stakeholders and spend more time on explaining to them what will be delivered (new process and tools)"* **[C]**

**[Survey]** After several questions on of what causes problems in IT projects the respondents were given the option to comment further:

> *"Come up from the pre-study with an agreement and UNDERSTANDING of the scope is crucial. Both parties must understand each other's difficulties and must compromise, but in the end, biggest efforts must be put on adapting the tool to the process or expectation, and not the contrary."* **[C]**

> *"Globally, I'd like to see more customer oriented attitude from Volvo IT to adapt the tools and proposals to customer's processes and need instead of pushing to adapt the customer's needs to the tool."* **[C]**

> *"I see a potential risk with IS-GDP; introduction and usage have to be practical, not dogmatic. A benefit must be clearly visible for EVERYBODY. If the process is excessively complex, users will try to outsource from other suppliers to avoid internal Volvo IT bureaucracy. Local opinion in general about Volvo IT is anything will take at least two years to be done, and that it is far more expensive only to try to comply with IT governance"* **[C]**

**Conclusions on IT-product vs. Business value**
The fundamental difference of focus between the customer and the supplier causes misunderstandings in software development projects and one result of this is systems that are not successfully implemented into the customer's business and alas IT projects that are not successful. A shift in the focus of the supplier is called for towards that of the customer, in order to understand the customer better, to manage implementation of software successfully and to increase success rates for IT-projects in general.

# 6 Discussion and analysis

*This chapter presents a discussion on the material from the results chapter. The discussion is focused on whether agile methodologies may be applicable for Volvo IT in any form today and if it could have a positive effect on any of the problem areas or if it instead - to the contrary - could have a negative effect.*

There are inevitably many various opinions in a large company. One interviewee states that one of the most important tasks that has to improve in projects are that the scope of the project must be agreed upon and fixed ("locked") *earlier* in the process, and not be allowed to change. Another interviewee is just as convinced that the problem is that the scope instead should be allowed to change; that the problem is that there is no good way of dealing with changes and requests as the project has started.

There is a complexity and many different ideas and views are represented in a large organisation. Volvo Information Technology employs +5600 employees worldwide; the company accommodates Volvo Group company customers worldwide alongside other customers. The organisation is incredibly large and widely spread and all parts of it are tied together in a multi layered web with multiple connections. Looking at the many various project models and methods that are being used and practiced at Volvo IT it does seem as if all possible aspects and risks are addressed and that "everything is covered". However, if this was true there would not be any unsuccessful IT projects.

## 6.1 Identified problem areas & agile methodologies

The material in the discussion and analysis chapter is presented as in the results chapter with themes and additional headlines from the various topics that were brought up in the interviews and the survey. These are then commented on from an agile point of view; would agile methods affect the theme and would it have likely positive or negative effects.

The themes and problem areas are:

**Being global**
- Face to face
- Teambuilding
- Organisational culture differ between Volvo sites
- Extremely large projects

**Organisational legacy**
- Assuming that everything is understood
- Scope creep
- Unwanted customer service
- Professionalism between former colleagues
- Internal administrative functions
- Technical history

**Line vs. projects**
- Management level - Prioritising the work and various projects
- Operational level - Who/What decides on who is doing what?
- A focus on project management

**Role definition**
- Managing uncertainty
- Role definitions may affect the formation of a project group
- No official project document on role definitions
- Having the right people in the right positions
- The list of requirements – whose responsibility is it?
- Comparison with product development projects

**Work organisation**
- The Steering Committee - Are the right decisions made?
- The big picture
- Personal development
- Lack of resources
- The need for sharing information and documentation
- The intense changing pace
- Keep up the structure – control chaos
- The pre-study

**IT-product vs. Business value**
- The time aspect
- Sell the product
- Shift of focus
- The understanding of the scope

### *6.1.1 Being global*

There is a clear consensus among the interviewees and the respondents of the survey that communication needs to be improved in IT projects - even more so because of the project's global status. It is impossible to fully comprehend how large and widespread the effects of a project is in the organisation and how many levels one project may affect and how many interdependencies there unavoidably are throughout several organisations. The sheer magnitude of the projects complicates collaboration, organisation, planning, execution and communication. These areas, contradictorily, all needs to be even more efficient in large and sometimes safety critical projects since even small mistakes in those projects can become very serious and costly.

This also creates and magnifies uncertainties because of all different factors that will effect and influence a large project, and with the size of the project increasing; the seriousness of its problems will increase as well. If a project becomes too big for everyone involved in it to not be able to communicate effectively it is sure to create problems and complications that will have an undesirable affect on the outcome of the project in time, cost, quality and business value for the customer.

**Face to face**

Not being able to see the colleagues and co-workers complicates communication and thus consequently collaboration. It is under normal circumstances difficult to fully understand and communicate well with other people, and considering the fact that most of our communication is not verbal at all, it is not surprising that the aspect of not being able to see one's colleagues or customers is brought up as one common source for misunderstandings in global projects.

The language issues and cultural differences play a part and it is easier to overcome those issues and still create an understanding of each other if everyone are physically in the same room when they communicate. However, without being able to see each other and additionally have different mother tongue and culture complicates the situation further.

Agile methodologies preach of face to face communication as the supreme form of communication that can not be overrun. In addition, that the best form for sharing information and knowledge between people is by having those people in the same room together. This may not be fully feasible to achieve for large global organisations. This is one area where it is clear that agile methodologies are difficult to attune to larger project organisations'[161] reality in comparison with the smaller flexible projects the methodology was intended for originally. However, there are suggestions for practices on how this issue can be addressed in large organisations should there be a wish to work in a more agile way. Large projects should preferably be broken down in to as small teams as possible and then co-ordinated with representatives meeting up regularly for workshops to discuss the latest progress. These small workshops should not have a set agenda other than discussing the latest and next iteration in each small project team[162].

[161] Lindvall et al, 2004., *Agile Software Development in Large Organizations*,
[162] Highsmith, J., 2002, *Agile Software Development Ecosystems*

The representatives would after these small workshops be able to go back to their each individual project team fully updated on the progress of all related project groups involved in the same project. The iterations in the separate smaller projects could be linked together at various levels in corresponding functions, or whichever way seems most appropriate for the project at hand. Other suggestions could be to look into possibilities to incorporate daily or weekly 15-30 minute project video meetings including all team members, even in global teams; in accordance with Scrum project management practices.

It can be concluded that the issue of not being able to meet with everyone personally often in large projects should be taken seriously and be appropriately addressed, more than it is today at Volvo IT and its customers. It could be effective to consciously in the organisation - at a higher managerial level – raise a consciousness that large project teams may be counterproductive in its sheer size and that smaller project formations would likely – given the right conditions – improve the total flexibility, manoeuvrability and response rates for the overall bigger project.

The all important aspect here is that it is not enough to realise that it is a problem unless no new work practices are being implemented in the work methods and then allowed the right conditions to be effective and productive.

**Agile principle 6**:
The most efficient and effective method of conveying information to and within a development team is *face-to-face* conversation

**Teambuilding**

The importance of organising teambuilding activities for a project group is mentioned by the interviewees regarding how problems and misunderstandings can be avoided in a project and how new team members can be brought up to speed with the projects' progress. In an ideal world perhaps it would be possible to have teambuilding activities across country borders and across various company sites and departments, but although large, expensive and global status of projects as most projects at Volvo IT are, at an organisation at the size of Volvo IT it is not feasible to believe that teambuilding in its original sense could be implemented in the standard project plan. Possibly for the biggest and most critical projects, however, virtually every project is global in some form and "smaller" projects are still large, expensive and complex. This is an area that agile methodologies do not address directly. It does address it indirectly by stating that project groups should be physically based together and there is an emphasis on the individuals in the project instead of the processes and that the best communication is the one that is face to face. With that way of working can almost teambuilding be argued to be incorporated into the daily work practices.

