Ilari Pulkkinen

Unifying the Process for Requirements Gathering in Multi-site Organizations: A Case Study with Interface Development Tools

School of Science

Master’s Thesis which have been submitted for evaluation
for Prof. Marko Nieminen in Espoo 30.09.2011

Supervisor:

Prof. Marko Nieminen

Instructor:

M.Sc (Tech) Mika Luotojärvi

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Ilari Pulkkinen

Title: Unifying the Process for Requirements Engineering in Multi-site Organizations: A Case Study with Interface Development Tools

Date: 30.09.2011 Language: English Number of pages: 7 + 98

Department of computer Science and Engineering
Software Engineering and Business Code: T-121

Supervisor: Prof. Marko Nieminen
Instructor: M.Sc. (Tech) Mika Luotojärvi

Requirements gathering is challenging as it is but it becomes even more challenging when it is done in a multi-site organization consisting of several business units having their own specialized needs.

This work is based on a requirements gathering project done for CPM (Collaborative Production Management) -technology R&D unit of ABB Process Automation Division. The goal of this project was to find out if it is possible to unify CPM user interface development efforts throughout all the business units of ABB Process Automation Division. The goal of this work on the other hand was to form a process that could answer several challenges in requirements gathering. In the beginning of the work challenges for requirements gathering presented in literature are covered. After that a process is presented that was developed and used in the requirements gathering project. In the end there is an analysis about how the presented process met the challenges presented in the beginning of the work.

The results reveal several critical factors about how the process helps to manage large amounts of highly diverse requirements and gather them efficiently. The used mind map framework and prioritization scheme help to cope with the diversity. Furthermore the context specific brainstorming workshops and ULCR (User-led Construction of Requirements) -methodology result in correct and relevant requirements. And finally the moderator control and having domain experts in the team ensure efficient workshops.

Keywords: Requirements, user interface development tool, workshop, process automation, multi-site organizations

AALTO-YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Ilari Pulkkinen

Työn nimi: Vaatimusmäärittelyprosessin Yhtenäistäminen Hajautetussa Organisaatiossa: Tapaustutkimus Käyttöliittymäkehitystyökaluilla

Päivämäärä: 30.09.2011 Kieli: Englanti Sivumäärä: 7 + 98

Tietotekniikan laitos

Käytettävyys Koodi: T-121

Valvoja: Prof. Marko Nieminen

Ohjaaja: DI Mika Luotojärvi

Vaatimusmäärittelyn tekeminen on haastavaa jo sinänsä, mutta siitä tulee vielä haastavampaa, kun se tehdään hajautetussa organisaatiossa koostuen useista eri liiketoimintayksiköstä joilla jokaisella on omat erityistarpeensa.

Tämä työ perustuu vaatimusmäärittelyjenkeräämisprojektiin, joka tehtiin ABB:n Prosessiautomaatiodivisioonan CPM (Collaborative Production Management) -teknologian T&K -yksikölle. Tämän projektin tavoitteena oli selvittää, miten voidaan yhtenäistää CPM -käyttöliittymäkehitystoimintaa ABB:n prosessiautomaatio-liiketoimintayksikössä. Toisaalta tämän työn tavoite oli muodostaa prosessi, joka vastaa useisiin vaatimusmäärittelyjen keräämiseen liittyviin haasteisiin hajautetuissa organisaatioissa. Työn alussa on kirjallisuuskatsaus, jossa esitellään haasteita vaatimusmäärittelyjen keräämiselle. Sen jälkeen esitellään prosessi, joka kehitettiin ja jota käytettiin vaatimusmäärittelyjen keräämisprojektissa. Lopussa analysoidaan, kuinka esitetty prosessi vastasi alussa esitettyihin haasteisiin.

Tulokset paljastavat useita kriittisiä tekijöitä miksi valittu prosessi auttaa hallitsemaan suurta määrää hyvin erilaisia vaatimuksia, sekä keräämään ne tehokkaasti. Käytetty miellekarttakehys, sekä priorisointijärjestelmä auttavat käsittelemään erilaisia vaatimuksia. Lisäksi kontekstispesifiset aivoriihityöpajat, sekä ULCR (User-led Construction of Requirements) -metodiikka johtavat oikeisiin ja oleellisiin vaatimuksiin. Sen lisäksi moderaattorien kontrolli, sekä domain-asiantuntijoiden mukanaolo tiimissä takaavat tehokkaan työpajatyöskentelyn.

Avainsanat: Vaatimukset, käyttöliittymäkehitystyökalu, workshop, prosessiautomaatio, hajautetut organisaatiot

Foreword

This Master's thesis has been carried out as a part of requirements gathering project done for CPM technology R&D unit of ABB Process Automation Division located in Pitäjänmäki in Finland.

I'm grateful for the cooperation and support received from my colleagues in Process Automation unit located in Pitäjänmäki. I want to especially thank my instructor Mika Luotojärvi for his guidance and readiness to offers his expertise to aid my work. I would also like to thank Simo Säynevirta for his useful feedback and support I received when I was working with my thesis at ABB Pitäjänmäki. I also want to thank my supervisor Marko Nieminen for useful advice I got in the beginning of my work.

I would also like to thank Kim Bergman and Eljas Perheentupa from Linja for cooperation in the requirements gathering project to which this work is based on. Furthermore I want to thank all the people from different CPM technology units of ABB Process Automation Division who took part in this requirements gathering project.

I'm also grateful for the support I received from my friends and my family.

Espoo, 30.09.2011

Ilari Pulkkinen

Content

Abbreviations and terms

Requirement = A requirement is a necessary attribute in a system, a statement that identifies capability, characteristic, or quality factor of a system in order for it to have value and utility to a customer or user. [1]

Requirements gathering = It must be mentioned that in this work the term requirements gathering consist of requirements elicitation, validation, prioritization, organization, and analysis.

Business unit = Business Unit or BU refers to a global unit of a division providing products and services for some particular industry worldwide, facility or department of an organization. Operations in ABB are divided in divisions and further to several business units.

Process automation system = A process control or automation system is used to automatically control a process such as pulp & paper factories, oil refineries, and mineral processing operations. The process automation systems uses network to interconnect different sensors, controllers, terminals, and systems.

User interface development tool = Consist of tools for creating graphical user interfaces as well as support for adding functionality to the interface with a programming language of choice

CPM = Collaborative Production Management solutions integrate all manufacturing systems, providing the real-time visibility, execution, tracking, reporting, and optimizing of manufacturing processes. [2]

cpmPlus History = CpmPlus History consists of data collector, Main cpmPlus History Application Server and Visualization client. Data collector node collects data from control systems in customer sites and forwards the data to cpmPlus History Application Server. Main cpmPlus History Application server stores the data which is then visualized by Visualization Clients presented in section 3.2. [3]

Thin client = A computer program which depends heavily on the server to perform its operations [4]

The Case Company = Refers to CPM operations in different industry specific Business Units of ABB Process Automation Divisions

1. Introduction

Before any product or service can be built there must be an understanding what is really needed and what kind of problems are there to be solved to meet this need [5]. This understanding is stored in requirements. Requirements provide the basis for all of the development work that follows such as system design, development, testing, implementation and operation [1]. When a product is being built in a small project by small company with a small customer base things are quite straight forward: changes for conflicting requirements, hard trade-offs, wrong requirements, are quite small. Things start to get quite problematic when we are considering a big international highly distributed organization. This creates major challenges in coordination and communication of requirements gathering activities [6][7][8]. In addition the in the project to which this work is based on the goal was not simply to gather requirements in order to build a specific product. Instead it was about finding out if it is possible to unify the process automation user interface development efforts inside a highly distributed company (the reasoning for this goal is further explained in section 1.1). This creates additional challenges since in this case it is very unsure what is to be expected and what is actually relevant since the way how process automation user interface operations can be unified is determined by the gathered requirements. This is the case that this work is dealing with based on a requirements gathering project conducted for the Case Company.

First in this introduction part there is a short description of the Case Company for whom the requirements gathering project is conducted to. Also the reason for doing the requirements gathering project is explained in more detail. After that the structure of the thesis is presented showing how the work will proceed. The work starts of by presenting challenges in requirements gathering, then a process to meet these challenges is presented, and finally there is analysis about how the presented process met the challenges presented in the beginning of this thesis.

1.1.Research background

Since the requirements gathering project that is in the focus of this work was done for the Case Company a short description of ABB and the unit for which the work was conducted is given. First of all ABB is a global leader in power and automation technologies that enable utility and industry customers to improve their performance while lowering environmental impact. The ABB Group of companies operates in around 100 countries and employs about 124,000 people. ABB consist of 5 divisions Power Products, Power Systems, Discrete Automation and Motion, Low Voltage Products and Process Automation. The unit that I was working for was the CPM technology R&D which is a part of the Process Automation Division. [10]

The operations of CPM unit revolve around collaborative production management solutions which are used to integrate all manufacturing systems, providing real-time visibility, execution, tracking, reporting, and optimizing of manufacturing processes. [11]. The benefit of these solutions for customers representing diverse industries is decreased cost through better production scheduling, execution and management, and increased revenues through improved customer service and production quality. Ultimately this leads to increased profits and enhanced shareholder value. In practice CPM solutions helps customers to connect systems and visualize information from several other manufacturing operation systems that exist inside the manufacturing environment. [3]

The current state in the Case Company is that their operations are distributed in industry specific business units that have their own application portfolio for CPM. All the business units utilize the same back-end, i.e. control systems and process historians, for their applications. The control systems are used to gather control information in customer sites for the use of the applications. The historians are used in industrial and manufacturing environments to record and report on historical data over time, which can include process and product information, as well as network performance and IT monitoring. The front-end on the other hand, i.e. the visualization part of the interface, is different and typically purpose built for each industry.

This requirements gathering project was conducted to find out whether a common tool for visualization for different applications can be created and also to reduce the costs for overlapping developments. To motivate this requirements gathering project even further several challenges due to distributive nature of the Case Company are presented. The first problem is that sharing innovations between business units is hard since technologies used to develop visualization of applications are different and therefore implementing and integrating solutions across business units becomes virtually impossible. The second point is that most of the business units operations are too small scale to develop and maintain their own world class user interface development tools. If all the development efforts would be unified also then a unified support could be offered. In practice it means that application engineers could focus more clearly on efforts that create value for customer without having to worry about for example technology being up to do date, security solutions or interoperability. The third point is that sales people in the Case Company have difficulties offering full CPM portfolio of services for customers since they are not fully aware of what can be actually offered due to the complexity of the Case Company solutions. Because of this the Case Company can't reach its true business potential in the CPM area. The fourth point is that customers do not exactly know what the Case Company can offer to them. Customers do not fully understand the benefits of new technological innovations that are continuously happening inside the Case Company.

In order to make it possible to unify the operations in the Case Company there should be a clear vision of the end result. In order to find this vision a more radical approach for requirements gathering must be taken. If the change would be done incrementally it is very unlikely that the goal of unifying the operations would be reached. One reason for this is that there is constant change in the ways of operating due to changes in customer demand as well as changes in technology. Therefore the incremental nature of the changes will lead to different end results in separate business units of the Case Company and unity of operations would not be reached.

1.2.Structure of the Thesis

This work starts by presenting several challenges in requirements engineering. These are challenges caused by having a multi-site organization, challenges for participants in requirements gathering, challenges due to human factors, and challenges due to the nature of requirements. After presenting overall challenges from the literature, there is a high-level presentation about the domain in which the requirements are gathered. This domain presentation consists of presenting different user interface development tools used in the Case Company as well as presenting different business units of the Case Company that were involved in the study.

After presenting the challenges for requirements gathering the process for gathering requirements is presented. The requirements gathering process consist of methods used for requirements elicitation, prioritizing the requirements, validating the requirements, organizing and combining the requirements, and analyzing the requirements. This part explains why these methods were chosen and how they were used.

In the final part of this work the process for requirements gathering is analyzed against the challenges in requirements gathering presented in the beginning of this work. There it is explained how this process met the challenges, what challenges were unresolved and what should have been done to these unresolved challenges.

2. Goal of the work and research questions

The focus of this research was to find out how different the needs of different business units of the Case Company actually are. The focus was on the interface and the functionality of different user interface development tools used in the Case Company as well as on the business requirements.

The ultimate goal was to find out if it is possible, based on the results, to unify the operations involved around application development inside the Case Company. The idea was to gather the requirements from different business units and then see how similar they were. Based on the level of similarity it is possible to define how easily the application development efforts in the Case Company could be unified.

The main research questions for the work are:

- How to gather requirements from very different business units in a way that they are still comparable with each other?
- What kind of process for gathering requirements should be designed to meet the challenges in requirements gathering in multi-site organization?
- How to cope with very limited time in conducting the requirements gathering project?

3. Challenges in requirements engineering

First in this chapter the challenges in requirements gathering based on literature review are presented. After that the background of the Case Company is presented. This consists of descriptions of 3 different user interface development tools used in the Case Company as well as short descriptions of different CPM business units of ABB.

3.1.Challenges in requirements gathering in a multi-site organization

In this section several challenges in requirements gathering are presented. These challenges are caused by the complexity of multi-site organizations, user-developer interaction, limitations in human cognitive processing, and due to the difficulty in documenting the requirements.

3.1.1. Challenges due to multi-site organization in requirements gathering

The works [6][7][8] present several problems in requirements gathering in multi.-site organizations. After presenting the main challenges these challenges are decomposed and the remaining parts are analyzed individually.

One of the problems in requirements gathering is **inadequate communication**. The biggest reason for inadequate communication is distance which causes face-to-face communication to become very rare and therefore dependency on synchronous and asynchronous communication tools such as phone and email increase. Because of this the quality of communication deteriorates [6] which can lead to unclear requirements [7].

Another problem is dealing with **knowledge management**. In requirements gathering projects conducted in multi-site organizations the sheer quantity of the requirements gathered from multiple sources creates great challenges for knowledge management. The diversity of information sources poses challenges especially for sharing the information. When information from one business unit is gathered only a small portion of the information can be written down since otherwise the task would be too exhaustive. Therefore a lot of valuable information about the context is not documented. Instead this information is only stored in participants' minds. Because of this it is almost impossible for people outside the requirements gathering team to understand the true meaning of requirements. [6] In addition the organization has to cope up with many barriers for requirements gathering including language, cultural, social, and legal barriers making the knowledge management even harder. [7]

Yet another problem affecting requirements gathering in multi-site organizations is **cultural diversity**. The differences in stakeholder's language and national culture create challenges for global collaboration. [8] Also differences in organizational and functional culture create challenges for requirements gathering. Geographically distributed branches of a company have the tendency to create their own organizational culture. All these issues create a significant challenge in achieving a common understanding about the requirements [6] which can lead to ambiguous or even incorrect requirements [7].

Another problem in requirements gathering in multi-site organizations is the **time difference**. If there is a great time difference between the participants and organizers of the requirements gathering there is little overlap available for synchronous collaboration. Because of this asynchronous channels can become the main form of communication [8]. This leads to difficulties in understanding and negotiating about the requirements. Also when different sites are located across continents it is always either too early or too late for the other party for having a teleconference [6]. Fortunately the time difference wasn't a problem in my work since all the parties involved were located within 2 time zones having only 1 hour time difference.

Next the different aspects of inadequate communication, knowledge management, cultural difference, and time difference are explored in more detail.

Differences in culture and business

Since most of the business units of the Case Company are outside Finland the common language used is English, which is the second language for everyone. This might have had a direct impact on the requirements elicitation and validation. Although the level of English was quite high among participants there were a few cases where the form of the requirement lead to small debates and in this case the language might have been one of the reasons.

Distance widens the gap between the different functional departments of the organization. Therefore different business units have differences in organizational culture as well as in functional culture. This makes it a lot more difficult to achieve a common understanding between different parts of a multi-site. [6]

Also because of the differences in culture and business requirements are being expressed using diverse terminologies, thus making the analysis for conflicts, and redundancies difficult. Furthermore the prioritization and negotiation of requirements, in the context of diverse business units and limited development resources, becomes a challenge. [6]

Appropriate participation of system users and field personnel

The distance represents a significant barrier to interactions between development and system users, affecting developers' involvement in the gathering, analysis, and validation of requirements. [6]

This study didn't include any end-users. It was a project that was happening inside the Case Company only including its employees. It is true thought that the effects of requirements gathering project will also affect end-users if these changes are to be implemented this. This is because if a user interface development tool is radically changed the interfaces created by it will also have a different form which will affect end-users' working experience throughout the interface. Unfortunately in order to keep this project at a reasonable scale we couldn't include end-users in this study.

Not having end-users effects on the priorities of requirements since only the Case Company people were evaluating them giving only their point of view on the matter. For example business impact and relevance for the end-user were only evaluated based on the Case Company employees (further explained in section 4.2). Also the end-user category used to categorize all the requirements containing what end-users might want was only evaluated by the Case Company employees. Since the information about customers was indirect a lot of potential information was left to be gathered affecting the results of the research. Other than that the effects are less obvious and very indirect but still need to be taken into consideration when evaluating the results of this requirements gathering study.

In order to run the project in reasonable time and keep the expected diversity of the study at a manageable level it was a clear choice not to include end-users in to the study despite knowing what kind of effects it might have.

Awareness of local work context and informal communication

Stakeholders have diminished opportunities to know “what is going on at the other site”, because of insufficient familiarity with the activities of remote group members and background information that make work contexts meaningful. This leads to misunderstanding of the nature of the requirements since the context might not be fully understood. [6]

Informal communication within an organization is important due to its numerous positive effects [8]. When there is a culture of informal communication everyone knows that they can just walk to someone’s desk every time there is a requirement related issue. This way the chances for misunderstanding and finding disagreements are quite small. Also this way it is easy to make immediate adjustments in the requirements based on the feedback collected from others. In addition informal communication gives info about local work context for example with whom to negotiate with to “get things moving” (who is in the right position) to guarantee the success of the requirements gathering project. On the other hand lack of informal communication doesn’t really have much of an effect inside the requirements gathering team since they already have developed their culture and processes which keep them together. [6]

When I started the project for the Case Company I had no previous knowledge of process automation applications and how their functionality. Because of the short duration of this project it wasn’t really possible for me to obtain the same level of understanding that my colleagues who had been working on this domain for numerous decades. In other words during the project my domain understanding was lacking and that created problems in planning and executing the requirements gathering process. I couldn’t fully understand all the finest details in operations related to process automation application creation. Furthermore I wasn’t fully aware how the business and operation of each business unit of the Case Company worked. This most likely has an effect on the level of detail of the gathered requirements.