Activities as suggested above with regular though informal (video) meetings with project members all over the world could be one way to address this. However, people being based in different time zones are one aspect alongside everyone being very busy and fully booked. It would most likely be difficult to motivate team members to schedule even more meetings in to their schedules unless they would see the direct benefits of it and this is almost impossible to 'show', the approach would have to be 'sold' to the team members before they can be expected to be positive towards adapting it.

**Agile principle 4**:
Business people and developers must *work together daily* throughout the project
**Agile principle 6**:
The most efficient and effective method of conveying information to and within a development team is *face-to-face* conversation.
**Agile principle 11:**
The best architectures, requirements, and designs emerge from *self-organizing teams*.

### Organisational culture differ between Volvo sites

The fact that various sites in a company have been given slightly different directives for how to work and address project organisations and the work involved is a question for higher management levels. Regardless of which project methodologies that may be chosen for an organisation it is of utmost importance that all involved, on all sites where team members may be situated, are given the same training and directives to follow or it is sure to create problems, misunderstandings and difficulties in project organisations when different approaches are put together to function as one. The experiences expressed by the interviewees and the respondents of the survey tell that this is a problem in Volvo IT's global organisation.

Agile methodologies do not address this issue other than in the form that project teams should be physically placed in the same room or area and through this there is inevitably a common project culture developed, regardless of any previous formal directives there may have been. 'People trump processes'[163]. Nevertheless, problems on accessing organisational project tools such as commonly used software, mentioned by a survey respondent, is a problem that can not be overcome by a project team becoming more closely connected on a social level. All project members and participant should have access to the same tools and be given the same conditions by which to work under if collaboration should be supported effectively.

### Extremely large projects

The views that the projects are simply too big in size is brought up by the interviewees and the respondents of the survey. The negative aspect of this is are, among other, that the projects become;

- Inflexible
- Difficult to manoeuvre
- The big picture is lost
- Issues to solve communication which are difficult to overcome
- The wrong decision may be made, it can almost be expected – as a result of no one being able to keep the overview of the desired outcome and be able to take all affected factors in to account
- Customer's changing requirements are increasingly difficult to accommodate, and the bigger a project becomes and the longer it will run the more requests

[163] Cockburn, A., Highsmith, J., 2001, *Agile Software development: The People Factor*

there are likely to be because of the ever changing environment surrounding the organisations.

This is one idea behind agile methodologies, that people work together more effectively in smaller groups which increases productivity and flexibility, and therefore should as small project groups as possible be desired and large projects be broken down into smaller part-projects. This is with the added problem of inter-team-communication issues which can be difficult to solve effectively, but still it would be preferable to have large organisations with no internal groupings into smaller formations that can be left to work as smaller groups on their own to a great extent.

One important factor with large projects is the **staff turnover** among team members. The bigger a project becomes, for the longer period of time it runs, the more likely it is that staff and team member changes in the project. This is more likely the bigger the project organisation. In agile this is considered as very serious since the agile projects' structure it based around the individuals that make up the project team and the loss of an important member may be 'catastrophic'. One way to minimise the affects of this can part be to use rotation on the working roles in a project team, which also allows team members' individual skills to be developed. That means that if one member were lost in a project someone else would be able to take over the former team member's job duties easier than if the job functions had been run by only one person. It can be viewed out of a safety perspective, to not loose too much important knowledge that may be crucial to a project if one team member were to be lost.

Another way to address the issue of staff turnover in a project would be to keep relevant and updated documentation on the projects progress which could help shorten the learning curve for any new members to the project organisation. Agile advocate minimised documentation in order to minimise unnecessary work – meaning documentation that may not even be read or used later on is unnecessary and may only take time away from more important work in the project. In large organisations it is not possible to completely cut back on documentation because changes need to be able to be traced back for maintenance, and out of safety issues the history of a large project needs to be documented. The documentation is also a way for new project members to be informed of the status of a project.

**Agile values:**
Individuals and interactions over processes and tools
**Agile principle 4**:
Business people and developers must *work together daily* throughout the project
**Agile principle 6**:
The most efficient and effective method of conveying information to and within a development team is *face-to-face* conversation.
**Agile principle 10:**
*Simplicity*--the art of maximizing the amount of work not done--is essential.
**Agile principle 11:**
The best architectures, requirements, and designs emerge from *self-organizing teams*.
**Agile principle 12:**
At regular intervals, the *team reflects* on how to become more effective,
then tunes and *adjusts* its *behaviour* accordingly

### Conclusions

Agile approaches are meant to increase flexibility, agility and to be more adjusted to the environment where software development projects are present and working today. This is a contradiction to large global project organisations with no overview and multiple interdependencies that can not be effectively monitored. However, nothing speaks against incorporating ideas and practices from agile methods in order to increase agility even in large projects though keeping in mind that the fundamental conditions are different and that that needs to be fully understood. The ideal approach would likely be to break large projects into smaller projects which would become more flexible. This idea is brought up both by the interviewees, respondents of the survey as well as recommended by the agile methodologists.

### 6.1.2 Organisational legacy

Several aspects of the problems that can be linked to the history of having been the same company as the customer, according to this study, can be expected to be affected - should agile methodologies be implemented. Even though agile methodologies do not address the actual situation of having a history together with the customer, many of the problems that arise from it can be addressed with some agile practices.

**Assuming that everything is understood** and **Scope creep**

The problems described in **Assuming that everything is understood** and **Scope creep** are related in the sense that they arise when the two sides, of customer and supplier, believe that they are in agreement but it later turns out that their opinions indeed are different from each other regarding what was actually agreed.

Agile methodologies are different from the traditional waterfall model[164] in that there is no detailed scope and no extensive pre-study where all requirements are gathered. Instead the formation of a project group that (preferably) *includes* the customer overcomes these problems by close collaboration and an iterative approach to the work in progress. The assumption is not that it is possible to find out all requirements beforehand and create a complete understanding of each other but instead the opposite; that it is not possible to understand each other so instead we need to work closely together in order to be able to view and review and discuss the features *together continuously* and only by doing that a successful understanding of each other is possible to achieve.

**Agile principle 1**:
Our highest priority is to satisfy the customer through early and *continuous delivery* of valuable software
**Agile principle 2**:
*Welcome changing requirements*, even *late in development*.
Agile processes harness change for the customer's competitive advantage.
**Agile principle 3**:
Deliver working software frequently, from *a couple of weeks* to *a couple of months*, with a preference to the shorter timescale.
**Agile principle 4**:
Business people and developers must *work together daily* throughout the project

**Unwanted customer service**

Though not seemingly a big problem (according to the survey) the problem of unwanted customer service with unnecessary features being added to the product without the customers' request for it, does exist. It could also be avoided by having a customer involved *continuously* in the development process in the project organisation.

[164] "*Managing the Development of Large Software Systems*", Royce, W., 1970 explained in Larman, C., 2005, *Agile and Iterative Development – A Manager's guide*, p.103-106

With a closer working relationship with the customer there is no room for 'assuming' anything, or if so, any assumptions can be cleared at the time they arise.

The prospect of having a customer present during the duration of a large project is however not always possible to arrange, particularly not if the customer consists of several stakeholders based in different countries. Nonetheless, if the idea that everyone would work better if physically based closer together and that the best idea to communicate progress of software is not by presenting numbers and ticking off a pre-designed check-list; then various ideas can be discussed on how a closer working relationship can be 'simulated'. Ways to do this could involve iterative processes with iterative deliveries of the product with useable code that the end-user can visualise and fully understand by seeing it. The real end-user in that case would need to be identified, or a knowledgeable representative of the end-user, the members in the steering committee with decisional powers may not necessarily be the 'real' end-users according to the collected data for this thesis.

**Agile principle 1**:
Our highest priority is to satisfy the customer through early and *continuous delivery* of valuable software

**Agile principle 4**:
Business people and developers must *work together daily* throughout the project

**Professionalism between former colleagues** – as above, does not really apply to the aspects of agile methodologies. With closer working relationships this problem should 'automatically' be resolved.