Trusting working relationships

Due to distance it is harder to become a team and to get the feeling of belonging to the same team, going for the same goals, and having each other's best interest in mind. In other words it becomes harder for the team members to trust each other. Knowing each member of the team personally helps to build trust [7]. Trust consists of knowing the values of a person, to become engaged and committed, and to follow the same agenda (knowing the other person doesn't have any hidden agenda). Trust is developed slowly through evolutionary stages of working together and it is especially slow for distributed teams. Lack of trust leads to situation where different parties are just supervising their own rights looking for things, that might be taken to their disadvantage rather than focusing on getting the job done in the best possible way. Fear of this lead to giving only ambiguous information to other parties. [6]

Lack of trust also creates challenges in managing conflict and having an open discussion of interests. Distance makes more difficult to handle problems especially if they have organizational, political or social nature. For this reason cooperative strategic planning among different parties becomes very limited. Also distance makes it more difficult to manage conflict since it is not possible to openly discuss about different stakeholders' interests. Distance also makes negotiation about trade-offs more difficult. How well these trade-offs are made is dependent of the stakeholder's communication and knowledge management techniques in this distributed environment. [6]

Common understanding of requirements

When common understanding about requirements is lacking a problem called culture gap exists. The culture gap emerges when the developers are translating users' intentions. This way the developers have a strong effect on how the requirements are understood. Cultural gap is a common problem because usually the users and the developers use different languages to communicate about the domain. This can lead to a situation where the users do not understand the requirements developed by developers. Also users and developers usually tend to have different views of the same domain. The developer view is more focused on the organizational viewpoint where the users view is more focused on just getting the job done. [9]

Furthermore a different language is used between developers and business people: in the workshops there were present both software developers and marketing staff. This can easily lead to misunderstandings. Usually software developers are demanding more detailed requirements while business & marketing people want to get the "big picture" without being distracted by the details.

In order to make sure that all the parties understand the requirements similarly everyone should be included in requirements gathering. Unfortunately it is not possible to include all the key stakeholders in the meetings easily. Instead the information is channeled through a few people. This means that understanding of the requirements by people who didn't attend the meeting will get quite dependent on others' opinions.

Effective meetings

It is required that the key decision makers are present in the requirements gathering meetings. This is difficult since they usually have quite busy schedules and fitting schedules of people located in geographically highly disperse locations together is challenging. The whole project can get delayed because of difficulty in matching schedules. When the meeting date is finally decided there will be a lot of pressure put on one meeting since most people will have the attitude that everything should be done at once which is not very realistic.

Since this project there was several business units involved from different geographical locations while having the time pressure the only option to get everyone together was to arrange a teleconference meeting. Therefore teleconferencing wasn't chosen as the medium for requirements gathering in this project and there are several reasons for that. In teleconferences participants can misinterpret information about requirements and therefore information about requirements can get distorted. This leads to lack of understating about the meaning and purpose of requirements. This lack of understanding leads to difficulties in managing uncertainty, prioritizing requirements, and negotiation of requirements. In addition lack of visual contact leads to lowered awareness of presence and group behavior. Because of this some people can join and leave at different times and therefore the presence of some members might not be noticed. Also there will be difficulties in knowing who can be addressed about certain issues and therefore making the participation harder. Furthermore in teleconferences the mute button is often used and this adds to the creation of coalitions further decreasing the chance for truly open communication. It is also hard to present your ideas since tools for that are quite limited and therefore the ideas won’t be clearly understood by others. [6]

Delay

Speed is regarded as one of the most important success factors in modern technology business and it is becoming of concern in global software development. It is possible that after the meetings changes must be made to the requirements due to for example misunderstandings or lack of information. Mostly global companies’ communication is handled through email, making phone calls or waiting for meetings to take place. The biggest reason for this is time zone differences. Therefore the common time that can be spent on solving misunderstandings about the requirements is very limited because of this small misunderstandings can become bigger problems. [6] Fortunately in the Case Company this wasn’t an issue since all the business units are located within 2 time zones only having a one hour time difference.

Pinning down requirements

Due to distance it is difficult to make trade-offs on a large list of diverse requirements in the face of uncertainty since the means and time for common negotiation is quite limited. Also prioritizing requirements becomes very difficult since different parties have different business goals which in the worst case can also be contradicting making the prioritizing of requirements even more challenging. [6]

3.1.2. Challenges for participants in requirements gathering

Participant unsure about their needs

One challenge in requirements gathering is related to the participants themselves and the way how they express the requirements. It is possible that the participants unsure about their needs. They might only have a high-level idea about what they need but they can’t really pin-point what is the actual need. Therefore it is also challenging for them to articulate the ideas clearly because the form is not yet known. [14]

Different ways of articulating needs among participants

In a multi-site organization there are people participating in the requirements gathering from very different cultural and organizational backgrounds. Therefore there exists many different ways of articulating true needs. For example business minded people tend to think in a higher level of abstraction whereas engineer minded people tend to think on a higher level of detail. This is just a generalization but the key point is that people from different backgrounds think differently and therefore it creates challenges in understanding the requirements. [13]

If the cultural differences didn’t cause any problems in understanding the requirements still participants can have difficulties in articulating needs clearly. The participants might think that they addressed something, and perhaps they did, as far as they are concerned, but not in the way someone else would recognize or understand. Further challenge in articulating the needs clearly is that most of the participants are not used to articulate their needs. Participants’ thinking is typically oriented toward doing their job, rather than defining it. Usually requirements are gathered from people who are experienced and know their job well. In this situation their very skill makes it hard for them to recognize all that goes on in what they do very well. For the same reason sport super stars rarely become effective coaches: they know they are good but they can’t define why because it is so deeply rooted in their mind. [13]

Difficulties of scope

There is always the difficulty of scope present in requirements gathering projects. The problem is that the participants are usually asked to cover the whole scope of their requirements in a limited amount of time. The problem here is that their thoughts might not be that well organized. They may overlook things and make assumptions, often without being aware they are doing so. [13] In larger scale many relevant requirements can be left out from several business units resulting in inadequate picture about the needs of different business units. Also in order to make the scope of the study adequate enough in the project for the Case Company a great number of business units should be included which causes challenges with scheduling if the project is wished to be completed in fast pace.

Managing change becomes inevitable in requirements gathering when it is time to implement the requirements but the earlier it is started the easier it is. The reason for this is that then organization has enough time to adapt to the change. Therefore it is essential to act as early as possible. The best approach for change management is the incremental approach. The requirements should offer enough quality information for strategic decision making. The requirements gathering should build organizational awareness, understanding and psychological commitment to the change. The requirements gathering process should decrease uncertainty surrounding such decisions and psychological commitment by allowing for interactive learning throughout the Case Company. In practice the people should be convinced why the change in needed, how it will be implemented, how is the change progressing at all times, and in the end let everyone know how it all worked out and how the change will actually affect them. Also the gathered information should improve the quality of decisions about the change itself by systematically involving those with most specific knowledge and including those in the process who must carry out the decisions. The challenge is how to make all of this happen. [15]

In requirements gathering the researchers are not objective observers but are involved in the research situation. The researcher's effect arises when expectations or values of the evaluators distort participants' evaluation. For example the researcher can be too cynical or enthusiastic about some aspects mentioned during a workshop. [12] Especially if the researchers have authority in the organization they will guide participants' thinking resulting in wrong requirements. The gathered information is never perfectly complete so therefore in the analysis phase researcher has a big responsibility on how the results are interpreted. Wrong interpretations can lead to wrong requirements resulting in a product that is not needed.

3.1.3. Requirements gathering challenges due to human factors

Limitations of humans as information processors

The processing capability of working memory is constrained, having a capacity of seven plus or minus two items of information for most of the people. For requirements elicitation this means that the analyst and user can only focus on small amounts of information at a time. This also means that the analyst may only consider a narrow set of sources of information which may lead to loss of highly relevant information. As a result of working memory constrains, people cannot consider all possible alternatives to a problem and must construct simplified models for solving them. As a consequence people are satisfied with suboptimal solution that fits their simplified models. [14] Also since only a few people from a business unit can be present at a time participants can also think about their colleagues opinions trying to form an overall consensus of everyone's needs. This creates an additional cognitive constrain for the participants and slows down the progress requirements elicitation.

Long-term memory describes the cognitive structures that facilitate the storage and retrieval of information not active in working memory. While the storage capacity of long-term memory is virtually limitless, people have difficulties in the recall and use of information from long-term memory. As a result, requirements tend to be overly focused on the analyst’s knowledge of current procedures, prior experiences, or recently encountered information. Also research has shown that people store their memories just by storing the general idea of it. To recall this information an analyst must reconstruct the knowledge from the general picture that is stored. Unfortunately, this reconstruction can lead to an inexact recollection of knowledge leading to ambiguous or even wrong requirements. [14]

Heuristics and biases

Availability refers to the tendency of people to rely on the ease which information can be remembered or visualized when assessing the likelihood of events. Although this heuristic works well much of the time, its use invokes biases resulting from the recency and vividness of an event or simply from information that is easily recalled. This will limit the breadth and depth of requirements gathering. [14]

Anchoring on a problem structure usually occurs when some piece of information has been offered early in the problem-solving process. Subsequent information will result in adjustment of the initial problem-structure but this adjustment is insufficient. The information offered early will still have a stronger effect on the problem structure since it was based on that. In other words the problem structure is anchored to this early piece of information. As a result, the analyst has difficulty exploring alternative paths of requirements elicitation. [14]

Another bias resulting from anchoring and adjustment bias is overconfidence. People consistently exhibit overconfidence in knowledge across a wide range of tasks and difficulty levels even if their knowledge about the domain is fairly poor. This may lead to misunderstandings about the true nature of requirements. This is a threat especially when the analyst is required to perform requirements elicitation in various functional areas of an organization, possibly including some in which he has little or no substantive knowledge. [14]

Representativeness is defined as the degree to which an event or item is representative of the characteristics of the population from which it came. People usually categorize problems based on initial cues and take what is known about this category to complete the formulation of the problem. One bias resulting from this heuristic is insensitivity to sample size. Typically people do not give appropriate consideration to sample size and draw faulty conclusions based on small samples. [14]

Other challenges

Experimental evidence reveals that people possess a strong tendency to seek only confirmatory evidence and fail to consider alternative hypotheses. This is a threat especially if there is a relatively short time offered for requirements elicitation in the project. [14] This was the case in my work.

3.1.4. Challenges due to the nature of requirements

Changing requirements

The requirements can change in time and therefore it is a challenge to keep the requirements up to date. Outdated requirements are worse than no requirements at all since they lead the project to a wrong direction. [16] Although in this work this issue wasn't addressed at since the purpose was only to find an answer to the question 'is it possible to unify the Case Company's user interface development efforts?' The nature of the question implies that the process itself is not continuous.

Incomplete requirements

If the requirement is complete then all conditions under which the requirement applies are stated, and the requirement expresses a whole idea or statement. [16]

Unclear requirements

If the requirements are clear they must be concise as well as unambiguous. Being concise means that the requirements are stated as simply as possible. Unambiguous on the other hand means that the requirement can be interpreted in only one way. [16] It is challenging for a requirement to be concise as well as unambiguous since these goals are actually contradicting. If a requirement is stated as simply as possible it can become ambiguous. On the other hand if a requirement is stated in great detail to avoid ambiguousness it will not be concise.

Invalid requirements

In order for a requirement to be correct the facts related to the requirement must be accurate as well as technically and legally possible. [16]

Infeasible requirements

It means requirements that are technically difficult or even impossible to implement. [16] Checking the feasibility of requirements wasn't in the scope of this work but in the project it was the next step to do after the requirements had been gathered.

Non-Traceable requirements

Traceability means that the requirement can be traced to its source, and it can be tracked throughout the system (e.g. to the design, code, test, and documentation). [16]

Inconsistent requirements

If the requirement is consistent it is not conflicting with other requirements. [16] However in this work it must be mentioned that this consistency is assumed to hold only within a business unit of the Case Company since their goals and ways of operating are quite different.

Duplicate requirements

All the requirements are unique requirements and there are no duplicates. [16]

3.2.Challenges in the Case Company: Different user interface development tools

One of the key problems in the requirements gathering project done for the Case Company was that the requirements where gathered from business units using different user interface development tools, namely Vtrin, Personal Assistant, and Smart Client. These user interface development tools are all used to create application displays and configure application functionality to the displays. The problem here is that these different tools are using different technologies and have different operational logic. Because of different technologies and operational logic the needs for new features, fixes, and support is quite different. As an example a user interface development tool like Vtrin gives a lot of freedom in creating presentations whereas Personal Assistant offers quite limited options for presentation creation.

In this section 3 different user interface development tools used in the Case Company are presented. They are Vtrin, Personal Assistant, and Smart Client. The purpose of this section is merely to give an overall view of what kind of user interface development tools the Case Company is currently using. Since in this work the focus was on high level user requirements this section will not go into technical details. Different visualization tools will be presented from the point of view how an application engineer can create visualization with them. The reason for this is that the step by step task based presentation is the most intuitive way for understanding the basic structure of each user interface development tool.

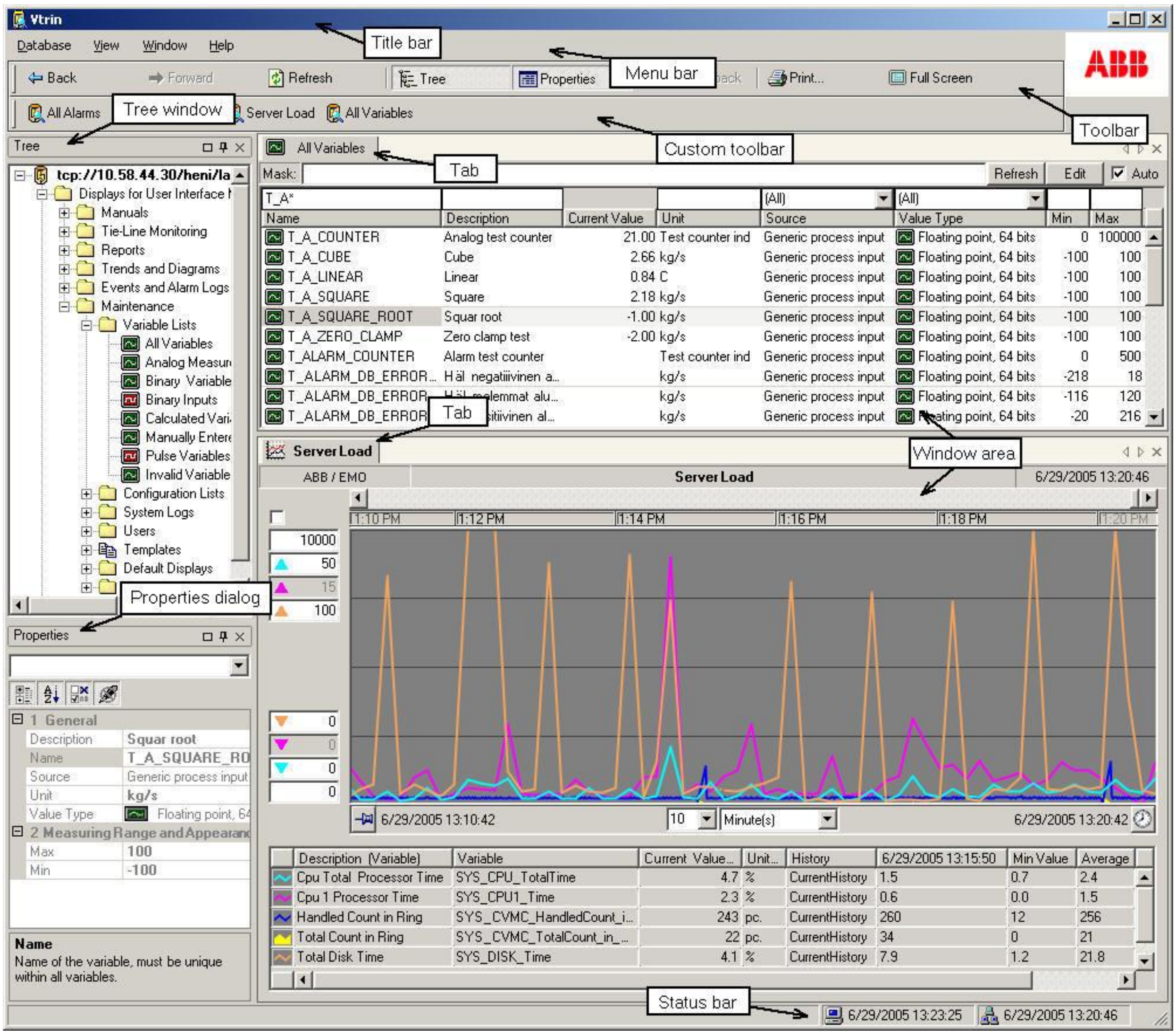
3.2.1. Vtrin

Vtrin is the name used for the Industrial ^{IT} User interface used in the Case Company. Vtrin is a .NET click-once user interface and a development tool for ABB cpmPlus History and for CPM applications and solutions in ABB Process Automation Division. Vtrin can be used to connect to CpmPlus History databases (for storing process information gathered from industrial process equipment) and also to several IndustrialIT systems such as Production Planning, Production Management, Quality Management, and Energy Management and Optimization. In this section Vtrin's user interface, different displays and basic functionality for application creation are presented shortly. [3]

Basic concepts

The main view

The main view is the place where creation of the graphical user interface takes place. In figure 1 the workspace of Vtrin user interface is presented.



T_A*				(All)	(All)		
Name	Description	Current Value	Unit	Source	Value Type	Min	Max
T_A_COUNTER	Analog test counter	21.00	Test counter ind	Generic process input	Floating point, 64 bits	0	100000
T_A_CUBE	Cube	2.66	kg/s	Generic process input	Floating point, 64 bits	-100	100
T_A_LINEAR	Linear	0.84	C	Generic process input	Floating point, 64 bits	-100	100
T_A_SQUARE	Square	2.18	kg/s	Generic process input	Floating point, 64 bits	-100	100
T_A_SQUARE_ROOT	Squar root	-1.00	kg/s	Generic process input	Floating point, 64 bits	-100	100
T_A_ZERO_CLAMP	Zero clamp test	-2.00	kg/s	Generic process input	Floating point, 64 bits	-100	100
T_ALARM_COUNTER	Alarm test counter		Test counter ind	Generic process input	Floating point, 64 bits	0	500
T_ALARM_DB_ERROR...	Häl negatiivinen a...		kg/s	Generic process input	Floating point, 64 bits	-218	18
T_ALARM_DB_ERROR	Häl molemmat alu...		kg/s	Generic process input	Floating point, 64 bits	-116	120
T_ALARM_DB_ERROR	sitiivinen al...		kg/s	Generic process input	ating point, 64 bits	-20	216

1 General	
Description	Squar root
Name	T_A_SQUARE_RO
Source	Generic process input
Unit	kg/s
Value Type	Floating point, 64
2 Measuring Range and Appearanc	
Max	100
Min	-100

Name
Name of the variable, must be unique within all variables.

6/29/2005 13:10:42 — 6/29/2005 13:20:42

Description (Variable)	Variable	Current Value...	Unit...	History	6/29/2005 13:15:50	Min Value	Average
Cpu Total Processor Time	SYS_CPU_TotalTime	4.7	%	CurrentHistory	1.5	0.7	2.4
Cpu 1 Processor Time	SYS_CPU1_Time	2.3	%	CurrentHistory	0.6	0.0	1.5
Handled Count in Ring	SYS_CVMC_HandledCount_i...	243	pc.	CurrentHistory	260	12	256
Total Count in Ring	SYS_CVMC_TotalCount_in_...	22	pc.	CurrentHistory	34	0	21
Total Disk Time	SYS_DISK_Time	4.1	%	CurrentHistory	7.9	1.2	21.8

6/29/2005 13:23:25 6/29/2005 13:20:46

Figure 1: Main view of Vtrin graphical user interface

1. **Title bar**: Shows the title of the window
2. **Menu bar**: Consists of database, view, window and help menu
3. **Tree window**: Shows all the windows that are currently available in the system in a hierarchical tree structure. It also shows all the variables used to display data.
4. **Window area**: The charts and other graphical presentations of data in the tree are displayed in the window area.
5. **Properties dialog**: displays the properties of a selected element in the window area.
6. **Toolbar**: Consists of most common functionalities: browsing system state, data playback, shortcuts, and designer mode.
7. **Custom toolbar**: used to put shortcuts to most frequently used windows as buttons.
8. **Tab**: shows all the currently opened windows in the window area.
9. **Status bar**: shows the current time and date of user’s PC and server’s PC.

Designer mode

When designer mode is on it is possible to modify all the aspects of graphical user interface components in the Window area such as size, positioning, visual appearance, and also the properties of each component. When designer mode is off the user can only run the application and browse the windows and test the enabled functionalities. [3]

Drag and drop functionality

The drag and drop functionalities of Vtrin user interface offer a lot of flexibility and customization for the user. The location of tabs can be changed to be in any side of the screen. Tree window and properties window can be detached as separate windows and relocate them to any side of the screen, their size and position can be changed freely and they can also be dragged and dropped as tabs. All in all there are a lot of possibilities for customizing the main view. [3]

Creating basic displays and adding controls

It is usually the case that in Vtrin applications are merely modified rather than created from the scratch, but in order to give a clearer idea about the interface it is better to start with an empty workplace. Also it must be mentioned that the interface offers many different ways of operating and here only one way of working around the interface is presented. When the user starts creating a whole new application the first thing to do is to create required displays that will contain the functionality needed for the application. Different displays are created using **Tree Window** presented in figure 1. [3]

Tree window in figure 1 shows all the windows that are currently available in the system in a hierarchical tree structure. The tree consists of folders and several sub-folders. These folders can contain charts, process diagrams, lists, tables, or reports. Tree items can be added and removed freely from a menu that is opened by right-clicking the tree area. Tree items including charts, process diagrams and lists will be presented in more detail later on. Search functionality allows searching for items matching the criteria inside the **Tree window**. The items found are based on the names of the nodes. There is also a search used to find tree items. It uses auto-complete functionality narrowing the search as letters are typed. [3]

The user can create new displays by right-clicking on the folders on the **Tree Window**. It opens a pop-up menu where user selects Add Tree Item in order to add new displays. These displays can be for example charts, process diagrams or lists. When a user creates a display an empty template of a selected display containing the basic components is created and further content can be added later. [3]

In order to add data for the system to process the user must be connected to a database. The data will be viewed as variables in the **Tree Window**. The number of variables can get huge quite easily and managing them becomes challenging. To improve the situation a **Plant Model** is used. It can be found from the **Tree Window**. The **Plant model** is used to group and organize variables into a hierarchical structure. Grouped variables can be dragged and dropped to a list, a trend or some other appropriate display. Depending on the display a certain type of graphical display is automatically generated from the variables. [3]

If there is need to change properties of any items in the window area **Properties dialog** can be used. **Properties dialog** can be seen in figure 1. It displays the properties of a selected element in the window area. Any items in the **Window area** can be selected and their properties can be seen in the properties view by clicking on the item. For example there can be a table in the **Window area** and you can view the data of selected variable in the table. [3]

Shortcuts

Roles

The roles are used for storing the state of the workspace for different users or roles. The state selected consisting of visible displays as well as their positioning on the screen can be saved to a role. For example there can be a maintenance role which only shows the functionalities required by the tasks that are relevant for maintenance. Each role can be accessed from the role tab in the lower left corner of the screen. [3]

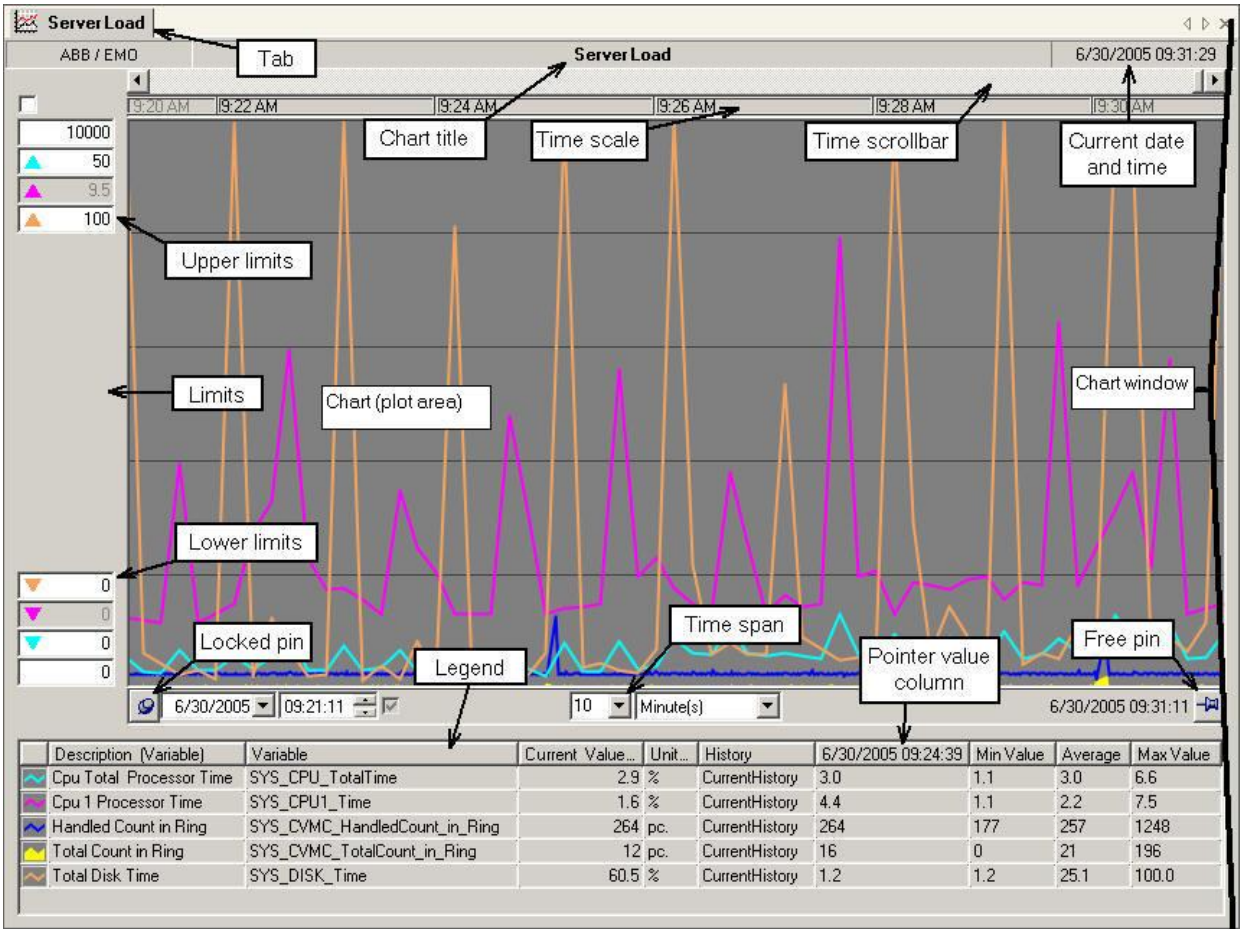
Custom toolbar

Custom toolbar shown in figure 1 can be used to put most frequently used windows as buttons in the toolbar. These windows can be opened by clicking the buttons assigned for the window. It is possible to add a view from the tree to the toolbar by drag & drop functionality. [3]

Presenting the data visualization capabilities of Vtrin

Charts

The graphical information in Vtrin window area is presented by charts. The charts can contain various graphical elements, such as pictures, graphs, plots, value controls, text blocks and so on. One example of a chart is called a trend. A trend is a control used to represent both run-time and historical data in the form of graphs. More detailed description of a similar trend component is presented in subsection 3.2.3. In figure 2 there is a model graph. Below the graph there is a list of variables that are displayed on the graph. The variables are automatically displayed in a certain way depending on their type. [3]



	Description (Variable)	Variable	Current Value...	Unit...	History	6/30/2005 09:24:39	Min Value	Average	Max Value
∿	Cpu Total Processor Time	SYS_CPU_TotalTime	2.9	%	CurrentHistory	3.0	1.1	3.0	6.6
∿	Cpu 1 Processor Time	SYS_CPU1_Time	1.6	%	CurrentHistory	4.4	1.1	2.2	7.5
∿	Handled Count in Ring	SYS_CVMC_HandledCount_in_Ring	264	pc.	CurrentHistory	264	177	257	1248
∿	Total Count in Ring	SYS_CVMC_TotalCount_in_Ring	12	pc.	CurrentHistory	16	0	21	196
∿	Total Disk Time	SYS_DISK_Time	60.5	%	CurrentHistory	1.2	1.2	25.1	100.0

Figure 2: Example chart

There are several functionalities available in the charts. Only a few of them will be presented here. User can view the values on a certain point on the graph my placing the cursor there. There is also a zoom functionality which helps to get a value from a short period of time. It is also possible to change the time range of the variables if the graphical presentation of variables during a certain period of time needs to be seen. If this is not enough the form of the whole data can be also changed on-the-fly. Data from a list can be directly viewed as a graph or data from a graph can be viewed in a list form. [3]

If the user wishes to find specific variables to edit their properties the filter functionality becomes useful. The user can define a filter to perform the desired operations by typing in the keywords defined in the filter syntax for example the keyword COUNT counts the number of the samples within period. [3]

When the required data is inserted on the chart it is time to analyze it. There is functionality for creating a summary of the data. It can show for example the minimum, maximum, average and deviation values from the data. If the functionalities offered by Vtrin are not enough for data analysis the data can be copied to clipboard and then sent to for example to Excel for further analysis. [3]

Since the systems are used to gather data continuously there is always possibility for errors in the data. To correct the data the user can update the values and activate so called recollection to compute the calculated variables and transformations. This propagates the changes to all affected variable sets. [3]

Process diagrams

Process diagrams present current and history values. Process diagrams are displayed in chart windows as seen in figure 3. The process diagram displays a fixed process background, as well as updated process data in the form of values, columns, symbols or plots (sub-diagrams presented in boxes next to the graphical presentation of an industrial process). It may also contain links to other figures, either as an individual link text or linked in current values and symbols. [3]

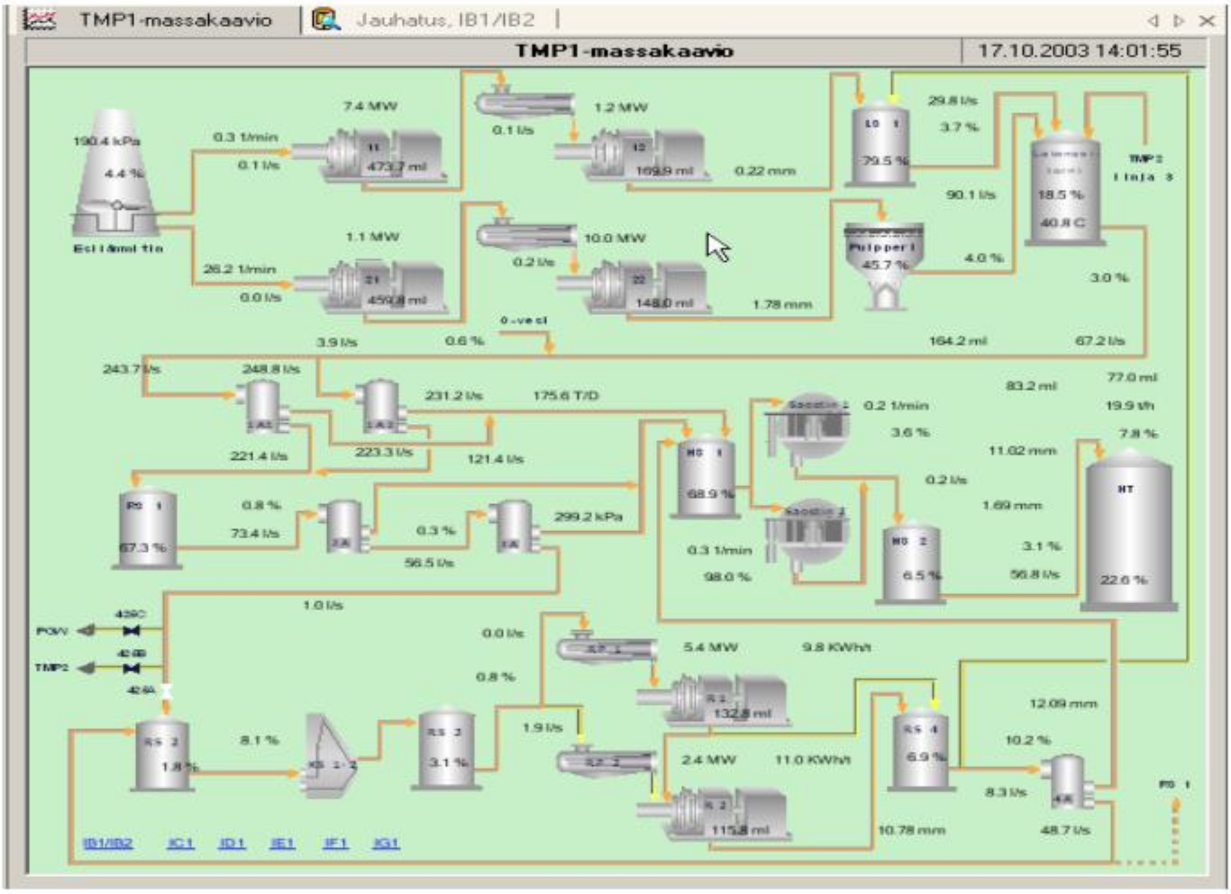
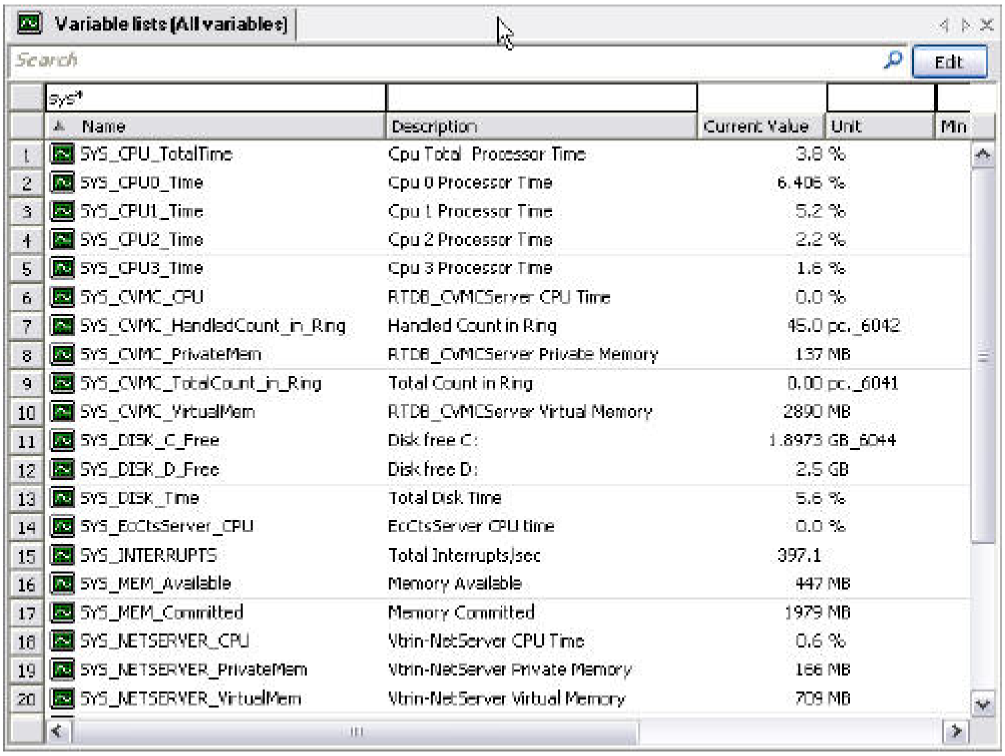


Figure 3: An example process diagram

Lists

They are used to show database data in a list format. Lists can be variable lists, alarm logs, event logs, maintenance logs and user logs. In figure 4 a variable list presented. All the lists present the data in almost identical manner. There are 3 most common list types including list screen, variable list and also event and alarm tables. Variable list is basically a list screen containing variables with advanced sorting options whereas alarm and event list is used to acknowledge new alarms and it shows all active unacknowledged process alarms triggered by the system. Alarms are used to alert about unexpected incidents in the production. [3]



	Name	Description	Current Value	Unit	Min
	sys*				
1	SYS_CPU_TotalTime	Cpu Total Processor Time	3.8	%	
2	SYS_CPU0_Time	Cpu 0 Processor Time	6.406	%	
3	SYS_CPU1_Time	Cpu 1 Processor Time	5.2	%	
4	SYS_CPU2_Time	Cpu 2 Processor Time	2.2	%	
5	SYS_CPU3_Time	Cpu 3 Processor Time	1.6	%	
6	SYS_CVMC_CPU	RTDB_CVMCServer CPU Time	0.0	%	
7	SYS_CVMC_HandledCount_in_Ring	Handled Count in Ring	45.0	pc._6042	
8	SYS_CVMC_PrivateMem	RTDB_CVMCServer Private Memory	137	MB	
9	SYS_CVMC_TotalCount_in_Ring	Total Count in Ring	0.00	pc._6041	
10	SYS_CVMC_VirtualMem	RTDB_CVMCServer Virtual Memory	2890	MB	
11	SYS_DISK_C_Free	Disk free C:	1.8973	GB_6044	
12	SYS_DISK_D_Free	Disk free D:	2.5	GB	
13	SYS_DISK_Time	Total Disk Time	5.6	%	
14	SYS_EcCtsServer_CPU	EcCtsServer CPU time	0.0	%	
15	SYS_INTERRUPTS	Total Interrupts/sec	397.1		
16	SYS_MEM_Available	Memory Available	447	MB	
17	SYS_MEM_Committed	Memory Committed	1979	MB	
18	SYS_NETSERVER_CPU	Vtrin-NetServer CPU Time	0.6	%	
19	SYS_NETSERVER_PrivateMem	Vtrin-NetServer Private Memory	166	MB	
20	SYS_NETSERVER_VirtualMem	Vtrin-NetServer Virtual Memory	709	MB	

Figure 4: Sample list view

In order to find certain type of items from a list the sort functionality can be used. It sorts list items using column headings either in ascending or descending order. If an item matching more specific criteria needs to be found a search functionality can be used instead. The search can be used to look for items by their name or their property. It is also possible to search items with several criteria using Boolean variables (=, >, <). Only the variables matching the criteria are displayed. [3]

3.2.2. Personal Assistant

The personal assistant is part of the Production Reporting product and it is used to create reporting applications based on the data offered by customer. Personal Assistant is a reporting tool for applications and solutions for minerals industry. The data is collected from all parts of the plant (by the control system). The data stored in a database on a cpmPlus History Server and processed, sorted and formed into various reports that can be accessed by Personal Assistant. Personal Assistant uses a web browser user interface. It means that all the usual browser functions will be available for the user in addition to Personal Assistants own functionalities. [9]

Through Personal Assistant the users can access various types of data such as production data, quality data, emission measurements and so forth. Users can also access varied range of reports such as production reports, operation reports, event/alarm reports, operating hour reports, maintenance reports and so forth. [9]

In figure 5 the general display of Personal Assistant's user interface is presented. It consists of 1. Tree view, 2. Work Area, 3. Browser Header, and 4. Tab Strip.