**Internal administrative functions** – the problem of having slow or complicated in-company administrative functions would most likely not be resolved or affected by any incorporation of an agile approach to project management, it is a problem which needs to be brought to attention and be addressed at an executive management level in the company since it is directly linked to the company infrastructure.

**Technical history.** It is an impossible task to 'clean up' or in any thorough way affect or change the existing technical systems at Volvo IT which affects all Volvo Group software worldwide. The agile approach dictates continuous refactoring[165] and continuous testing of all code and software throughout the development process in order for problems and mistakes to be discovered early. This approach also dictates continuous integration of new code to the product in order for problems with integration to be discovered early. All code is written iteratively and simplicity should be the aim for every part of code that is written. The agile practices in XP with pair programming, with two programmers working on one computer, is one way to administer this

[165] Refactoring = rewriting programming code to improve its readability, maintainability, structure or performance *without* changing its functionality or behaviour. Informally referred to as "cleaning up the code". Each refactoring should be very small and accompanied by automated tests of the code to ensure that the behaviour of the code has not changed. The reason to perform refactoring is to improve understandability and make human maintenance easier in the future. *The impact of Agile Methods on Software Project Management*, Coram,M., Bohner,S., 2005, also; http://www.refactoring.com/, also; Larman, C., 2005, *Agile and Iterative Development – A Manager's guide* , p.149.

approach. Assuming this approach is welcomed by the developers it may be interesting to discuss further if the practices could be productive to implement at least at some stages through the development processes in Volvo IT's large software development projects. As agile dictates; every project should be viewed individually and what may be right for one project at one stage, may not be right for every similar project there may be.

The approach to perhaps adopt agile practices to Volvo IT's established software development organisation would not solve problems with existing old code that has been produced over many years but would, supposedly - given the opportunity of the right development conditions, have a positive effect on present and future software development in terms of facilitating future maintenance and up-keep of the programming code and systems.

**Agile principle 9**:
*Continuous* attention to technical excellence and good design enhances agility.
**Agile principle 10**:
*Simplicity*--the art of maximizing the amount of work not done--is essential.

### 6.1.3 Line vs. Projects

The conflicts and problems created from having both line responsibilities and project responsibilities (in several projects) are not addressed by agile methodologies as it is a project methodology and not an organisational architectural approach. Agile methodologies presupposes that project members should be available to work in joint development processes since it is part of the fundamental conditions which agile strives to create; a common project culture with close working relations to achieve flexibility and increase productivity. Nevertheless agile principles point out the importance of allowing project teams to be self-organising since those teams are the once with the best productivity and efficiency according to agile.

**Management level - Prioritising the work and various projects**

An agile approach is likely to create a conflict with the existing decision-making hierarchy in a traditional software development organisation. Agile dictates that the project development team has the power to make its own decisions and has a representative and one or several empowered customer representative present in the team that will be able to directly and without lengthy delays be able to decide on features and changes to the product. Project teams are often held accountable for product delivery and release, but seldom given the direct decision power to control it[166] this is instead left to outside functions such as steering committees and other management groups as change decision groups and change control boards. An agile approach to software development projects need to be understood by all stakeholders and the decision-making powers should probably be discussed and decided on beforehand. Perhaps the project group should be taking on a bigger part of the decision-making responsibilities (in accordance with the agile approach to 'trust people to be able to do what they are paid to do') since the presence of a customer in the project group would have the additional role of making sure the features are in accordance with the requirements. The steering committee role could possibly take on more the role of cost controlling function, if cost and time is set in the scope and the features are more flexible and planned to be developed during the development process.

The idea of having external control functions outside the agile project team goes against the effort of creating a truly agile team since answering to an outside function requires a gathering and preparing of presentation of material from inside the project team, and this inevitable takes time away from other development work with the product, and as such will affect the agility. Awaiting decisions to be made by an external group may also delay progress in a project group and affect its agility[167]. These factors need to be understood and addressed. Perhaps the most important aspect is to understand them and bring them to attention and make sure that all involved stakeholders are aware of them since abolishing the traditionally and very established role of the steering committee function completely may rather cause more problems than it would solve them.

[166] Cockburn, A., Highsmith, J., 2001, *Agile Software Development: The People Factor,*
[167] Lindvall et al, 2004., *Agile Software Development in Large Organizations*,

**Agile principle 5**:
Build projects around *motivated individuals*. Give them the *environment* and *support* they need, and *trust* them to get the job done.
**Agile principle 11**:
The best architectures, requirements, and designs emerge from *self-organizing teams*.

**Operational level - Who/What decides on who is doing what?**
As mentioned in the beginning of this text on Line vs. Project Responsibilities is the problem of not being able to dedicate 100 percent of a working week to one project not addressed in agile methodologies since it is a more a problem created by an organisational structure. It is an important issue to address irrespectively since it can be argued that the organisational structure does not support its own work practices (of being a project oriented IT development organisation) but this thesis aim is to focus on the aspects agile methodologies may affect.

**A focus on project management**
The request for a greater focus on project management from the respondents of the survey and the interviewees are linked to this theme with respondents for example saying that the role of the project manager is not taken seriously enough by the higher management/organisational structure since a project manager may not be 'allowed' to focus on one project enough because of other line responsibilities.

**Conclusions on Line vs. Projects**
As stated above agile methodologies do not address the conflicts and problems of having both project and line responsibilities at the same time. However, the decision making groupings that affect decisions regarding project teams will have an impact on any eventual agile approach to software development projects since the agile approach is formed on the conditions that the project group is an adjustable 'eco-system' that has the only goal of developing software with the best business value for the customer, and tying that evolving 'eco-system' to fixed external control functions that have an overriding monitoring and decision making function will be counterproductive. This 'fixed external control function' however is most likely not feasible to do completely without in a large organisation with software development projects spanning several years.

### 6.1.4 Role definition

It can jokingly be said that by *not* defining the roles in an agile project would be an effective way to make that project unsuccessful[168], and this is most likely to be true no matter what kind of project methods and practices that are used. If a project member is unsure about what is expected of him or her it creates uncertainty, for example regarding ownership of responsibilities and functions in a project, this may lead to problems such as functions that are overlooked and it may translate in to time delays and eventually cost increases for the project and the customer.

Another important factor is the aspect of job satisfaction, if a project member is uncertain about his or her job assignments or do find out that he or she has devoted time to the wrong tasks it may lead to frustration. The basis is rather straight forward; in order to do a good job there should be minimal confusion as to what is expected from a specific working role in a specific project, also, there need to be as good an understanding of what all other project members are assigned to do as well for a project organisation to be successful.

When an employee is fully aware of what is expected of him or her it creates a feeling of responsibility in a different way than if it is surrounded with uncertainty and confusion. In order to be able to exceed expectations, should that be an ambition (an ambition which ought to be supported by every organisation), does the expectations for the working role need to be clearly understood by everyone without doubt.

**Managing uncertainty**

The uncertainty created from the confusion of not having clearly communicated job descriptions on every project is not desirable in any project model, traditional or agile or any other. The project methods used at Volvo IT do surely have job descriptions for the various roles; the problem seems to be more that the roles are not updated in every new project, and they are also not continuously updated as the project progresses (which can be for a period of several years) when new people join the project team or when someone is leaving. This means that it will be left up to each and every one in a project to have their own opinion on the job description and responsibilities of that working role in a project, and as this study shows this is a cause for misunderstandings and problems in projects.

The idea that every project is unique and can not effectively be compared to any other project is dictated in agile methodologies. It is encouraged to only apply the techniques and practices that are suitable for the project at hand and understanding the uniqueness of every project is very important in order to choose the appropriate work practices and approach[169]. The working roles are important to define and communicate to everyone and make sure that they are understood by everyone. Working roles in agile projects can for example be manager, architect, developer, agile coach and product owner.

[168] Gurses, L., Dr Dobb's journal, http://www.ddj.com/dept/architect/193402902

[169] Highsmith, J., 2002, *Agile Software Development Ecosystems*, also
Fowler, M. at http://www.martinfowler.com/articles/newMethodology.html#ShouldYouGoAgile

However, the understanding of the importance of having clearly understood working roles in a project is far from unique for agile methodologies. It is more likely to believe that the importance of updating the job descriptions is 'easy to forget' in the intense changing pace and vast quantity of projects at Volvo IT. Applying an approach with agile methodologies would probably not address this problem in other ways than by being a new methodology and as such it would need to be understood by everyone because those project roles are new.

**Role definitions may affect the formation of a project group**
It is essential to understand the working roles in a project and the various job descriptions as well as understanding the scope, 'mission' and goal of the project if the formation of the team should be appropriately designed in order for the project to be successful.

In accordance with the project models used at Volvo IT do the appointed project manager present his or her requests for resources (~ number of project members) to the steering committee for a project (and this routine is repeated whenever the project changes or transforms). Many factors play a role in whether the project manager is given the requested number of resources (availability of resources, priorities for- and importance of the project, and more) and already at this stage in a project can problems be caused if not for example the project manager and the steering committee members have the same perception of what various roles involve.

**No official project document on role definitions**
As discussed in some of the interviews; one approach to address this particular problem could be to, regardless of whichever methodology practices:

1. Realise that it is a problem
2. Address it accordingly

One hope is that this study in some small way may help to bring knowledge about the problem to some level and highlight the idea brought forth in the interviews; to have – and continue to use – a document of some form in *every* project that is updated continuously regarding the job descriptions and changing responsibilities.

**Having the right people in the right positions**
It is easy to draw the conclusion that if only the right people were placed in the right positions many problems would be avoided or immediately solved, in projects and in organisations in general. This can be as straight forward as being an HR (human resources) question; of people 'ending up' in positions they may not really be suitable in. However, when there are varying opinions on what the various working roles are and what the responsibilities of those roles are can be contributing to the opinion of 'not having the right people in the right positions' because there in fact are different opinions on what is expected of the many various working roles.

If the organisation is not supporting the working roles that its structure consists of efficiently it is likely to create confusion and frustration both for those who unknowingly or knowingly are 'in the wrong positions' as well as for those who perceive it as a problem that 'other people end up in the wrong positions'. If this is true, and it can only be speculated on this, would it mean that the people employed are expected to succeed in working roles that are not fully supported by their organisation, undoubtedly would this cause problems with misunderstandings and having people 'not stepping up to their plate' or indeed 'stepping outside of their plate'.

**Agile principle 5**:
Build projects around *motivated individuals*. Give them the *environment* and *support* they need, and *trust* them to get the job done.

### The list of requirements – whose responsibility is it?

This was not planned as a separate study-in-the-study but during the interviews the question and frustration about the list of requirements were brought up as a regular topic as something that is the cause for many problems in projects. The list of requirements was said to often be incomplete, not detailed enough, not be understood as important by those writing it, and often that those/that one who compiled it did probably not have a proper understanding of how it actually should be put together in order to meet the requirements of a "list of requirements" for a project. One natural follow-up question would be;

- *Whose responsibility is it then to write the list of requirements with the customer, and who actually writes it (if those two are not the same)?*

When three separate interviewees had brought up this list of requirements as one cause for problems, without having the interview questions specifically asking about it, and none of the interviewees were really sure on whose responsibility it really was; I decided to ask every interviewee about this from then on. The list of requirements was mentioned by virtually every interviewee at some stage as problematic in projects but just one interviewee were sure that it was the business analyst's responsibility. Some of the interviewees were *quite* sure that it was the project manager's responsibility but most seemed hesitant and other suggestions of working roles came up as well. This was not a scientifically justifiable research-in-the-research but it can be argued to be indicating that roles are not properly communicated when many project members are not sure on whom they are 'accusing' for not having succeeded with performing one of the project's important functions, not even the person whose responsibility it is may be aware of it.

**Agile values**:
Customer collaboration over contract negotiation
**Agile principle 2**:
*Welcome changing requirements*, even *late in development*.
Agile processes harness change for the customer's competitive advantage.

### Comparison with product development projects

Perhaps should software development projects be compared to product development projects in order for the confusion on role descriptions to be cleared up? These thoughts came up in some of the interviews. The big difference being that in product development projects everyone knows what they will end up with in the end (one car, four wheels, windscreen, etc), and that is not a reality for a software development project. However, as an attempt to make the roles easier to comprehend for everyone this comparison was mentioned.

Agile emphasises the difference between the two types of projects and the problem is addressed instead by closer working relationships and working roles that correspond better to the individual project at hand.

### Conclusions on Role definition

An effective project organisation requires that everyone knows what is expected of him or her in his or her working role, and it can be concluded that on this point there are confusion and misunderstandings in IT projects at Volvo IT. Agile methodologies do not address this issue specifically more than many other project methods – it is important for everyone to understand the working roles, one's own as well as everyone else' for a project to be successful. A project consists of people and it is of utmost importance that the working roles are understood in order for problems - created from any confusion from this - to be avoided.

### 6.1.5 Work organisation

When there are statements saying that 'there is just not enough time to do everything right' it should be viewed as a serious risk since it indirectly means that there are functions that are not performed effectively. When 'having too much to do' is almost considered to be the norm many areas are likely to be affected. If nothing controls and regulates the employees' workload the production levels are going to drop in the long run and have a negative effect on both time and cost and success rates for projects.

**The pre-study**

The pre-study is criticised for being a regular source for problems in IT projects for similar reasons as the list of requirements that; it is not done thoroughly, it is not detailed enough, it is not always done by knowledgeable employees and it is not given enough time to be produced. This problem arise in the traditional waterfall model[170] where there is an initial "pre-study" phase for gathering all requirements in one place leading up to the first gate[171]. The best solution to this seem to be to raise an awareness of *the importance of the pre-study* and to 'sell' this understanding to the customer for the customer to be willing to pay for the extra time that it would (should) take.

In an agile approach the need for a pre-study is replaced with early iterations and close working relationship with the customer among other agile practices and the underlying opinion is that it is not possible to find out everything that is needed for a project beforehand, and even if it was; those requirements are likely to change before the project is finalised anyway. Looking at it bluntly; agile could solve the problem of not having sufficiently conducted pre-studies by minimising the factual need for a pre-study. It should however not be interpreted as that there is no documentation being done in agile.

In the idea of minimising documentation in agile - and additionally minimising the work that may be unnecessary - do agile dictate *less detailed* documentation regarding such documentation as project scope and project vision. By not having this documentation too detailed – when the scope changes as the project progresses – would additionally the need for updating it be minimised as well, thus avoiding further 'unnecessary' work later on in the project.

[170] "*Managing the Development of Large Software Systems*", Royce, W., 1970 explained in Larman, C., 2005, *Agile and Iterative Development – A Manager's guide*, p.103-106
[171] See chapter on Volvo IT

**Agile values**:
Customer collaboration over contract negotiation
**Agile principle 1**:
Our highest priority is to satisfy the customer through early and *continuous delivery* of valuable software
**Agile principle 2**:
*Welcome changing requirements*, even *late in development*.
Agile processes harness change for the customer's competitive advantage.
**Agile principle 10**:
*Simplicity*--the art of maximizing the amount of work not done--is essential.

**Keep up the structure – control chaos**

It is important to remember that everything is interconnected and that an organisation consists of people first and foremost; if employees make bad mistakes or show bad judgements in projects which delays the project, cause problems for the customer, if the project is considered unsuccessful and so on; everything that may cause project members to make mistakes should be understood and also eliminated as far as possible. It is of great importance to be able to be structured if a good and productive job effort should be possible.