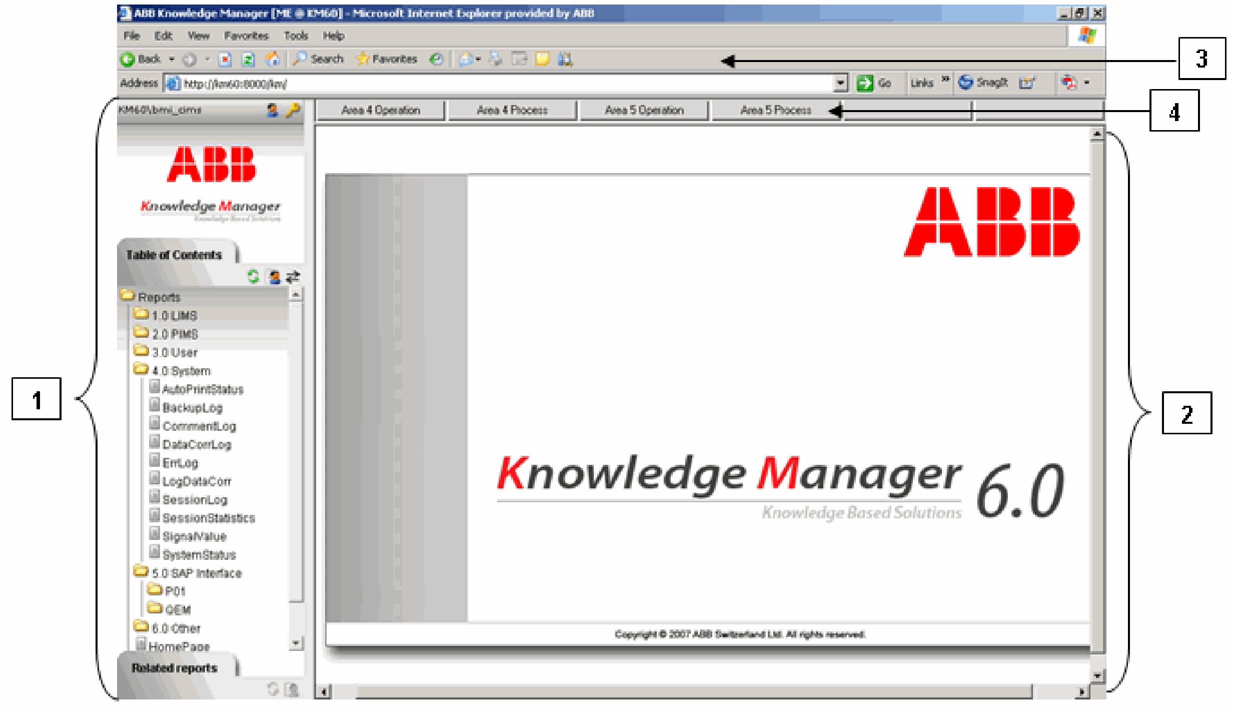
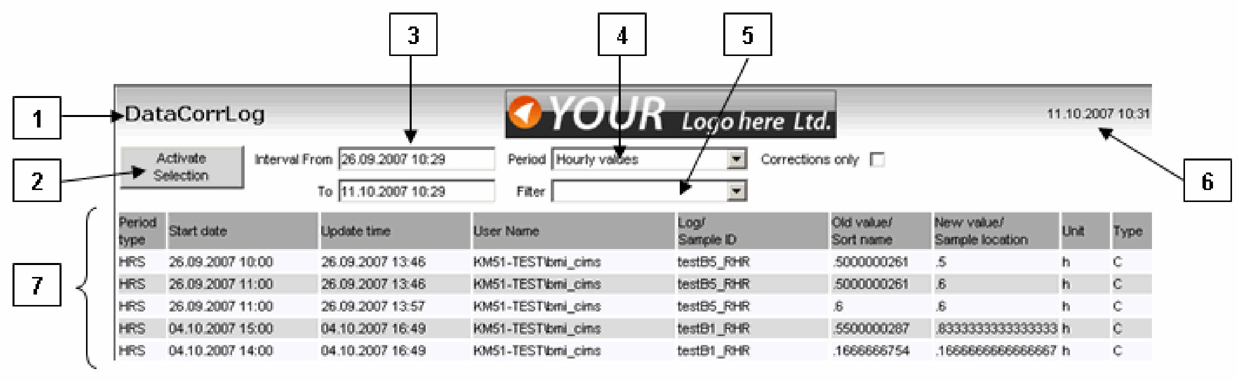


Figure 5: General Display of Personal Assistant

Tree view shows the folders which contain the reports that are available for the user. The visibility of the folders is configured by the system administrator. Reports are opened to the Work Area to be viewed by clicking on their name. The tree view can be managed through pop-up menus opened by right-clicking either the folders or the reports. [9]

Users can create a new report or an URL document by right clicking on a folder on the Tree View and selecting one of the options (Operation Report, Sample Report, URL document, Trend Report). After that the new report must be named and also a description needs to be written for it. After creating the report user must configure it in order to have the desired visual appearance as well as the data that the user wants to be displayed. The user can also modify how the data will be displayed from the user preferences. The settings are attached to a user name which means that if a certain user logs in to the system the settings set up by that user will be activated. These preferences can be for example language, Date/Time format, or how to display the different views on the interface. [9]

The system has several shortcuts to make it more efficient to use. The Tab Strip is used to directly navigate to a desired report without having to search from the Tree View. The tabs are usually used for around 8 most important reports. When a report is selected from the Tab Strip the set of tabs defined for the related folder are activated (context sensitive). Another set of shortcuts are keys located at the bottom part of the screen. A click on a key takes user directly to a predefined linked report. [9]



DataCorrLog

11.10.2007 10:31

Interval From 26.09.2007 10:29

To 11.10.2007 10:29

Period Hourly values

Filter

Corrections only □

Period type	Start date	Update time	User Name	Log/ Sample ID	Old value/ Sort name	New value/ Sample location	Unit	Type
HRS	26.09.2007 10:00	26.09.2007 13:46	KM51-TEST\bmi_cims	testB5_RHR	.5000000261	.5	h	C
HRS	26.09.2007 11:00	26.09.2007 13:46	KM51-TEST\bmi_cims	testB5_RHR	.5000000261	.6	h	C
HRS	26.09.2007 11:00	26.09.2007 13:57	KM51-TEST\bmi_cims	testB5_RHR	.6	.6	h	C
HRS	04.10.2007 15:00	04.10.2007 16:49	KM51-TEST\bmi_cims	testB1_RHR	.5500000287	.8333333333333333	h	C
HRS	04.10.2007 14:00	04.10.2007 16:49	KM51-TEST\bmi_cims	testB1_RHR	.1666666754	.1666666666666667	h	C

Figure 6: Typical fields of a report

There are several report types in Personal Assistant. In figure 6 there is a preview of the visual appearance of a report, in this case a backup log report. Here are a few examples:

Backup Log Report: The Backup Log provides detail on the overnight backups and the events that occurred during the backup [9]

Chart Reports: Graphically displaying the data in a report. It is usually a graphical presentation of several values with time at x-axis and value at y-axis. It is possible to make either bar, line, pie or stacked charts. In figure 7 there is one example of a chart report called bar chart report. [9]

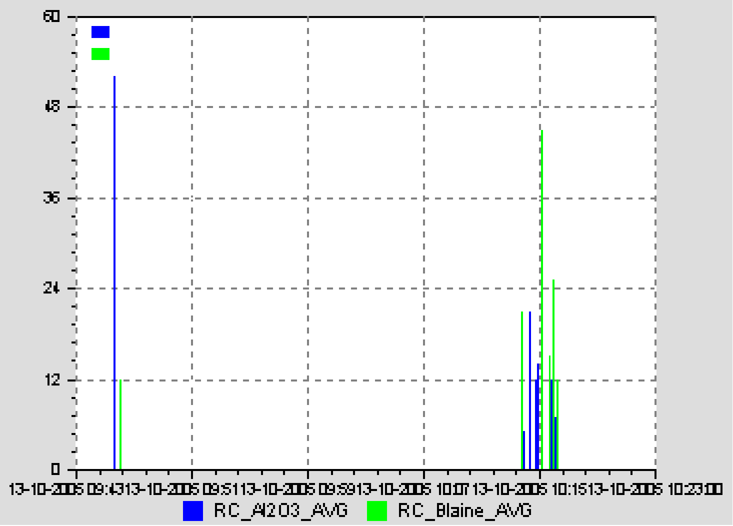
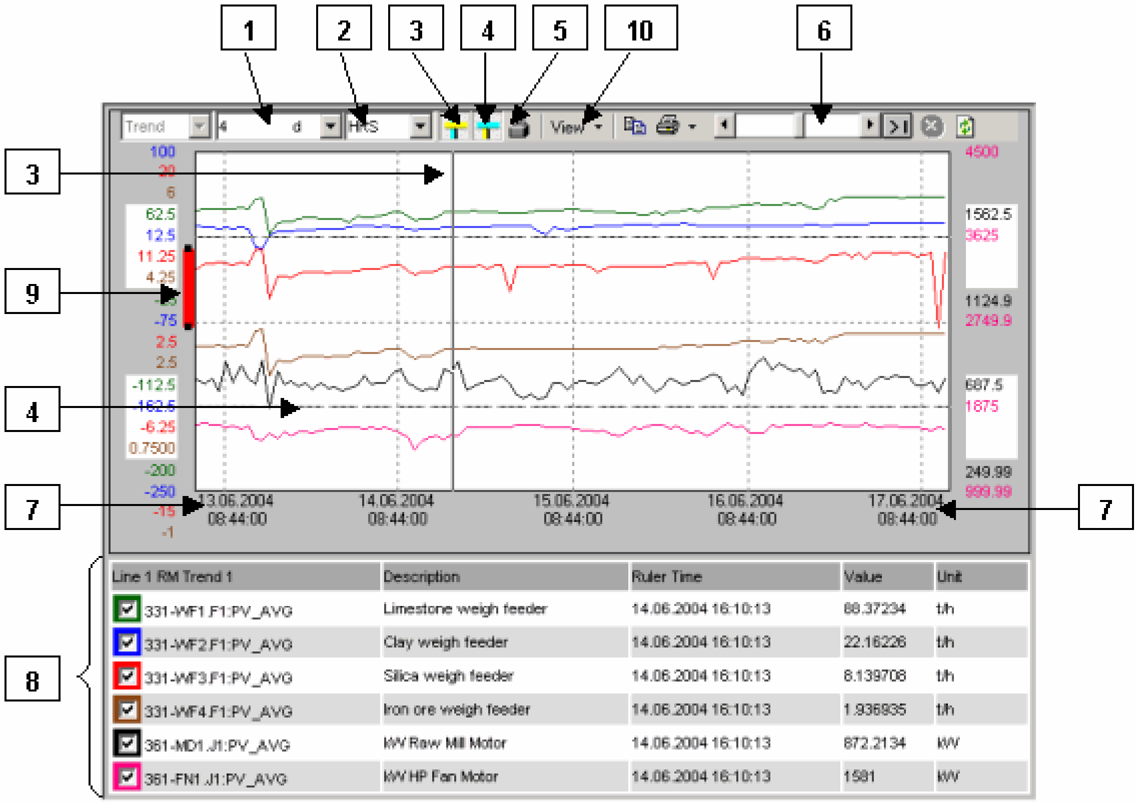


Figure 7: Bar Chart Report

Trend reports: With the trend report it is possible to show the recorded data graphically. Below there is signal information about production events and above that this information is presented graphically during selected time span. Vertical time (point 3) –and horizontal value (point 4) rulers are used to get detailed information from specific points of the graph.



Line 1 RM Trend 1	Description	Ruler Time	Value	Unit
☑ 331-WF1.F1:PV_AVG	Limestone weigh feeder	14.06.2004 16:10:13	88.37234	t/h
☑ 331-WF2.F1:PV_AVG	Clay weigh feeder	14.06.2004 16:10:13	22.16226	t/h
☑ 331-WF3.F1:PV_AVG	Silica weigh feeder	14.06.2004 16:10:13	8.139708	t/h
☑ 331-WF4.F1:PV_AVG	Iron ore weigh feeder	14.06.2004 16:10:13	1.936935	t/h
☑ 361-MD1.J1:PV_AVG	kW Raw Mill Motor	14.06.2004 16:10:13	872.2134	kW
☑ 361-FN1.J1:PV_AVG	kW HP Fan Motor	14.06.2004 16:10:13	1581	kW

[9]

Figure 8: Trend

In the figure 8 the upper part (points 3, 4, 7, and 9 in the figure) displays the trends and the lower part contains signal information (point 8 in the figure) from which the graphs in the upper part are formed.

3.2.3. Smart Client

CpmPlus Smart Client is a .NET click-once user interface for ABB 800xA control system (a system providing intelligent data access functions and views to both real-time and historical information from all customer applications) and ABB cpmPlus History. Smart client is a browser based thin client [17] that retrieves data from ABB's System 800xA and ABB cpmPlus History. Displays offered by Smart Client can be inside or outside the plant facility, as long as a connection to the plant exists. Smart client can be run for example with Internet Explorer browser. [9]

Smart client provides visual representation of industry graphics. The graphics represent physical entities like valves, pipes, gauges etc. in the form of controls. The controls can be either static controls for example tanks, pipers or dynamic controls like gauges, date and time. Each dynamic control can be associated with information which is continuously updated. Such information is called an Object. In general graphic displays are often used to present the state of the process or a part of a process, but are useful in any context where dynamic graphical information is needed. The operation of Smart client is focused around 4 key features including graphical displays, trending and statistical process control, alarm and event reporting and Microsoft Excel interface. [9]

Basic concepts

Design Mode/Run Mode

Smart Client uses two modes of operation, namely Run mode and Design Mode. The Design mode is used to build graphical displays, which involves drawing controls, and configuring their properties. All the changes made are stored to a server. After designing the workplace, the user can view its operation in Run Mode. The Run Mode is used to only execute and view the graphics for example to change the trend color thefore the changes are not saved. [9]

Workplace

When starting Smart Client a workplace is shown to the user. It contains the commands and tools to create, edit, print, analyze, export and save information. Workplace consists of one or more panels. Here the user creates the visual appearance of an application by combining controls such as trend graphs, text, numeric fiends, gauges, reports and so on. [9]

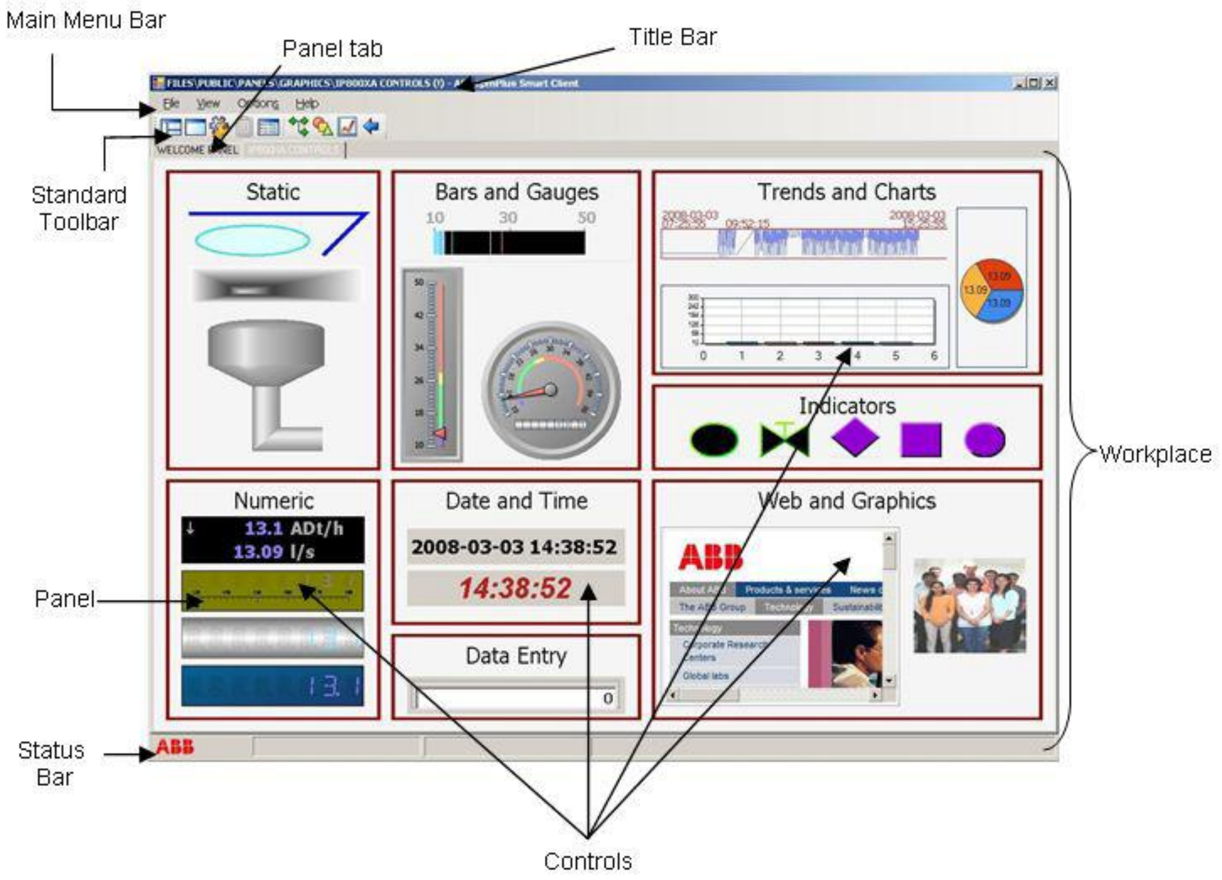


Figure 9: Smart Client main window

When Smart Client is started a main window appears which is shown in figure 9. The most important parts of the main window are **Workplace** and **Standard Toolbar**. The **Workplace** contains the commands and tools to create, edit, print, analyze, export, and save information. It also contains one or more panels which contain several controls. The **Standard Toolbar** on the other hand contains all the main functionalities in Smart Client. [9]

Drag and Drop Feature

All major Smart Client activities are performed using drag and drop operations. Users can freely drag components from the toolbox to the panels in the **Workplace**. The user can also drag and drop objects, containing continuously updated production information from a customer site, from the Property Browser or from any other control onto a control. [9]