Agile methodologies address this issue when pointing out that everyone in a project should be able to maintain a constant pace indefinitely. Every team member should be able to stay alert and creative throughout the projects' complete life-cycle.

**Agile principle 8**:
The sponsors, developers, and users should be able to *maintain a constant pace* indefinitely.

**The intense changing pace**

Agile methodologies are all about being 'agile'; increase flexibility and manoeuvrability for software development projects. With the agile approach it should be possible to welcome change requests even late in a project's life-cycle.

When concluding that the changing pace is intense at Volvo IT and that this is one of the main reasons to why many projects are experiencing problems – it is easy to pose the question whether not aspects of an agile approach could be very effective in the organisation. Exactly which aspects that would be most appropriate to try out on a small scale to be able to evaluate the result from it is something that is recommended for further studies however.

**Agile principle 2**:
*Welcome changing requirements*, even *late in development*.
Agile processes harness change for the customer's competitive advantage.

### The need for sharing information and documentation

It may be easy to prioritise other job duties in the workload other than to read and write documentation, especially if those tasks are accompanied by a belief that it is unnecessary work that no one will read anyway. Interviewees say that documentation is not always emphasised and put into focus by the top management in project organisations either, and it may lead to important information being missed by project members if it is too easy to prioritise other responsibilities. Also, if an employee, at any level, can not trust the information forwarded to be read by other employees, this may cause serious problems; the eternal question of 'how to get the information out to the receiving party and how to make them read it'? With incorrect or outdated information the confusion on what the priorities are will be increased. This can further cause misunderstandings and cause problems in projects.

The agile approach to this is often misunderstood as not having any documentation at all but this is not correct, the unnecessary documentation should be minimised and it is by working closely together that information is best shared.

**Agile principle 6**:
The most efficient and effective method of conveying information to and within a development team is *face-to-face* conversation.
**Agile principle 10**:
*Simplicity*--the art of maximizing the amount of work not done--is essential.

### Lack of resources

The lack of resources to take part in project teams are not addressed by agile since it is a project methodology and the issues regarding the lack of resources is a problem on an organisational level. However it is strongly emphasised that a project should always be based around the – motivated - individuals that form the project team.

**Agile principle 5**:
Build projects around *motivated individuals*. Give them the *environment* and *support* they need, and *trust* them to get the job done.

### Personal development

It is a risk for the whole organisation in the long run if its employees do not have the time to develop and evolve in their working roles; it is a risk for the organisation if its employees do not have the time to learn new techniques and work processes as well as think about and plan their own personal development. Individual development will lead to committed and interested employees that is an advantage for any company - instead of a possible cause for problems - which it can be to not have the necessary competences available inside the own company.

Agile methodologies emphasise personal development saying that skilled individuals working together under good conditions with working communications can overcome any inadequate processes, but the right processes can not make up for inadequate personnel[172]. Rotation in the project group between the team members and their working roles is one way to increase the team members' skills and knowledge.

### The big picture

Unless it is possible to see the big picture and to keep an overview of a work situation it becomes impossible to understand the consequences and foresee the results that it may bring. Without being able to see the possible consequences it becomes impossible to be sure that the work processes that have been decided on are really the correct ones. The project group can also not be sure they are performing their jobs correctly without a view and understanding of the big picture.

Agile methodologies recommend project groups that are self-organising since it is considered that the project group itself will have the best understanding of its own reality and obstacles, but other than that the agile approach do not directly addresses the need for keeping the big picture to know that the right decisions are made.

### The Steering Committee - Are the right decisions made?

If the employees whose job it is to make decisions, for those who are overwhelmed with work in projects, also are overwhelmed with work – is it likely that mistakes will occur. The Steering Committee function is sometimes referred to as a problematic and uninformed group which is mostly interested in numbers and time plans and lacking in understanding of quality issues in software development and detailed knowledge on their decisions. The reality seems to be that the members in the Steering Committees, who are 'normal people' like the ones they make decisions for in projects, are just as overwhelmed with work. Adding up the various statements on what a job description for a Steering Committee member actually may include; be a member on as many as 10-15 or more different Steering Committees for different projects, attend several different committee meetings per week deciding on the various projects, read all information on- and form an informed understanding of *each* project to be able to make the right decisions, alongside all other line responsibilities; it is not surprising that criticism is aimed towards 'uninterested' Steering Committee members that 'do not know what they have decision power on'.

It may be argued that a more agile approach to projects could ease the pressure and expectations on the Steering Committees, if applied in that way, since agile advocate that the best and most productive project teams are the self-organising ones that are left to make its own decisions and make their own reviews on their own work organisation and progress.

In an extremely large organisation like Volvo IT it will probably not be productive if a control function like that of Steering Committees was fully exchanged for having all self-organising teams instead. Such an extreme paradigm shift would probably be causing more problems than it would solve, at least in the short term. However, the idea is interesting and is recommended for further research; the possibilities to incorporate a higher degree of self-organising teams in the project organisation.

[172] Cockburn, A., Highsmith, J., 2001, *Agile Software Development: The People Factor,*

**Agile principle 5**:
Build projects around *motivated individuals*. Give them the *environment* and *support* they need, and *trust* them to get the job done.
**Agile principle 11**:
The best architectures, requirements, and designs emerge from *self-organising teams*.

**Conclusions on Work organisation**

Having too much to do in too little time is a reason that affects many different levels of why problems are created. The sheer fact that people do not have the time to work on solidifying project organisations, not have the time to read or take the time to write documentation, do not have the time to keep themselves or others informed, do not have the time to perform project functions 'by the book' like updating role definitions or the pre-study being dedicated sufficient time frame; will contribute to increased amount of unsuccessful projects and an overworked and inefficient workforce. It is of interest for a whole organisation to keep the workload for its employees at a feasible level since an overwhelming workload is counterproductive in both regarding wellbeing for its employees and its financial results.

Agile methodologies could indirectly have an effect in this problem theme with its emphasis on project teams being most efficiently productive when allowed to be self-organising as far as it is possible and implement the ideas of not having pre-set and too tight deadlines for final deliveries.

### 6.1.6 IT-product Vs. Business Value

There will always be different perspectives represented by the customer and the supplier and they use their own languages with business people traditionally talking about business value and finances and the software development supplier talking about technical solutions and tools. Misunderstandings between the two sides are created from their different perspectives and it causes unsuccessful projects and likely outcomes are software with features that are never used, features that are unnecessary for the customer, software that is not implemented properly or integrated into the existing system that is already in use in the customers' organisation. It is in the interest of both sides to try and bridge the gap between the two. The customer wants good value for its investments and good support from the software for its organisation. The supplier wants successful projects with a happy customer that wants to come back for future projects.

**The understanding of the scope**
It may lead to practical collaborating issues in the project organisation with problems in communicating and understanding each other when two sides work for a common solution but with different views on what the focus really is. However, with an extended and well developed understanding of the customer's needs and business may the supplier be able to minimise those misunderstandings. That however presupposes that a fundamental understanding of the customer's business is possible to develop without being a part of the actual organisation.

Agile methodologies use practices that are designed to support flexibility and make it possible to incorporate differences into a software development project, even late in the process. This is one of the core ideas with agile, the environment surrounding a software development project is constantly changing, and therefore should the project team be able to meet these changing requirements. One approach to handle the problem with conflicting opinions on what is included in the scope is to deliberately not freeze the scope, but instead make sure it is abstract in the beginning with the intention of having it change several times in the project. The time and cost can be set variables but not the features – this is another agile approach. One main idea with agile methods is that there is no better way to understand the customer than to work *with* the customer, and allowing the scope and design of the product evolve over the process which is the project.

In a large, global organisation like Volvo IT it may be a difficult to adopt the agile aspects of a less detailed scope, mostly everyone seems to agree that it instead should be more detailed before the projects even starts. These are two opposite views of how to address the problem.
Another agile approach that may be 'easier' for a very large traditional software development company to adopt is the practice of several small iterative deliveries to the customer. Actually seeing the product that has been developed at a certain phase will help the customer and the supplier to understand whether they are seeing the project from the same view or not.