Build graphics and basic controls

Standard toolbar shown in figure 10 consists of all the basic functionalities needed for creating interface graphics for applications. It also contains functionalities to associate data to user interface objects, perform calculations for this that, and analyze it. [9]

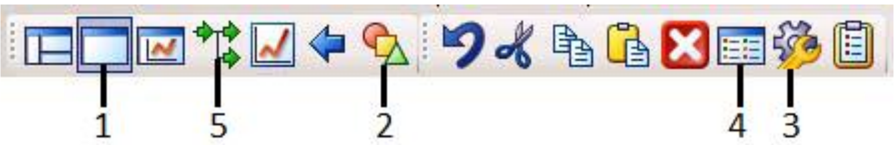
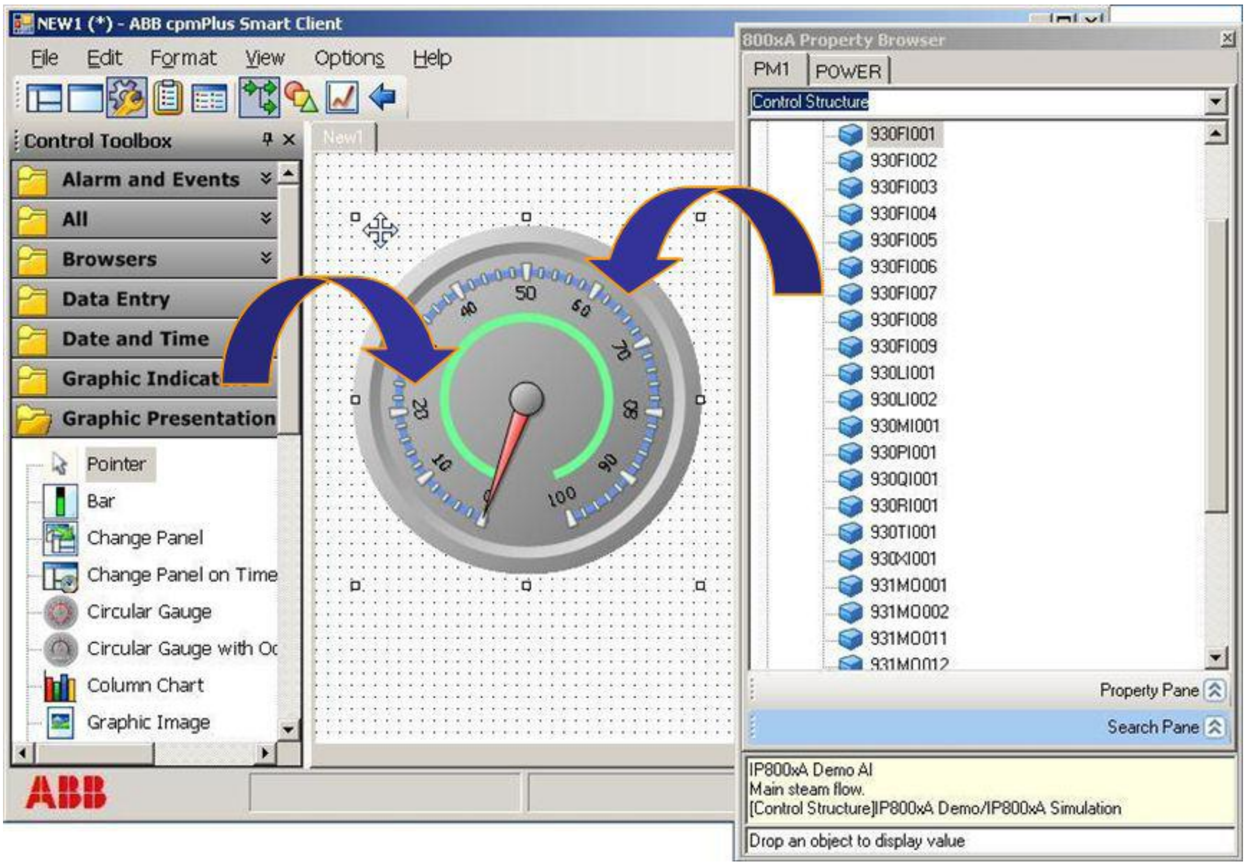


Figure 10: Standard Toolbar

In the beginning the **Workplace** (shown in figure 9) is empty so the first step is to add panel templates where graphical controls can be inserted. The panels can be dragged and dropped to the **Workplace** from the **Panel Browser** located in **Standard Toolbar** (icon 1). After that graphical primitives can be added to build the visual appearance of a panel. Examples of these graphical primitives are cones, lines, rectangles and text. These can be accessed from **Symbol Library** located in **Standard Toolbar** (icon 2). [9]

After required panels are finalized the next step is to add controls to the panels. Controls can be dragged and dropped from **Control Toolbox** located in the **Standard Toolbar** (icon 3). **Control Toolbox** contains controls such as data entry sheets, graphic indicators, motors, valves as shown in figure 11. The properties of controls can be changed from **Control**

Properties accessed from **Standard Toolbar** (icon 4). [9]



IP800xA Demo AI
Main steam flow.
[Control Structure]IP800xA Demo/IP800xA Simulation

Drop an object to display value

Figure 11: Adding a control to A Graphic Panel from Control Toolbox and configuring its properties

Once the required controls have been added it is time to configure them with a set of properties. This is done by dragging and dropping cpmPlus history objects to the control from **Property Browser** accessed from **Standard Toolbar** (icon 5). CpmPlus history objects containing all the relevant control properties such as measured value, description, engineering unit and limits defined in the Smart Client system. In figure 11 a control is added and then it is associated with a cpmPlus history object. [9]

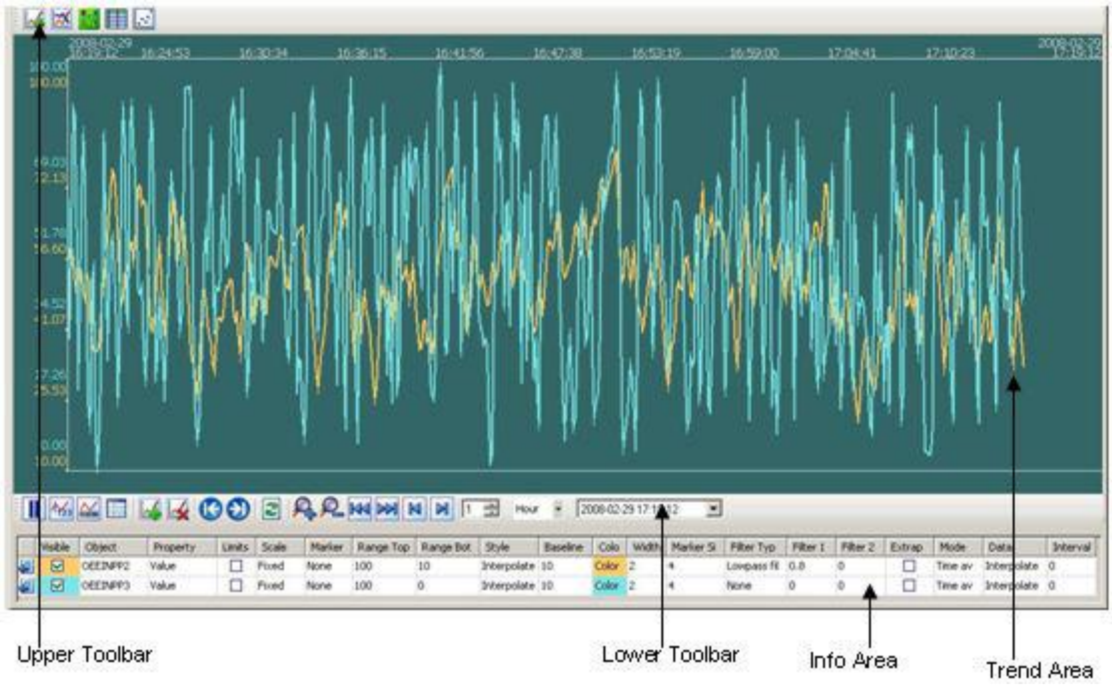
In order for the control to do something data must be associated to the control. This can be done by adding a Data Subscription to the control. Retrieval options for the data can be set in order to get specific values for the controls to use. [9]

In order to do operations with the data custom scripts, data functions, and calculation functions are used. Custom scripts are written using C# programming language. The script consists of data and calculation functions to perform various operations on the data. Data functions can be for example “num” which is used to convert decimal values to numeric values or “bln” which is a Boolean function testing if a value is meeting specific conditions. Calculation functions are pre-made functions that are usually more complex such as function for calculating entropy or values above or below a specific threshold value. [9]

Advanced controls

Trend

A trend is a control used to represent both run-time and historical data in the form of graphs. A Trend Panel containing a trend is displayed in figure 12. The trend component can display up to six graphs and it is possible to trend any property. Trends are usually used to look for immediate variance in the expected production. Functions for rulers, time zooming, etc. are available in order to analyze the trend data. The trend control can work with data from several different data sources. The data for example from cpmPlus History logs is collected and stored in the Smart Client server. [9]



Visible	Object	Property	Limits	Scale	Marker	Range Top	Range Bot	Style	Baseline	Colo	Width	Marker Si	Filter Typ	Filter 1	Filter 2	Extrap	Mode	Data	Interval
☑	OEEINPP2	Value	□	Fixed	None	100	10	Interpolate	10	Color	2	4	Lowpass fil	0.8	0	□	Time av	Interpolate	0
☑	OEEINPP3	Value	□	Fixed	None	100	0	Interpolate	10	Color	2	4	None	0	0	□	Time av	Interpolate	0

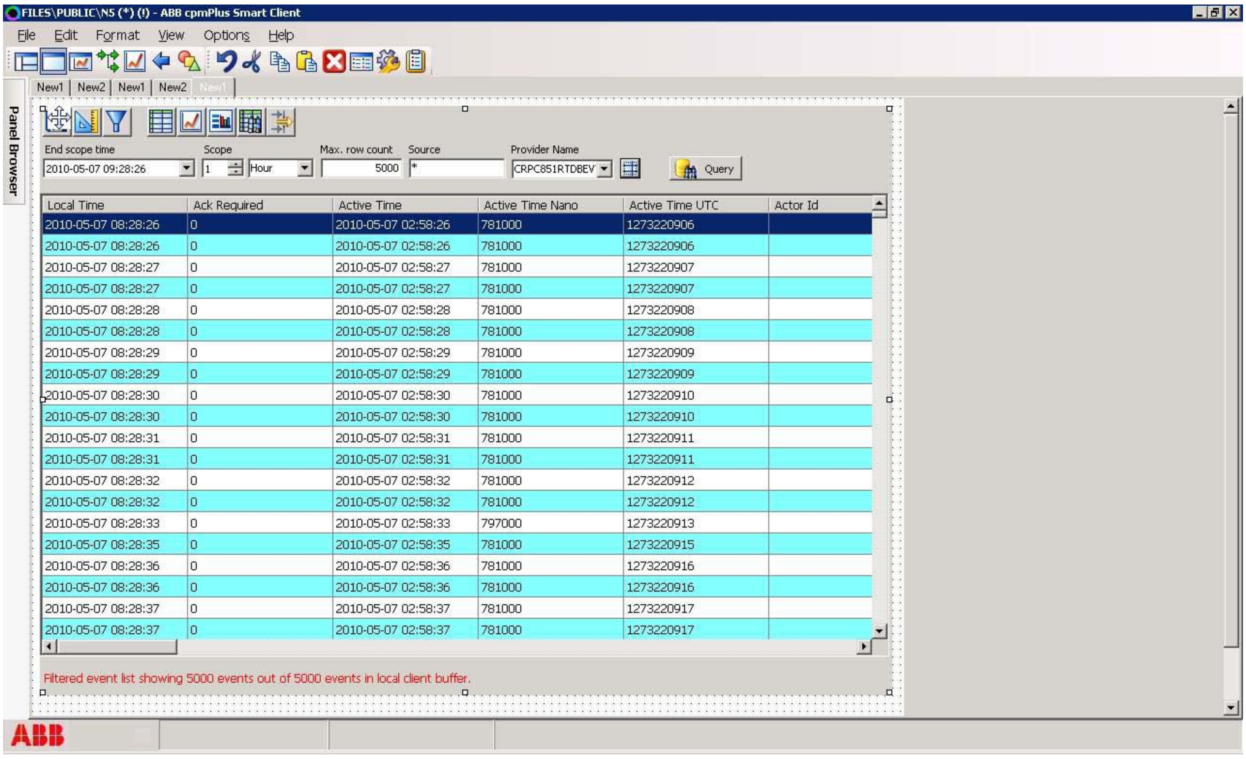
Figure 12: Trend Panel

SPC

Statistical Process Control (SPC) charts provide a set of charts to graphically analyze process behavior and patterns. Even if the production is running normally there are always variances in the process properties. SPC charts are used to analyze processes statistically, in order to control production. By using SPC, the manager of a plant can discover cause of problem, and correct the problem before producing product that do not meet the customer specifications. To set up the trend or SPC the user needs to add a trend or SPC template from the file menu. Then the trend needs to have a data source called a trace. Traces and the data they contain can be dragged and dropped from Property Browser located in Standard Toolbar to the trend panel. [9]

Alarm and Events

An alarm is an event that alerts the user about an abnormal state in the process and needs to be acknowledged. An event on the other hand is an occurrence, for instance, the object's transition in or out of high level alarm condition (as mentioned before object is the continuously updated information associated with a UI control). Events list is displayed in figure 13. Events are the focus analysis which is performed with the alarm and event analysis tool. These events are logged on a cpmPlus History Server. [9]



End scope time: 2010-05-07 09:28:26
Scope: 1 Hour
Max. row count: 5000
Source: *
Provider Name: CRPC851RTDBEV

Local Time	Ack Required	Active Time	Active Time Nano	Active Time UTC	Actor Id
2010-05-07 08:28:26	0	2010-05-07 02:58:26	781000	1273220906	
2010-05-07 08:28:26	0	2010-05-07 02:58:26	781000	1273220906	
2010-05-07 08:28:27	0	2010-05-07 02:58:27	781000	1273220907	
2010-05-07 08:28:27	0	2010-05-07 02:58:27	781000	1273220907	
2010-05-07 08:28:28	0	2010-05-07 02:58:28	781000	1273220908	
2010-05-07 08:28:28	0	2010-05-07 02:58:28	781000	1273220908	
2010-05-07 08:28:29	0	2010-05-07 02:58:29	781000	1273220909	
2010-05-07 08:28:29	0	2010-05-07 02:58:29	781000	1273220909	
2010-05-07 08:28:30	0	2010-05-07 02:58:30	781000	1273220910	
2010-05-07 08:28:30	0	2010-05-07 02:58:30	781000	1273220910	
2010-05-07 08:28:31	0	2010-05-07 02:58:31	781000	1273220911	
2010-05-07 08:28:31	0	2010-05-07 02:58:31	781000	1273220911	
2010-05-07 08:28:32	0	2010-05-07 02:58:32	781000	1273220912	
2010-05-07 08:28:32	0	2010-05-07 02:58:32	781000	1273220912	
2010-05-07 08:28:33	0	2010-05-07 02:58:33	797000	1273220913	
2010-05-07 08:28:35	0	2010-05-07 02:58:35	781000	1273220915	
2010-05-07 08:28:36	0	2010-05-07 02:58:36	781000	1273220916	
2010-05-07 08:28:36	0	2010-05-07 02:58:36	781000	1273220916	
2010-05-07 08:28:37	0	2010-05-07 02:58:37	781000	1273220917	
2010-05-07 08:28:37	0	2010-05-07 02:58:37	781000	1273220917	

Filtered event list showing 5000 events out of 5000 events in local client buffer.

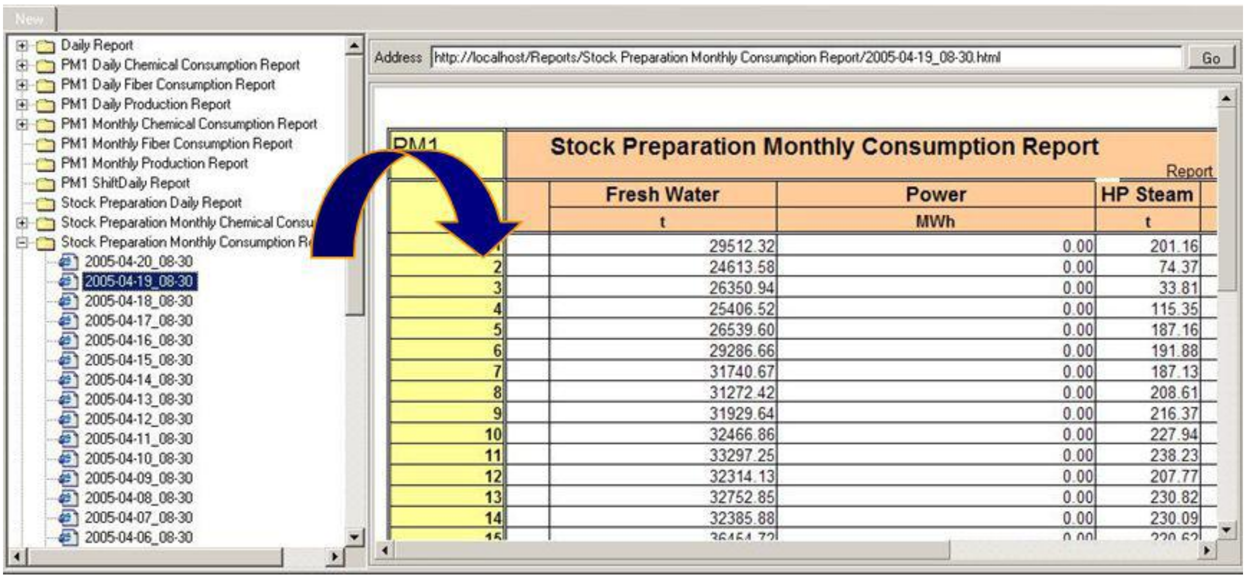
Figure 13: Alarm and Event Panel

An Alarm and Event panel can be dragged and dropped from the **Standard Toolbar** to the **Workplace**. The panel consists of a toolbar and a main display window. The toolbar consists of icons that are used to filter, manage and acknowledge the alarm conditions. The main window displays the results of an alarm filter. The data can also be exported to Microsoft

Excel for further analysis. [9]

Scheduled Reports

Reports give a summary of the data over a certain period of time. In order to view a report a Report Viewer panel must be added to the **Workplace** which is displayed in figure 14 with an example report included. The user can then select the desired report by dragging and dropping it from the Report List. Now the report will be visible in the Report Viewer. Also now the report is ready to be printed if needed. [9]



Daily Report
PM1 Daily Chemical Consumption Report
PM1 Daily Fiber Consumption Report
PM1 Daily Production Report
PM1 Monthly Chemical Consumption Report
PM1 Monthly Fiber Consumption Report
PM1 Monthly Production Report
PM1 ShiftDaily Report
Stock Preparation Daily Report
Stock Preparation Monthly Chemical Consu
Stock Preparation Monthly Consumption R
2005-04-20_08-30
2005-04-19_08-30
2005-04-18_08-30
2005-04-17_08-30
2005-04-16_08-30
2005-04-15_08-30
2005-04-14_08-30
2005-04-13_08-30
2005-04-12_08-30
2005-04-11_08-30
2005-04-10_08-30
2005-04-09_08-30
2005-04-08_08-30
2005-04-07_08-30
2005-04-06_08-30

Address http://localhost/Reports/Stock Preparation Monthly Consumption Report/2005-04-19_08-30.html

PM1	Stock Preparation Monthly Consumption Report			Report
		Fresh Water	Power	HP Steam
		t	MWh	t
1		29512.32	0.00	201.16
2		24613.58	0.00	74.37
3		26350.94	0.00	33.81
4		25406.52	0.00	115.35
5		26539.60	0.00	187.16
6		29286.66	0.00	191.88
7		31740.67	0.00	187.13
8		31272.42	0.00	208.61
9		31929.64	0.00	216.37
10		32466.86	0.00	227.94
11		33297.25	0.00	238.23
12		32314.13	0.00	207.77
13		32752.85	0.00	230.82
14		32385.88	0.00	230.09
15		36454.72	0.00	220.62

Figure 14: Report Viewer Panel

3.3.Challenges in the Case Company: Varied range of business units

Another challenge in requirements gathering was this project is that the requirements were gathered from several different business units of the Case Company. They all have their own industry specific area of business, different ways of operating, and globally responsible CoEs (Center of Excellence) for CPM are geographically distributed to different countries. The purpose of this chapter is merely to display the diversity of business units to give an idea about the wide scope of this study.

The CPM operations of the following business units took part in the requirements gathering project (location of the CoE for CPM in parenthesis) [18]:

- Service / Energy Management for all industries (Finland)
- Pulp and Paper (Finland)
- Marine (Finland)

- Oil, Gas, and Petrochemical (Norway)
- Minerals and Mining (Switzerland)
- Control Technologies / Life Sciences (Denmark)

Not all the business units of the Case Company were included in the study. Below there is a list of these business units [18].

- Full Services
- Metals
- Measurement Products
- Turbocharging

Since there are many business units in the Case Company which will be affected by the results of this project it is essential for the success of the project that as many business units as possible get the feeling that they have had the chance to say their opinion. [15] Therefore the effects of leaving out several business units that should also be part of the process will be seen in the later phases when a system is being developed according to the requirements.

4. Research material and –methods

4.1.Requirements gathering plan

The goal of the project done for the Case Company was to define if it was possible to unify the user interface development efforts in the Case Company. My task was to design a process to find that out. While designing this process the challenges in requirements gathering presented in chapter 3 were taken in to consideration. Designing the process was mainly driven by keeping the diverse requirements comparable, managing time restrictions set by the project and managing my lack of domain expertise (compared to domain experts present in the workshops). How the process answers these challenges is further explained in chapter 5. The process of finding a solution for this problem is displayed in figure 15. In order to find this out 2 different research methods were used for gathering the required data to act as a support for decision making. The first method used for gathering the data was a brainstorming methodology used by a user experience development company called Linja [22]. The second method used was based on user-led requirements constructions. The data was gathered from the Case Company employees in several business units involved in activities concerning process automation applications used by several industry customers of the Case Company. These people involved on in the study ranged from software engineers to sales staff.

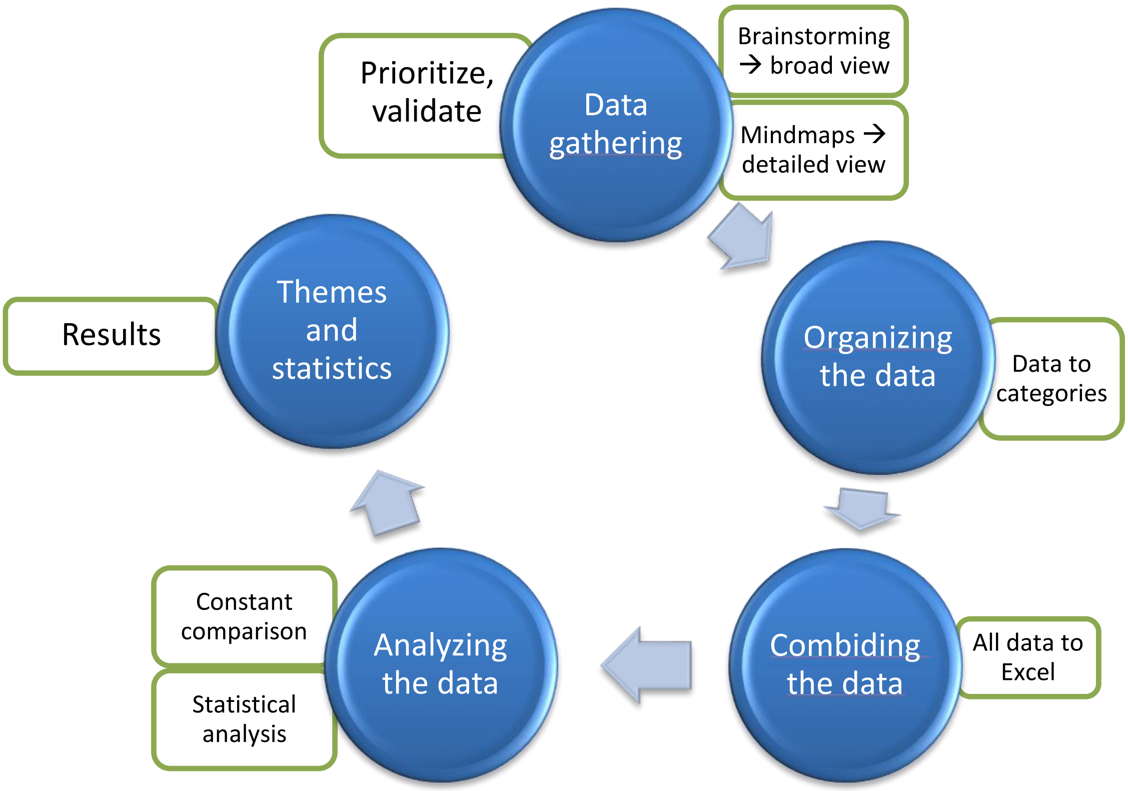


Figure 15: Whole process for gathering, organizing, combining and analyzing the requirements

After gathering the data it was time to organize and combine all the data to make the analysis phase easier. The data from brainstorming workshops were put on excel and categories based on categories defined in subsection 4.3.2. The data from requirements gathering were already in categories so the data was copied and pasted to excel. The analysis for the data was handled using 2 methods: constant comparison theory and quasi-statistical analysis. Constant comparison theory was used to visually look for repeating patterns in the data whereas quasi-statistical analysis was based priorities of individual requirements defined by a prioritization scheme that is presented in section 4.2. These efforts resulted in Themes describing common factors among requirements from different workshops and statistics showing how different business units valued individual requirements.

4.2.Evaluation criteria for the gathered requirements

Brainstorming workshops

The priority of ideas on brainstorming workshops is based on a voting scheme. There the participants vote for most important ideas in 3 different categories which are context, benefits and enablers. These categories will be explained later in subsection 4.3.1. The focus in the voting scheme is to vote for ideas that are seen as important now and in the future and they have enough depth in participants' opinion to be examined further in the workshop. In the voting scheme everyone has 3-5 votes depending on the number of ideas generated in each stage of the workshop. It is not realistic to keep the number of votes each person has the same since the amount of ideas vary a lot during the workshop. Each participant can only put one vote on each item they wish to vote for. Participants cast their votes by marking their vote next to the ideas which are written on post-it notes attached to a flipchart. As a result the most important ideas are decided based on the number of votes each idea received. [20]

Even though if an idea didn't get enough votes any participant can explain why that idea is important and if others agree the idea can be taken into further examination. It must be emphasized that the main purpose of this voting scheme is to decide which items presented by the participants were most important and interesting and therefore need further examination during the workshop.

Requirements gathering workshops

Prioritizing requirements in requirements gathering workshop sessions is based on prioritization scheme which will be presented next. It consists of 4 different dimensions: current state, frequency of use, business impact, and relevance for end-user. The level means the level of importance of an individual requirements for the participants.

Table 1: Prioritizing scheme in requirements gathering workshops

Dimension / Level	1	2	3
Current state of functionalities	Missing	Partly available	Very well supported
Frequency of use	Rarely used (< 10 %)	Frequently used (10-60 %)	Used all the time (> 60 %)
Business impact	No major impact	Key sales attraction	Real differentiator
Relevance for end-user	Not relevant	Optional	Mandatory

This prioritizing scheme was formed as a result of discussion between me, and two experienced employees of ABB with extensive experience of the process automation domain. It was essential that they were involved in the creation of prioritization scheme since due to their deeper understanding they could say what are the most important factors by which the requirements should be evaluated with.

Current state represents how extensively a feature has been implemented and how well it is supported currently. Different options are missing, partly available, and very well supported. If the functionality presented in the requirements doesn’t exist at all it is categorized as missing. If the functionality exists but it is not so well supported in the Case Company’s user interface development it is categorized as partly available. If the feature is very well supported there no need to offer further support.

Frequency of use represents how often a feature is being used during a customer project. If it is used rarely it means that it is used under 10 % of the time during a project. If it is frequently used it is used from 10 % to 60 % of the time during a project. If the feature is used all the time it is used more than 60 % of the time during a project.

Business impact represents how big of an impact a feature has in a situation where the software is being presented and sold for a customer. If the feature doesn’t have any relevant business impact it can’t really be used to attract customers to boost sales. If the feature is however a key sales attraction it is important for sales but still it is possible that competitors also have it. If the feature is a real differentiator it is very important for sales and none of the competitors have it which results in a clear competitive edge compared to competitors.

Relevance for the end-user describes how important a feature is for the end-user in doing his work. However it must be mentioned that there are many categories in the mind map and each category has different importance for different user groups. Therefore the end-user in this case is the user the one to whom this feature is important to. If the feature is not relevant for the end user it doesn’t really matter if it is there or not in the end-user point of view. If the feature is optional it is very useful for some end-user making their work more efficient but they can still do it without it. If the feature is mandatory it must be included since without it end-users can’t achieve their goals in required manner.

There are many different ways presented in the literature on how to prioritize requirements. My domain understanding is not at the same level as the domain understanding of the people who took part to the workshops. Therefore the importance of developing a simple prioritizing scheme with my highly experienced colleagues was very important. Also in the project that I did for the Case Company the time for gathering the requirements was limited to one day for each workshop. Therefore in practice there wasn’t time to have a very sophisticated prioritizing scheme.

For example a method called AHP is seen as most reliable by work of Mead, N.R. [21], but on the other hand it is the most time consuming. This method is based on pair-wise comparison matrix. In that method each requirement is put on Y and X axis and compared to each other. The users are asked to evaluate how much more important a requirement is to its comparison pair on a scale from 1 to 9. On the scale 1 means equally important and 9 means extreme difference in importance. In this study this kind of comparison will be too time-consuming, therefore less exhaustive methods must be used. Then again the problem in less exhaustive methods is of course their unreliability due to the fact that they can’t be very thorough. Also another factor that can create incorrect priorities is the fact that in the workshops there are only a few representatives from each business unit which means that they might consider some requirements as not important for them while those requirements might be important for others not present in the workshops.

In order to get reliable results the requirements gathering process requires more control from the workshop moderators. The control is imposed by having a diverse but simple priority scheme presented earlier. Furthermore in order to make sure that the priority is correct there will be a very short discussion around requirements which priority doesn’t reach consensus among the participants. In addition to the previous points the requirements will be sent to other members of the business units so that they can also correct the priorities if some of them are seen as incorrect.

4.3.Data gathering: methodology and implementation

In this chapter the methods used for requirements gathering are presented and their use is explained in detail. The methods used are brainstorming workshops and requirements gathering workshops.

The main idea behind brainstorming workshops is to get a broad view of how different business units of the Case Company see the future direction of process automation application business. Also here the participants’ thinking is not restricted in any way which can lead to innovative ideas but it doesn’t guarantee that all the relevant points are covered.

Requirements gathering workshops on the other hand offer a specific view on what is required by each business unit currently and in the future of process automation application business with the help of its predefined categories. Unlike brainstorming workshops the requirements gathering workshops restrict participants’ thinking by using predefined categories for the requirements which will guarantee that all the relevant areas of the domain are covered but the ideas might not be so innovative. Also in requirements gathering workshops it is possible to focus on relevant issues brought up in the brainstorming workshops

4.3.1. Brainstorming workshops

The methodology used in the brainstorming workshops is used by Linja in their projects for defining the type of interface customers are requiring. Linja is a user experience development company focused mainly on creating graphical user interfaces. [22]

The purpose of brainstorming workshop is to find high level business requirements that describe the true needs of different business units of the Case Company related to user interface development for process automation applications. The purpose is also to find new kind of functionalities for user interface development tools in order for the developers to do their work better in the future. The functionalities should however not be described in detail just merely on the level to give an idea about the direction of where the application development is going in the future for each business unit.

The idealistic end-goal is to find highly innovative ideas that would open a whole new market without much competition. This would give the opportunity for highly profitable growth. In other words the true goal behind these brainstorming workshops can be described as finding a “blue ocean strategy”. Traditionally the dominant focus of strategy work over the past twenty-five years has been on competition-based “red ocean strategies”. “Red ocean strategies are categorized as finding new ways to cut costs and grow revenue by taking away market share from the competition. The focus in “blue ocean strategies” is totally different from “red ocean strategies”. The idea is to avoid competing in existing markets. Instead the idea is to create a totally new uncontested market. This way the focus wouldn’t be in the strategies on how to beat your competitors. Instead the whole aspect of competition becomes irrelevant. In the “blue ocean” strategy the purpose is not to exploit existing demand. Instead the purpose is to create and capture new demand that no one else has thought about before. Since in “blue ocean strategy” the purpose is to create a completely new market the value/cost trade-off dilemma becomes irrelevant. Traditionally the strategy by which the organization should align its value generation processes is to choose either low cost or differentiation strategies, not both. The ideology in the “blue ocean strategy” is quite different. The freedom of opening a completely new market makes it possible to focus on achieving both low cost and differentiating oneself from the competitors. [23]

Choosing the right users

The purpose of brainstorming workshop is to get new ideas how user interface development tools in the Case Company could operate in the future. Therefore to get the best results it requires certain characteristics from the brainstorming workshop participants. The participants should have deep understanding about the business behind user interface development in the Case Company. Also the participants should have enough understanding about the whole application life-cycle of the Case Company’s applications for example the application development and interaction with the client including sales and support services. The reason for this is that the business requirements define what things will be implemented in the end. Therefore the business requirements should be known first to help to define what is actually relevant. Also the people holding a managerial position have the big picture of the whole operations. They have both the power and the understanding on how things could be in the future which makes them ideal participants as well.

Since the purpose of the brainstorming workshops was to look for ideas that could open a totally new market without much competition participants should have positive attitude towards change or at least see a change in the current working practices as a relevant possibility if it can improve their business and way of working. The criticism and feasibility considerations will be done later in this project so at this point the focus is to generate a lot of innovative ideas without criticism.

Linja brainstorming methodology

Linja brainstorming methodology can be shortly described as being a user-centric brainstorming method focusing on getting a wide variety of concrete ideas throughout the whole innovation space without being too exhaustive. First the methodology is being presented which will be followed by short analysis of the methodology itself.

The brainstorming workshops take place in a meeting room where everyone is gathered around a round table. In the beginning of the workshop the moderators will present the agenda of the workshop showing the schedule on how the workshop will proceed.

It is emphasized that since this is a brainstorming workshop certain rules apply: criticism is forbidden, being exited is desired, every idea is documented, the best ideas are voted, an idea can be presented any time, and finally mobile phones and laptops must be kept shut down so everyone can fully focus on the workshop. The idea is to help people understand the nature of the workshop. In order to make the rules clearer a few examples are presented. “do not decide yourself whether idea is bad”, “Don’t try to be too smart”, “Don’t try to be too original”, “Don’t say ‘yes but…’ instead use ‘yes and…”. These examples also indirectly show that the atmosphere should not be rigid and also that the workshop can be an enjoyable experience. [20]

The brainstorming workshops are focused around three main categories: context, benefits and enablers. Contexts can be offerings, applications, user roles, use cases, client business, organizational viewpoint, development viewpoint, or anything that helps to open up the key characteristics of particular domain which the workshop participants are representing. [20] The contexts in the brainstorming workshop are divided in 3 categories: end-user, project engineering and application development. End-user context describes different roles, characteristics and activities related on people who use the Case Company’s process automation applications. Project engineering context is related to all the activities between the Case Company and the customer including sales and lifecycle support. Application development context relates to activities related to developing the Case Company’s process automation applications. The benefits on the other hand describe the benefits for the users, clients or for the organization that can be generated in a specific context. And lastly enablers describe the means how to make a specific benefit possible in a specific context. These enablers can be features, methods, systems, guidelines, processes to name a few.

Next the process of the brainstorming workshop is presented. The process is presented in figure 16. [20]

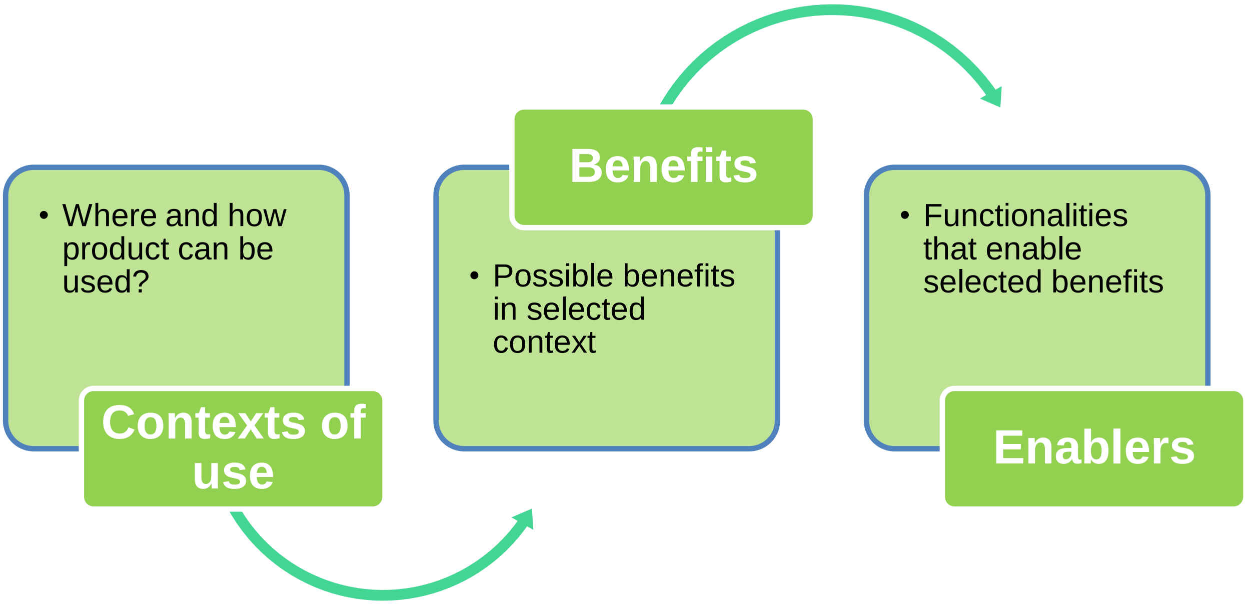

• Where and how product can be used?
Contexts of use
Benefits
• Possible benefits in selected context
• Functionalities that enable selected benefits
Enablers

Figure 16: Process of Linja brainstorming methodology

In the brainstorming workshop there are 2 moderators who control the workshop. First moderator 1 presents the agenda to everyone and explains how the workshop process works. Later on the role of the moderator 1 is to host the workshop and make sure that the brainstorming is flowing smoothly and new ideas are being generated. This is done by asking questions about interesting topics that came up in the discussion to generate further discussion or posing new ideas that might be relevant for the domain. Since the moderator is not a representative of the domain his suggestions can facilitate different kind of thinking in the participants. Moderator 1 also controls how the workshop proceeds by suggesting about going to a different topic when discussion around the current topic begins to diminish.