**Agile values**:
Individuals and interactions over processes and tools
Customer collaboration over contract negotiation
**Agile principle 1**:
Our highest priority is to satisfy the customer through early and
*continuous delivery* of valuable software.
**Agile principle 2**:
*Welcome changing requirements*, even *late in development*.
Agile processes harness change for the customer's competitive advantage.
**Agile principle 4**:
Business people and developers must *work together daily* throughout the project
**Agile principle 6**:
The most efficient and effective method of conveying information to and within a development team is *face-to-face* conversation.

### Shift of focus

How can a shift of the focus be created within the supplier's organisation? Can it be achieved without a closer working relationship with the customer? The question is open to be debated but the agile approach's answer to it is that the best and most effective form of communication to create an understanding is face-to-face communication and a close every day-to-day communication with the customer. The focus should be on intercommunication between different teams and not on the technical product; the technical excellence is important to create agility in the product and to make the project team able to respond to changing requirements but without having a correct understanding of what the customer wants, all technical expertise may be in vain if the wrong product is being developed. Agile methods emphasise the importance of having a common vision and use practices to make sure that vision is understood and agreed upon by everyone involved in a project. If the projects' members of the software development team have the same vision as its customer the project is much more likely to succeed[173].

**Agile principle 1**:
Our highest priority is to satisfy the customer through early and
*continuous delivery* of valuable software.
**Agile principle 2**:
*Welcome changing requirements*, even *late in development*.
Agile processes harness change for the customer's competitive advantage.
**Agile principle 4**:
Business people and developers must *work together daily* throughout the project

### Sell the product

If the product is not implemented properly into the customers' organisation it is not likely to be viewed as a successful project no matter how technically excellent it may be. Even though this seems to be the common understanding and view among the interviewees and the respondents of the survey, it is also their testimony that this is often

[173] Highsmith, J., 2002, *Agile Software Development Ecosystems*,

overlooked.

Agile methodologies preach flexibility of the software development projects's technical product and an understanding of the customer and the change the product will have on the customer's organisation. This is believed to effectively be achieved only by continuously deliver working and valuable software to the customer and work *with* the customer in the development projects. By having the customer more involved will the product be able to grow with the customer and the customer's organisation and no features will come as surprises to the customer or not be useful. The need for the product should also be better understood within the customer's organisation.

**Agile principle 1**:
Our highest priority is to satisfy the customer through early and
*continuous delivery* of valuable software.
**Agile principle 2**:
*Welcome changing requirements*, even *late in development*.
Agile processes harness change for the customer's competitive advantage.
**Agile principle 3**:
Deliver working software frequently, from *a couple of weeks* to *a couple of months*,
with a preference to the shorter timescale.
**Agile principle 4**:
Business people and developers must *work together daily* throughout the project

**The time aspect**
Requirements will change, even more so over a longer period of time. The projects at Volvo IT are almost all global and all very large in size, they span a long period of time and they are expected to produce products that should last and be maintained in the organisations for a lengthy period of time after the project has ended. Most of the software products must accommodate several organisations with diverse requirements and the requirements due to their multitude and due to the length of the projects are likely to change many times. The problem of monitoring change requests are brought up by the interviewees as something that is not successful in IT-projects at Volvo IT. Additional requests or changing requirements are seen as problematic and a cause for problems with lengthy and formal decision routines.

As mentioned in previous themes; the agile approach to welcome change instead of hoping that it will not happen and to promote flexibility seems almost perfect in this aspect. However agile promotes small teams in contrast to large which are more difficult to manoeuvre and because the agile team is built around the individuals that it consists of, and large teams are more likely to have a significant staff turnover. Can there be smaller development teams at Volvo IT? Can large projects be broken down into smaller projects? This is one aspect suggested for further research.

**Agile principle 1**:
Our highest priority is to satisfy the customer through early and
*continuous delivery* of valuable software.
**Agile principle 2**:
*Welcome changing requirements*, even *late in development*.
Agile processes harness change for the customer's competitive advantage.
**Agile principle 3**:
Deliver working software frequently, from *a couple of weeks* to *a couple of months*,
with a preference to the shorter timescale.
**Agile principle 5**:
Build projects around *motivated individuals*. Give them the *environment*
and *support* they need, and *trust* them to get the job done.
**Agile principle 7**:
*Working software* is the primary measure of progress.
Agile processes promote sustainable development.
**Agile principle 11**:
The best architectures, requirements, and designs emerge from *self-organizing teams*.

### Conclusions on IT-product vs. Business value

The need for creating a common vision between the customer and the supplier is expressed in the interviews and in the survey. It is a theme that almost transcends through all the other themes; the problems explained in the Being Global theme can be argued to be caused by not fully understanding that there is a difference between the customer and the supplier that needs to be addressed and amended. Organisational legacy also explains problems caused when the difference between the two sides of customer and supplier perspectives are not fully understood. Line vs. Projects explain problems with having an organisational structure that not support the project based working practices and through that do not support the projects the organisation is built around. Not having the Role Definitions clearly stated will create misunderstandings on both sides and it can be argued to not understand that the customer has one focus – the business value of the product and the product, the customer's focus will never be regarding the functions of a project team and to understand who is doing what. To not have that clearly stated is to ignore that the focus should at all times be the finished product. The theme regarding Work Organisation describe problems of an overwhelming workload which complicates the execution of work duties in a sufficient manner, that is to say that if the work practices are not supported by the organisation the execution of work duties cannot be performed efficiently and the projects that involve customers are negatively affected. An understanding of the importance of having a common vision can therefore be argued to be desired both on a project team level but also to be supported by the overall organisation.

## 6.2 Conclusion on the discussion and analysis chapter

There is talk about how the company should evolve and become more of a service and customer oriented company[174]; these are the intentions the interviewees talk of as the company goals; and the company information material on the company intranet does the same saying that the customer should always be in focus. However, from the gathered material for this thesis there is serious criticism aimed towards the organisation that can be interpreted as these intentions have not been successful this far. The intentions seem to be more on paper in many cases since the work practice has not been changed to accommodate, support and follow-up the new viewpoint and new focus of the company.

**Can agile methodologies make a difference? -** Agile methodologies aim is to move away from the view that humans are predictable machines and at the same time accept that the world we live in is changing and unpredictable too, saying that if someone tries to manage uncertainty with methodologies that are based on practices that demand certainty to be effective, the battle is already lost. One of the core values in agile methodologies is that **project teams** should preferably be **as small as** possible and physically based in the same room in order for communication to be flexible, in real-time and through this minimise the need for extensive documentation. Agile focuses on the result instead of appropriately checking off the 'to do'-list that leads towards the result.

This is a direct contradiction to the standard traditional work practices in most traditional companies. A large company relies on **detailed documentation** for safety reasons as well as for transferring information to new project members and project reviews (to the extent these are made). The traditional project organisation is based on having a detailed plan for every anticipated progress, in Volvo IT's case there are detailed descriptions and check lists of everything that should be done before each new gate in a project, and meetings with the steering committees are planned regularly before each gate, since it is the steering committee that decides whether a project is ready to be left to pass a new gate and continue into the next phase in the project's life-cycle.

This radical fundamental difference between agile and traditional viewpoint of project methodologies needs to be understood since it is affecting all areas of how project practices are designed and conducted if an agile approach was decided as desirable to adopt.