The role of the moderator 2 is to help the moderator 1 in practicalities such as simplifying the ideas that are said in the discussion, writing them down on post-it notes, and attaching them on the flipchart. This allows Moderator 1 to fully concentrate on keeping the workshop atmosphere energetic and innovative. Moderator 2 also takes care that the workshop is on schedule and that there is still time to go through the most relevant topics.

The session starts by brainstorming who are the different users in a particular domain that the workshop participants are representing. Moderator 2 writes them down on post-it notes which are attached on flipcharts. The purpose of brainstorming different users is to refresh participants’ memory about what is actually important in their domain and also to help the moderators understand what the domain is really about. Another reason is that this acts as a warm up for the actual brainstorming session. This kind of easy warm up is important since it gives the participants confidence on the actual brainstorming since it is possible they have not done anything like this before.

The actual brainstorming of high-level business requirements and possible key functionalities in the future starts by brainstorming different contexts related to the domain. These contexts are divided in 3 categories: end-user, project engineering and application development as described before. Participants start to verbally present different contexts while moderator 2 writes the key points of the discussion on post-it notes and attaches them on a flipchart. Moderator 1 can also say in what form the idea should be put on the post-it if he can crystalize the idea in a better way. This continues until most of the contexts have been covered or the discussion diminishes. Now there is a flipchart full of post-it notes with ideas on them. Next step is to prioritize these ideas to see to which context the participants want to brainstorm benefits to. Each participant puts a mark on a post-it he wants to vote for. After everyone has put their votes on post-its moderator 2 counts the votes and top 5 ideas are announced. From these ideas either moderator 1 or participants suggest a context they want to brainstorm benefits for.

After the most interesting context has been chosen it is time to brainstorm benefits in that context. [20] This is when the participants think of different ideas how they see their business and application development progressing in the future. Another important thing here is to brainstorm benefits that are relevant specifically for this context. Again participants have a discussion and moderator 2 puts the key points of the discussion on post-it notes and attaches them on the flipchart. After it seems that no more new benefits are coming up the moderator 1 decides that is time to vote again. This time participants vote for the most important benefits in the context. As before each participant puts their votes on post-its and moderator 2 calculates which ideas got the most votes. From these ideas moderator 1 or participants suggest a benefit for which they want to brainstorm enablers for.

After the most important or interesting benefit is chosen it is time to brainstorm enablers for that benefit in the specific context where the benefit belongs to. [20] Again there is a discussion among the participants and moderator 2 writes the key points of the ideas that came up in the discussion on post-it notes and puts them on the flipchart. This continues until moderator 1 decides that the ideation activities are diminishing. After that it is time to vote for the most important enablers.

After voting for the enablers it is time to choose another context that got a high number of votes. [20] The same process continues as explained before until it is time to vote for the most important enablers. This process continues in cycles until the time is up and it is time to end the workshop. At the end of the workshop moderator 1 presents a summary of the most important ideas. The summary is followed by a short discussion among the participants which may also result to new innovative ideas or clarification of the old ones. After this moderator 1 presents the next steps of the project and tells how these results will be used and also when results might be expected from this project.

Analyzing Linja brainstorming methodology

The fundamental idea behind this brainstorming methodology is to understand the user and the use context. The user and the use context act as a source of innovation and helping the participants to recall real life cases where they had actual problems with their work. Then the participants can actually focus on innovating ideas that can solve real problems. Because of this characteristic Linja brainstorming methodology can be seen as user-centric brainstorming methodology.

While this method is being used the abstraction level is of the ideas moving back and forth from high level use contexts explaining what and how real end-users do their work to more detailed level of enablers which make a certain benefit possible in a certain context for example in a form of a user interface feature. Figure 17 graphically expresses fluctuation in the process.

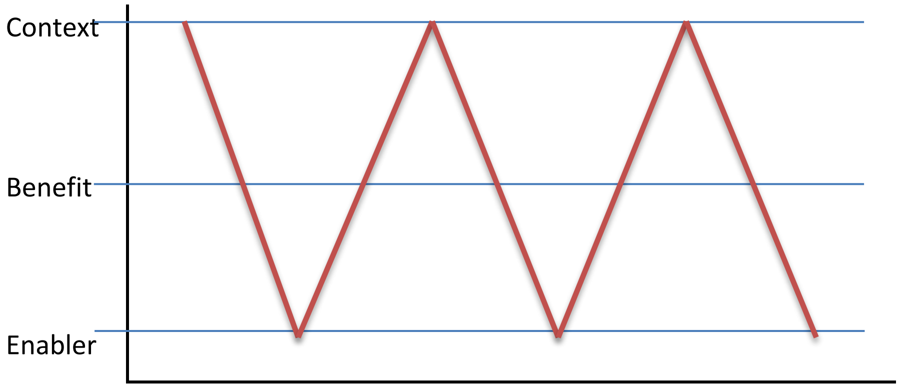


Figure 17: Linja-Design brainstorming methodology characteristics

This kind of approach gives a quite narrow view of the whole potential ideation space it has several benefits. This process gives quite detailed ideas about particular area of interest that was chosen by voting which means that the focus is in the things that matter the most for the participants. Because of greater level of details in ideas it helps to understand the given ideas from many different viewpoints taking full advantage of the best ideas. On the other hand because of this some ideas are left out from the ideation process. Then again the purpose is to get a better understanding about the ideas that give innovative approaches on areas that the participants see as most important for their work.

Another important thing in this methodology is that it helps to distribute the focus throughout the whole potential ideation space. It tries to make sure that very different areas of the whole innovation space are being explored by differentiating the areas of interest on the abstract use context level. Context of use acts as an umbrella concept for benefits and enablers and therefore by choosing very different contexts the ideation is differentiated quite broadly resulting in very different benefits and enablers, in other words very different ideas.

4.3.2. User-led construction of requirements (ULRC)

The used user-led construction of requirements methodology is based on the idea presented in the work of Flynn D.J et.al. [12] In that methodology the users build the requirements themselves. This done by using event flow diagram accompanied with associated training. The requirements gathering consists of 3 phases: training the users how to use the method, build a current model and then build a future model. The idea is to first train the users to know how to use the model based on assignments with the model. The key learning point of this training is to teach the users how to use the graphical notation used to present things in the framework. Then the users build a current model of their domain and refine it on their own time looking for relevant information to support it. In the final stage the users build a future model which addresses the problems that rose from the current model by suggesting improvements to the current model. In this phase users should also look for all the relevant information and communicating with their team members so that all the relevant requirements can be mapped out. [12]

Due to the fast paced schedule of the project done for the Case Company the model used in this work is a lot more simplistic. In the model used for this work the users do all the required steps during one work shop lasting a full working day. The users are taught to use the given framework while they are working with it. Also here creating a current model is skipped and focus is put fully on the future mode. The reason for this is that the current model could restrict participants thinking and it would prevent the formulation of truly innovative ideas. This is important since the true goal is to gather the future trends of the business and operations related to application development in the Case Company. Also by skipping the current model users can more freely think how to do their work better.

After each workshop the built mind map is distributed to be commented and modified for the people working for the particular business unit. The users are told to highlight the changes in the created mind maps using a”💡” -icon in order to differentiate the changes from the original mind map. Next to the symbol the participants were asked to leave a comment about the change and also contact information to clarify the change if needed. It is the job of the participants to further distribute this mind map for their colleagues. After the people in the business unit have formed a final consensus about the content of the mind map it is returned back to the requirements gathering team.

Differences between ULRC methodology and the methodology used in this work

The methodology in this work is based on a generic mind map framework. It has to be generic since more detailed framework will direct users thinking too much and creation of their own view is greatly affected by the framework. By having a generic framework the given categories are working as umbrella concepts for everything the users might say. This way it is easier for users to map their whole domain without forgetting important aspects. In addition the instructor has an already filled framework for each category linked to the main framework which is not shown at first for the users. The purpose of this is that it helps the instructor to keep track that all the relevant things in the domain have been considered. Also this helps to consider all the relevant things and not leave out something out of scope because user didn’t understand what exactly a certain category should contain.

In the ULRC model a specific graphical notation was used to avoid ambiguousness in the presented items. In this work it is not unfortunately possible to do that. One main reason for this is that techniques based on graphical notation do not scale well [12]. When considering the large scope of this project covering several business units of the Case Company it is not practical or even possible to use a methodology based on graphical notations. Another reason is that it is almost impossible to have a notation that can be used to explain the most complex and diverse requirements. Since this project covers many different industries diversity is expected. Because this study is capturing requirements from a wide range of industries a simpler mind map framework is used.

The mind map still offers the benefits of graphical presentation but it leaves more room for ambiguity and therefore it leaves more responsibility for the instructor of the session to make sure that the gathered requirements where understood correctly by the users and the instructor himself. Mind map works as clear communication assistance between the users and instructor of the session. Also it lets users to see what is already been discussed and what topics are there left to go through.

Choosing the right users

Since the focus here is to gather more detailed requirements there should be participants that have enough expertise on actual process automation application development. These participants can offer deep insight on application development due to their technical expertise. Also due to their knowledge about application development the participants understand the relevance of all the details in the interface development tools and their effect on creating the applications. Also there should be people with extensive experience of the business of a particular business unit to understand the decisions behind the technical requirements.

Executing the ULRC methodology in practice

The focus in this workshop is to find what are the things needed from an interface development tool so that the developers can do their work in the best possible way now and in the future. This consists of future trends in process automation application development and the possible future direction of the business from the perspective of each business unit of the Case Company who were included in this study.

This workshop is based on a generic mind map framework presented in detail later in this chapter. In short participants will build a mind map of their requirements guided by moderators. Each participant who took part to the brainstorming workshops also took part to this one. In this workshop the basic idea is that users articulate their requirements in a workshop discussion and the conclusions of the discussions are put on a mind map as requirements. The general guidelines presented for the participants emphasize:

- Focus around process automation application development and the business around it
- Requirements should be for participants' case only. This ensures that the requirements are indeed real requirements and not just beliefs about possible state.
- Requirements should offer means to improve participants' way of working.
- Generalization should be avoided to maintain the context specific nature of the requirements which makes sure that the requirements are in fact real requirements. The generalization will be done later by the analysts.

In practice there are 2 moderators (moderator 1 and moderator 2) and 2-6 participants who gather around a round table in a meeting room. In the meeting room the participants are allowed to have some relevant artifacts or software running on their computer that is relevant to their work. The purpose of this is to help the participants to recall relevant aspects of the applications and increase the probability that all the relevant things will be mentioned during the workshop. In the room there is a projector which is used to project the workshop agenda and a generic mind map framework which also works as an additional medium of communication during the workshop

The role of the moderator 1 is to keep track of the time, construct the mind map and control the conversation so that the discussion stays at relevant topics. Sometimes the conversation can go to a very detailed level on less relevant topics. This is the time for moderator to interfere by forcing the participants to make a conclusion and go to the next topic.

The role of moderator 2 is to keep the conversation up and make sure that all the relevant topics are covered. In order to make sure of this moderator must have a deep understanding about the Case Company's business and application development. The role of this moderator is also to go deeper into interesting topics presented by participants by asking follow up questions.

In the beginning the agenda of the workshop is presented for the participants. The agenda consist of the schedule of the workshop, the purpose, the methodology used and the process how the requirements are gathered. The requirements gathering starts by showing a generic mind map framework via a projector for the participants.

These mind maps have been built using FreeMind which is a simple open-source tool used to build mind maps. The mind map framework is presented in figure 18. The meaning of each category is explained briefly to the participants. If some category is left unclear participants are asked to ask questions to get a clear idea. It is crucial that there won't be misunderstandings about what each category can contain. More detailed explanations of categories can be found from Appendix I

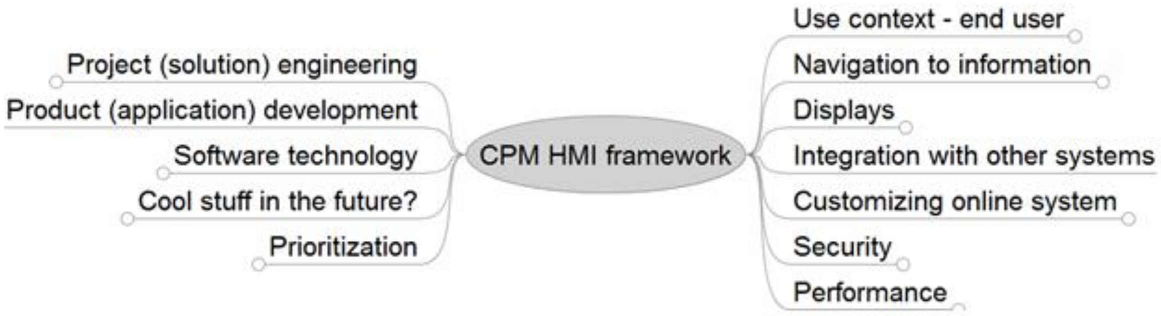

CPM HMI framework
Project (solution) engineering
Product (application) development
Software technology
Cool stuff in the future?
Prioritization
Use context - end user
Navigation to information
Displays
Integration with other systems
Customizing online system
Security
Performance

Figure 18: Predefined categories in requirements gathering workshops

Next the process of the workshop will be explained. This process is shown in figure 19. First a category is presented to the participants and explained what it contains. The "tips" subcategory is used to explain the meaning of the category if needed. Then the brainstorming starts having the focus on ideas around the chosen category. After that "tips" subcategory is checked to see if there are more topics that need to be discussed. Then the items are validated to see that they are in the form required by the workshop participants. After that it is time to prioritize the requirements according to the prioritization scheme presented in section 4.2. After this the category has been covered and a new category is chosen and the same process starts again. Next the process is explained in more detail.

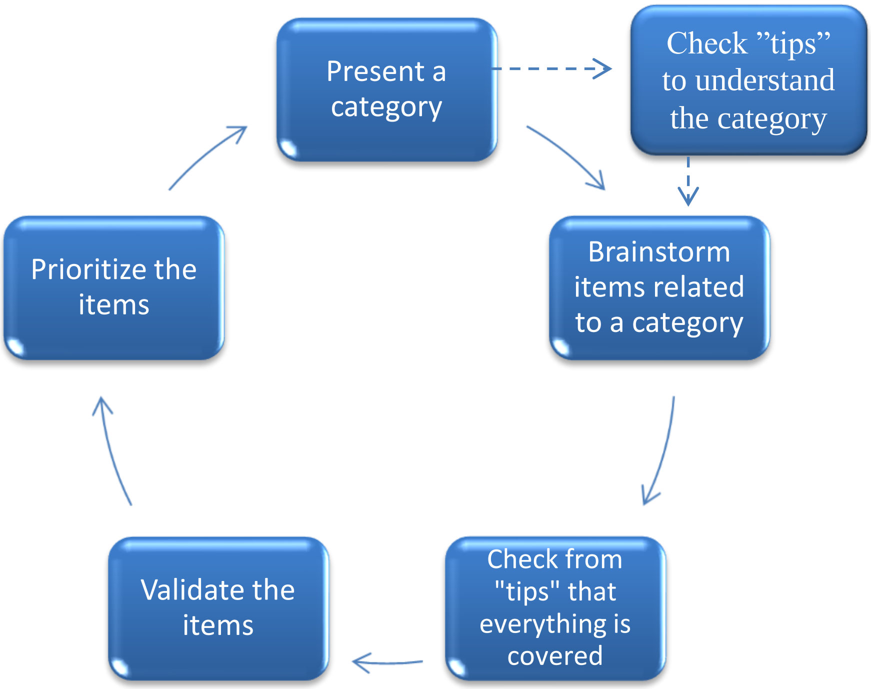

Present a category
Check ”tips” to understand the category
Brainstorm items related to a category
Check from "tips" that everything is covered
Validate the items
Prioritize the items

Figure 19: Process for requirements gathering workshop

Present a category

The requirements gathering begins by moderator 1 who presents a category by opening it from the mind map, which is projected on the screen located in the meeting room. The category contains some key questions that help the participants understand the category. Also the moderator explains the content briefly. In figure 20 the category called “Displays” is presented. In order to understand the “Displays” -category better there are 2 questions: “Which kind of displays your applications provide?” and “Which kind of functionality you expect in displays?” This helps the participants to understand what should be put on this category. For participants answering these questions gives a clear starting point in listing all the relevant requirements for this category.

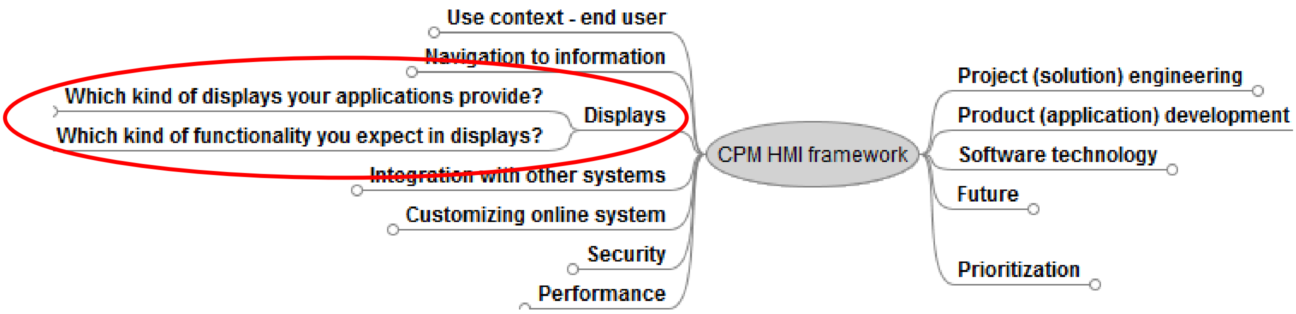

Use context - end user
Navigation to information
Which kind of displays your applications provide?
Which kind of functionality you expect in displays?
Displays
Integration with other systems
Customizing online system
Security
Performance
CPM HMI framework
Project (solution) engineering
Product (application) development
Software technology
Future
Prioritization

Figure 20: Example of questions in the generic mind map framework

If the example questions weren't enough to clarify what should be put in the category it is also possible to open a "tips" -branch from that category. It contains generic sub-categories for this main category. In figure 21 there is a subcategory presented for the category "Project (solution) engineering" as an example.