[174] See chapter on The setting, Long term focus areas

# 7 Conclusion

*Conclusions for this master thesis including a brief overview on the problem themes.*

The purpose for this thesis was to **investigate whether agile methodologies are likely to effectively support large IT development organisations in their IT projects**, even though the methodologies were originally developed for small-scale projects. It seems that the need for flexibility and to be able to respond to changing requirements, answer to a turbulent environment and at the same time produce high quality software are the same challenges that both large and small software development teams faces, which implies that large organisations unquestionably need a greater portion of the agility in their projects that agile methodologies can create. However the projects in large organisations are surrounded by many more external control functions because of their size and the complexity of large matrix organisations. This makes new approaches and work methodologies more complicated to implement. The question to follow the initial objective is then; **how** can they effectively support large IT projects? The greatest problem seems to be regarding integrating the agile work practices into traditional organisations with standardised processes. The six different problem themes addressing a wide variety of problems in IT projects summarised with suggestions on agile practices to address them:

Problems related to **organisational structure**

| Theme | Problem | Agile approach |
|---|---|---|
| **Being a Global** company | - Communication<br>- Collaboration | ▪ Split large projects into *smaller part-projects* with part-deliveries into a larger project, continuous integration |
| **Line vs. Project** responsibilities | - Prioritising the workflow | ▪ Emphasise the importance of having *dedicated* and *motivated* individuals on the team from the start and allow them to work in…<br>▪ Self-organising teams |
| **Work Organisation** with tight deadlines | - Prioritising the workflow | ▪ Focus on *iterative deliveries* and the *result* (instead of the scope)<br>▪ Minimise the unnecessary work that is done (software features, documentation and more that never will be used, focus on what is absolute necessary)<br>▪ Increase *informal & effective* information sharing (cut back on formal information sharing) |

| | | |
|---|---|---|

Problems related to **work processes**

| Theme | Problem | Agile approach |
|---|---|---|
| Having an **Organisational Legacy** and history with the customer | - Communication<br>- Misunderstandings from (un)professional attitudes | ▪ Involve *the customer as a team member* in the project on an operational level to improve actual mutual understanding<br>▪ Technical stability through *continuous testing* and integration (addressing the technical history) |
| **Role Definitions** not updated for each project | - Communication<br>- Misunderstandings regarding working responsibilities | ▪ Role definitions specified for *each new project*, continuously updated<br>▪ The *understanding* of the different roles among *all* team members is emphasised |
| **IT product vs. Business Value** for the customer, an historical legacy | - Different vision/focus customer – supplier | ▪ Agree on a shared *vision* with the customer at the beginning of a project, communicate it to all stakeholders<br>▪ *Support the vision* throughout the project life-cycle and work practices<br>▪ Arrange close *working* relationships with the customer<br>▪ View projects as *a joint effort* between customer-supplier<br>▪ Frequent (daily) communication<br>▪ Iterative deliveries of software to the customer that actually *show* the project's progress |

The themes regarding **Work Organisation** and **IT product vs. Business Value** are the areas that most correspond to an agile approach. Agile directly addresses the problem of rapid change which seems to be the biggest issue causing problems in the Work Organisation theme. To address the rapid changes in the environment and consequently changing requirements are work practices implemented to create an understanding of the customer's business through joint and close work practices together with the customer alongside the iterative approach to development work where both sides can see the reality of what is being produced.

It can be argued from the results of this thesis that agile methodologies are indeed likely to effectively support large software development projects and that there are several work practices that can be implemented to increase the amount of agility in large IT projects.

# 8 Agile work practices of interest for Volvo IT

*This chapter gives a short summary on suggestions of agile practices that may be interesting for Volvo IT's project organisations.*

Presented here are some suggestions on agile project practices that may be of interest for Volvo IT to adopt, these suggestions should be viewed as suggestions only, and not be seen as direct recommendations to implement without further research into to how best to implement them in detailed project circumstances:

**Communication within project teams**

- **15-30 minute daily informal stand-up meetings** in accordance with Scrum. Team members on other sites may be linked/connected to the meeting in real-time via video or telephone. (to improve effective communication)
- **Rotation in project roles,** where/when appropriate. (to further individual development and skills and to minimise the effect the loss of a team member could have on a project).
- **Pair programming,** where/when appropriate (to further individual development and skills; improve communication and knowledge sharing and to minimise the effect the loss of a team member could have on a project and give programmers increased confidence in the design of the code; minimise mistakes and gold-plating in code)
- **Collocate teams physically** where/when possible, if not every time and for every project (not likely to be doable) at least during a number of phases in the project's life-cycle (to facilitate informal and effective communication and sharing of knowledge; preferably including one or more customer representatives that take active part in the project)

**Technical aspects**

- **Continuous testing of code** throughout the development process (to facilitate implementation of changes to the product late in the process)
- Apply an **iterative development approach** (to facilitate implementing changes late in the process, also to continuously have something tangible to show the customer throughout the development process; progress that is possible to *see*; i.e. working/valuable code)

**Documentation**

- Address the issue of what is really **useful and necessary documentation** for each individual project (to minimise unnecessary work efforts; *which documentation is actually likely to be read and used?*)
- **Less detailed documentation,** where/when appropriate (to minimise the need to update and re-write documentation as changes are implemented).

**Communicate a change of attitude**

- **Change the terminology** used in gate meetings to resemble that of agile methodologies. For example, name gate meetings *part-deliveries*, *first customer delivery* and so on.
- Consider a change of **the content that is delivered** at gate meetings/*projects' part-delivery*. Present working code and prototypes that *show the actual progress* in the project at a certain point, to a customer that is fully aware of the end-user's needs, and be welcoming feedback and understanding that it usually means changes to the product. Even smaller iterative *releases of the product* to the customer, where/when possible (to create better value for the customer).

**Organisational level - Provide a setting that allows agile methodologies to function**;

- Consider contract negotiations on time and cost, **leaving out specified features** and instead specify **a common vision** for the project and the communication and collaboration for the team *together with the customer*. Draw ideas on contract negotiation from venture capitalists' contracts where the outcome often is as flexible as in software development projects; *incentives and contracts should be arranged so that all parties involved in a deal share the high inherent degree of risk, so that all are encouraged to take actions that are in the best interest of the whole project* [175].
- **Choose control functions for a project carefully** and economically, (realising that many external control functions which the project team have to report to, decreases the level of agility that the project methods are aiming to increase).

**Information - understanding**

Any new practices should be preceded by information and an *understanding* of the new approach since practices that are perceived to be forced upon anyone, no matter how effective the practices may be, are not likely to be successful.

**Higher management**

Primary efforts should be aimed at higher management since any approach that does not have the attention of, or are fully understood by, higher management are likely to not receive sufficient support or have the right conditions provided under which it can be successful and/or properly evaluated.

[175] Highsmith, J., 2002, *Agile Software Development Ecosystems*, , p.75

# 9 References

## Books and articles:

Backman, J. (1998). *Rapporter och uppsatser.* Lund: Studentlitteratur

Brooks, F. (1975). *The Mythical Man Monthr.* Reading, Mass.: Addison-Wesley

Cockburn, A., Highsmith, J. (2001). Agile Software Development: The People Factor. *Computer, volume: 34, issue: 11,* 131-133

Coram, M., Bohner, S. (2005). The impact of agile methods on software project management. *Engineering of Computer-Based Systems. ECBS '05. 12th IEEE International Conference.*

Dourish, P., (2006). *Implications For Design*. Dept. of Informatics, Donale Bren Schooll of Computer Sciences, University of California, Irvine

Easterby-Smith, M., Thorpe, R., Lowe, A. (2002*). Management Research – an Introduction.* Trowbridge, Wiltshire: The Cromwell Press Ltd.

Highsmith, J. (2002). *Agile Software Development Ecosystems.* Boston: Addison-Wesley

Larman, C. (2005). *Agile and iterative Development – A manager's guide.* Boston: Addison-Wesley

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., Kahkonen, T. (2004). Agile software development in large organizations. *Computer, volume: 37, issue: 12,* 26-34

Lycett, M., Macredie, R.D., Patel, C., Paul, R.J. (2003). Migrating agile methods to standardized development practice. *Computer, volume: 36, issue: 6,* 79-85

Marin, A., Biddle, R., Noble, J. (2003). How do XP, Scrum and ASD build the right software? *Position paper for a workshop at: OOPSLA 2003*[176]

Skowronski, V. (2004). Do Agile Methods Marginalize Problem Solvers? *Computer, volume: 37, issue: 10,* 120-119

[176] OOPSLA = annual conference on *Object-Oriented, Programming, Systems, Languages and Applications* organised by *The Association for Computing Machinery* (ACM), founded in 1947 as the world's first scientific and educational computing society. http://www.oopsla.org/, http://en.wikipedia.org/wiki/OOPSLA, http://www.acm.org/, http://en.wikipedia.org/wiki/Association_for_Computing_Machinery

The Agile Alliance
http://www.agilealliance.com/
[last visited 20060524]

Manifesto for Agile Software Development,
http://agilemanifesto.org/
[last visited 20060524]

Agile Modelling, Scott Ambler
http://www.agilemodeling.com/
[last visited 20070101]

Agile Sweden
http://www.agilesweden.org/metoder.htm
[last visited 20060524]

Ken Schwaber web page, Scrum
http://www.controlchaos.com
[last visited 20070102]

Dr. Dobb's Journal, The World of Software Development
http://www.ddj.com/
[last visited 20070103]

IT-Barometern
http://www.itbarometern.se
[last visited 20070105]

Martin Fowler web page
http://www.martinfowler.com/
[last visited 20070101]

Projektplatsen
http://www.projektplatsen.se
[last visited 20070105]

Refactoring
http://www.refactoring.com/
[last visited 20070101]

The Standish Group (The CHAOS Report)
http://www.standishgroup.com/index.php
http://www.pm2go.com
[last visited 20070109]

Volvo Group
www.volvo.com
[last visited 20060523]

# 10 Appendix

## 10.1 Interview questions

**Interview** ***- presentation and questions –notes***

**Presentation**
M.Sc of Informatics student at Gothenburg University School of Business Economics and Law / IT-University Gothenburg. Master thesis, spring semester 2006.

**Subject:**
Try to understand *what* it is that makes a successful project *successful…*
What actually *happens* when a project is *problematic* or *unsuccessful*?
"…what actually goes wrong when something goes wrong?"

Anonymous. No names.
Okay to record the interview?

A few structured, set, questions, otherwise an open discussion.
Personal experiences and opinions are what I am interested in primarily.

---

**Questions.**

**Background**

For which **department** do you work?
**For how long** have you been working for that department?
**For how long** have you been working for **the Volvo Group**?
What is the **title** of your working role?
What do you do in your job?
What is your **role** in IT-project?

**About projects.**

What do you consider to be a **successful** project?
What do you consider to be an **unsuccessful** project?
**What** are the most common problems?
**Who** is involved in those problems?
**When** do the problems occur? **Where** in the project plan?
**Why** do they arise then/there?
**What** causes them?
**How** can they be **avoided**?
Can anything be **changed** today to counteract them from occurring?
If not anything can be changed right now – what is **needed** in order to make that change?
How many projects could you be working on at the most?
Do you think that there is anything missing in the **project models** today?

## *10.2 Survey introduction*

Project Management

# Project management

| | |
|---|---|
| Survey Name: | Project management |
| Survey Description: | About opinions and experiences on project management. This survey is part of a research study about what the most common problems are in IT-projects, in order to design a solution or possible changes in work methods in projects. It should take no more than 5-10 minutes to complete. Respondents are Volvo employees at various companies who have worked in IT-projects. |
| Time Created: | 2006-02-28 12:57 |
| Number of Responses: | 29 |

## *10.3 Survey template*

Project Management

# Project management: New Item

1. Which department do you work for at Volvo? *

2. For how long have you worked in that department? *

○ <1 yr
○ 1-5 ys
○ 6-15 ys
○ >16 ys

3. How long have you been working for Volvo? *

○ <1 yr
○ 1-5 ys
○ 6-15 ys
○ >16 ys

4. What is your job title? *

○ Project manager
○ Business project manager
○ Business analyst
○ Business consultant
○ Group manager
○ System analyst
○ Specify your own value:

5. Have you worked in IT-projects at Volvo? *

○ Yes
○ No

6. Have you worked with IS-GDP as a project model? *

○ Yes
○ No

7. Any other project models you have worked with at Volvo?

8. Is anything missing in the project models you have worked with?

9. Do you agree or disagree with these statements?
Pls indicate below.

| | I completely agree | | Average | | I completely disagree | **N/A** |
|---|---|---|---|---|---|---|
| 1. The communication between Volvo IT and the customer is always professional. | ○ | ○ | ○ | ○ | ○ | ○ |
| 2. The working roles are well understood in projects, everyone knows what to do, on the business side as well as the IT side. | ○ | ○ | ○ | ○ | ○ | ○ |
| 3. The customer understands that additional requests may have an effect on the project in time and cost. | ○ | ○ | ○ | ○ | ○ | ○ |
| 4. Volvo IT always informs the customer when additional requests in projects may delay the project's final delivery. | ○ | ○ | ○ | ○ | ○ | ○ |
| 5. Volvo IT may add functions in systems without informing the customer first. | ○ | ○ | ○ | ○ | ○ | ○ |
| 6. The initial scope of projects are detailed and well defined. | ○ | ○ | ○ | ○ | ○ | ○ |
| 7. Volvo IT has a good understanding of the customer's business. | ○ | ○ | ○ | ○ | ○ | ○ |
| 8. The customer has a good understanding of IT and can communicate well with Volvo IT. | ○ | ○ | ○ | ○ | ○ | ○ |

10. How can the communication between the customer and Volvo IT be improved?

11. What is the MOST important for you in a successful project? *

- ○ That the project delivers on time.
- ○ That the project meets its scope for estimated time and cost.
- ○ That the customer is happy with the final product.
- ○ That the final product is actually used by the customer and its end-users.
- ○ That the final product looks like its initial scope and hasn't changed too much.
- ○ That everyone get along well with each other in the project team.
- ○ That you have learned something valuable from working on the project.
- ○ Specify your own value:

12. Have you worked on a successful IT-project? *

- ○ Yes
- ○ No

13. If Yes; What do you think MOSTLY made it successful?

- ○ The pre-study.
- ○ The list of demands on the system from the customer.
- ○ The communication with the customer.
- ○ The communication within the project group.
- ○ The project manager.
- ○ The planning of resources.
- ○ Specify your own value:

14. Have you worked on an unsuccessful IT-project? *

- ○ Yes
- ○ No

15. If Yes; What do you think MOSTLY made it unsuccessful?

- ○ The pre-study.
- ○ The list of demands on the system from the customer.
- ○ The communication with the customer.
- ○ The communication within the project group.
- ○ The project manager.
- ○ The planning of resources.
- ○ Specify your own value:

16. What causes MOST problems in IT-projects?

- ○ The pre-study.
- ○ The list of demands on the system from the customer.
- ○ The communication with the customer.
- ○ The communication within the project group.
- ○ The project manager.
- ○ The planning of resources.
- ○ Specify your own value:

17. Why do YOU think these problems occur?

18. Who/What causes these problems?
(A specific work title? A phenomenon like, everyone's workload is too big?)

19. What can be done TODAY to avoid or solve some of these problems?

20. Can any of the following help to avoid or solve some of the problems in IT-projects?
(multiple answers possible)

○ A more detailed pre-study.
○ More emphasis on producing a correct and detailed scope of the project with the customer.
○ If the communication with the customer is changed and improved.
○ If the communication within the project group is changed and improved.
○ By actions taken by the project manager.
○ With the planning of resources.
○ Specify your own value:

21. If you would like to comment on the previous question:

22. Do you have access to enough information about your IT-projects,
to allow you to do a good job? Or what would you like to change?

23. If you have a comment or anything to add to your responses:

* indicates a required field