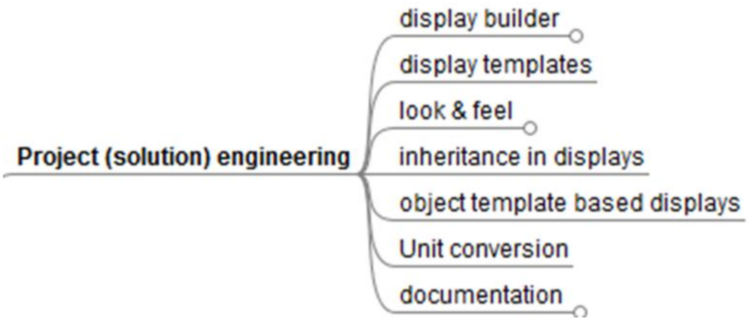


Figure 21: Generic mind map framework sub-categories

By showing the sub-categories from a category it helps participants build a mental model in their minds about what type of requirements this particular category can contain. It must be mentioned that showing these sub-categories will effect on participants thoughts guiding their thinking no matter how generic these categories are. On the other hand it makes requirements gathering workshops more efficient and make it easier to cover a wider scope of requirements.

Brainstorm items related to a category

After the first category has been presented and the participants have understood its possible content the participants start to brainstorm by verbally articulating their needs for the Case Company process automation applications and the business around it. Moderator 2 helps to facilitate this discussion by asking subsequent questions about interesting points that come up in the participants' discussion. While participants are discussing about the requirements the moderator 1 writes relevant points of the discussion to the mind map under the right category on separate branches. At the same time participants can freely comment and say how the content on the mind map should be modified by moderator 1. This way the form of the writer requirements will be exactly how participants want them to be. When the discussion becomes less active Moderator 2 can also present topics that were based on the previous discussion generated by the participants or on his domain expertise. Even though here the process might sound very straightforward in practice the brainstorming will generate ideas that do not belong to the current category that is being examined. Then it is the task of moderator 1 to think to which category it belongs to and put it there. Otherwise great ideas and their form would be forgotten if they are not written down immediately when they are formed. After this the brainstorming continues normally having the focus on the category that was previously under focus.

Check from "tips" that everything is covered

When discussion diminishes moderator 1 can open the "tips" subcategory from the mind map to check that all the relevant topics have been covered. This "tips" subcategory was created by several the Case Company personnel who have deep knowledge about process automation applications in the Case Company. They are the best people to decide what should be covered when requirements are gathered for a certain category.

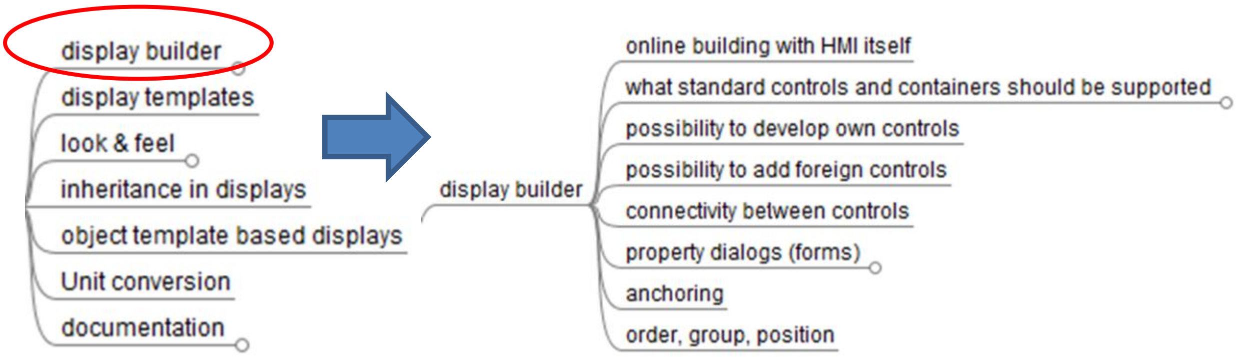

display builder
display templates
look & feel
inheritance in displays
object template based displays
Unit conversion
documentation
display builder
online building with HMI itself
what standard controls and containers should be supported
possibility to develop own controls
possibility to add foreign controls
connectivity between controls
property dialogs (forms)
anchoring
order, group, position

Figure 22: Mind map framework tips categories opened

In previous section this "tips" subcategory was used to present the idea of the category. When the discussion ends moderator 2 can further open this "tips" subcategory to check if all the relevant areas were covered. In figure 22 category "Application (solution) engineering" has been opened to reveal its "tips" subcategory and from there display builder category has been further opened to see all the items that might be relevant for this subcategory. If participants or moderator 2 see something relevant here the discussion continues around the topics not yet covered in the category under discussion.

It is moderator 1's task to make sure that the discussion stays in relevant topics around the discussed category. If the discussion seems to go out of topic the moderator 1 asks participants to crystallize their previous thoughts as requirements. After this the discussion can continue around the current category. If the item put on mind map wasn't understood by the moderators the moderators ask for further information about the requirements from the participants. This further information will be put on mind map as a subcategory for the requirement. Although all the relevant additional information will be put on the mind map in any case as subcategory to decrease its ambiguity.

Validate the items

At the end of each category the moderators and participants check that all the requirements are different from each other and all requirements are relevant. The similar requirements are combined together and irrelevant requirements are removed. It is also checked that the form of the requirements is correct and everyone understands them in the same way. This is done by moderators or participants by very briefly explaining how they understand requirements that they are not sure of. If this explanation differs from consensus a short discussion is made so that consensus can be reached and the final form of the requirement is written on the mind map.

Prioritizing the items

Before moving to next category all the requirements in that category are prioritized using a prioritizing scheme (presented earlier in section 4.2 Evaluation criteria for gathered requirements). Each requirement is prioritized one by one based on their current state, frequency of use, business impact and relevancy for the end-user. Each requirement is evaluation based on these 4 factors on a scale from 1 to 3, where the higher number indicates higher importance. A short discussion is being held to form a consensus among the participants about the priority of requirements. In practice the workshop participants present arguments which act as a proof a certain priority. After everyone agrees on the priority of a requirement for each factor the focus moves to the next requirement in the category. Here the role of the moderator is to keep the discussion about the priorities as short as possible since there are a lot of requirements to be prioritized and time is limited.

Closing the workshop

Before ending the workshop all categories are browsed through to make the final validation. In this phase it is checked that all the requirements are in the right categories and if they need some clarification. If some requirements are in a wrong category they are moved to the right one. If some of the requirements are unclear moderators or participants can present the concern and the form is clarified by either changing the requirements title or adding extra content under it as a subcategory.

4.4.Organizing and combining the gathered requirements

In this phase the data from brainstorming workshops and requirements gathering workshops are combined. The reason for this is to get more comprehensive base for the analysis making the end conclusions more reliable. The data at this point was 10 pages of insights and several flipcharts with post-it notes containing ideas from 4 different brainstorming workshops and 5 mind maps from requirements gathering workshops consisting of several hundreds of items. All this data was put on the same excel sheet. It is important to note that at this point the data from brainstorming workshops and requirements gathering workshops were in a different form and therefore the data needed to be modified before it could be combined. The structure to which the data is put is based on the categories defined in requirements gathering workshops under subsection 4.3.2.

It must be noted here that the items in brainstorming workshops and in requirements gathering workshops have different prioritization scheme. The prioritization scheme in brainstorming workshops was merely based on voting whereas in the requirements gathering workshops the priority scheme was based on 4 different priority categories presented in section 4.2. Therefore it is not possible to combine these two prioritization schemes. Therefore later on in this work there is a separate analysis only for the results of requirements gathering workshops in addition with the general analysis of the combined data.

In order to analyze the data with 2 different methods 2 different types of data structures were defined. The purpose of the first data structure is to put the data in a form that it can be easily analyzed by a method based on constant comparison/grounded theory. The purpose of the second data structure is to put the data in a form that it can be analyzed easily with a method based on quasi-statistics.

In order for the data to be easily used for constant comparison/grounded theory a data structure presented in table 2 was used.

Table 2: Data structure for constant comparison analysis

	W1				W2				W3				W4				W5			
	R	D	F	N	R	D	F	N	R	D	F	N	R	D	F	N	R	D	F	N
C1																				
C2																				
C3																				
…																				
C11																				

On X-axis W is the workshop where the item was mentioned, R is requirement title, D is detailed description of the requirement, F is the number of times a similar item appeared in different workshops, N is the item number. On the Y-axis there is C presenting the category where a particular item belongs to. There can be any number of items inside a category. One thing worth mentioning is that the purpose of the shell D, offering additional information, is to decrease ambiguity of the requirement as well as help to differentiate them from each.

The reason why the requirements are arranged by category and by workshop in this first data structure is that this is the easiest form to visually look for repeating patters between different workshops. It shows as many requirements as possible from all the workshops with a quick glance. Also the purpose for putting description on a separate cell and minimizing it is to get the requirements from each workshop as close as possible to each other to make the visual comparison easier.

In order for the data to be easily usable by quasi-statistical analysis a data structure presented in table 3 is used.

Table 3: Data structure for quasi-statistical analysis

	P1	P2	P3	P4	Sum	F	C	W	Theme	D	N
R1											
R2											
R3											
…											
R237											

X-axis contains the titles of all the different data types contained by the requirements. First there is the priorities of the requirements P1: Currents state, P2: Frequency of use, P3: Business impact, P4: Relevance for the end user. Followed by Sum, which is the sum of P2, P3, and P4 indicating the importance of the requirement. P1 is not included since it tells how well this requirement is supported. The reason for this is that if the feature was very well supported P1=3 it would raise the sum making it more important even though very limited effort would have to be put on achieving only small results. On the other hand if P1=1 saying that the requirement is missing, then it would make the sum smaller making the requirement less important even though it would require a lot effort to support the feature having a big difference. After that there is F showing the number of times a similar item appeared on different workshop, C showing the category where the requirement belongs to, and W showing the workshop where the item belongs to. Then there is Theme which tells the theme where the requirement belongs to. This is formed as a result of constant comparison analysis which is further described in subsection 4.5.1. Theme describes the group of similar requirements among different workshops, and the theme name describes the most important factor among these similar requirements between different workshops. Then there is D giving additional information about the requirement. Finally there is N showing the requirement number. Y-axis on the other hand contains all the gathered requirements listed one by one.

The reason why this kind of data structure was used is that this is the simplest form when exporting a table of items to other statistical analysis programs for example to SPSS statistics. In practice this is the simples form for creating graphs and other graphical presentations from the data.

4.4.1. Organizing brainstorming workshop data

In order to analyze all the data from brainstorming workshops together with the data from requirements gathering workshops the form of the data in brainstorming workshops needs to be modified to a more easily manageable form before combining the data. Even though this data is very diverse it is already in 3 categories: end-user, project engineering and application development. The post-it notes on flipchart express the ideas mentioned by the participants in a few words. Here the purpose of the written notes is to act as more detailed explanation about the content of each idea on post-it notes.

The easiest way to categorize the ideas from brainstorming workshops to the categories defined in requirements gathering workshops is to rearrange the post-it notes since the details from written notes can be attached later. A process for categorizing items from brainstorming workshops is presented in figure 23. The items from each workshop are still kept separately at this point.

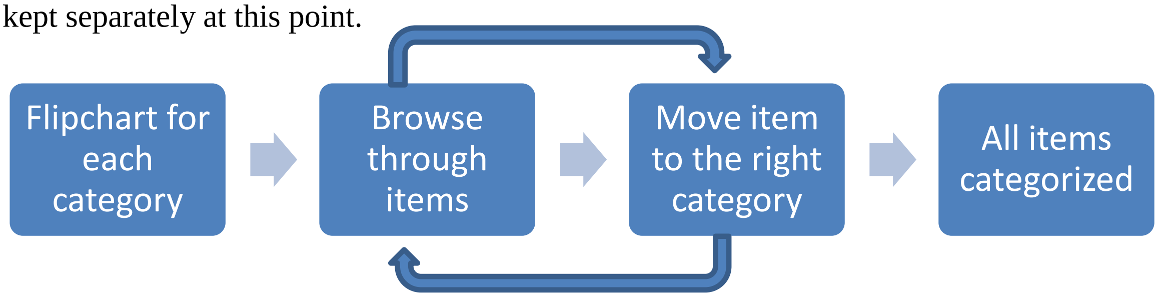


Figure 23: Process for categorizing brainstorming workshop items to requirements gathering workshop categories

The method used to this is called constant comparison/grounded theory [25] which is also used later on this work in the analysis phase in subsection 4.5.1. First step is to get empty flipcharts for each category defined in the requirements gathering workshop, having 11 categories in total. Then by taking advantage of the experiences learned from the workshops and gained domain understanding a visual evaluation for the ideas on post-it notes can be made. The purpose is to look to which category each idea on post-it note belongs to. After seeing that a certain item is clearly connected to a category it is moved to the flipchart dedicated to that category. This process is continues until all the post-it notes are categorized to these 11 categories.

After this the next step is to put the individual ideas on excel sheet under these 11 categories. This is a very straight forward task. Each idea from post-it notes were written on excel sheet under the category where it belonged to. After that the written notes are put on the description shell D as additional information for each item.

4.4.2. Organizing requirements gathering workshop data

In order to analyze all the data together the data from requirements gathering workshops the data needs to be put on the same excel sheet along with the data from brainstorming workshops. The material of requirements gathering workshops consists of 5 mind maps of about a hundred individual items from each workshop. The mind maps consist of nodes which are the requirements and their sub-nodes which are additional information about the requirements.

The requirements from requirements gathering workshops are already in the categories used in the excel sheet which makes it very easy to add them. In practice each mind map category is copied to the excel sheet. After that the sub-nodes of each requirement are copied and pasted to D shell as additional information.

4.5.Analyzing the gathered requirements

At this point all the data from brainstorming workshops as well as from requirements gathering workshops are in the same excel sheet. Also all the data is categorized based on 11 categories defined in subsection 4.3.2. The next step is to start analyzing the data. This categorization helps tremendously in analyzing the data since it divides it in 11 distinctive groups that can be analyzed individually. In practice it means that there is a lot less data to be analyzed at once making it a lot easier to see repeating patterns among the items in different workshops.

The data analysis can be divided in two parts based on the methods used. In the first part constant comparison/grounded theory is used [25] to visually look for repeating patters from the data. In the second part quasi-statistical analysis is used to find some repeating patters based on numerical properties of the data including priority categories and frequencies presented in section 4.2.

4.5.1. Forming themes from the gathered data

A process for visually looking for repeating patters form the data is described in the figure 24. An example how to use this process can be found from Appendix II. Requirements are processed one category at a time going through all the requirements in each workshop. These categories are presented in subsection 4.3.2. The process starts by looking at the titles of requirements and looking if there are some similarities among them. It must be mentioned that they are just titles for people who haven't been part of the project. On the other hand for the people who participated in the workshops the titles open up the whole detailed idea of the requirements.

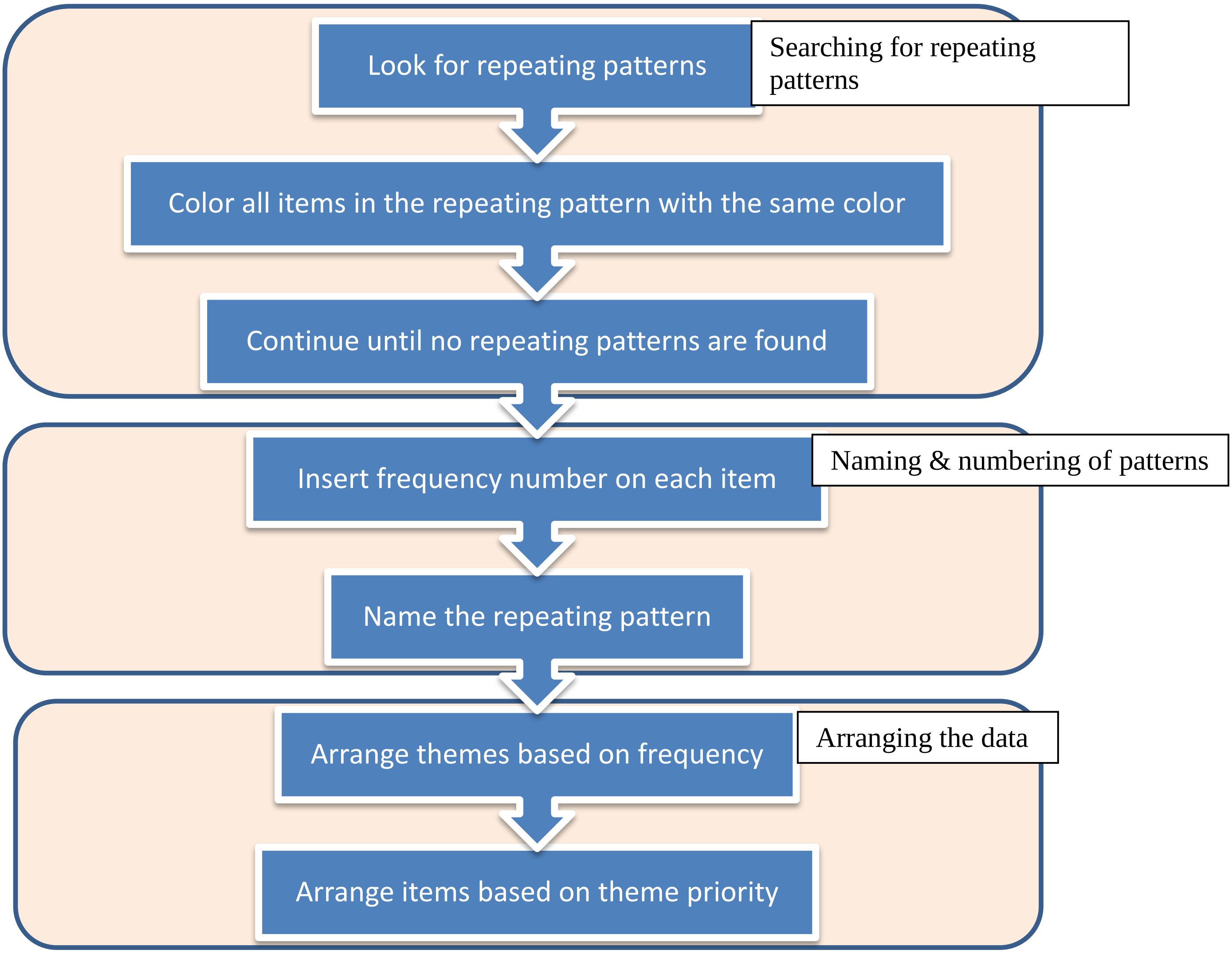

Look for repeating patterns
Searching for repeating patterns
Color all items in the repeating pattern with the same color
Continue until no repeating patterns are found
Insert frequency number on each item
Naming & numbering of patterns
Name the repeating pattern
Arrange themes based on frequency
Arranging the data
Arrange items based on theme priority

Figure 24: Process for analyzing the requirements

When a connecting factor is found between two or more items from different workshops these cells containing the items are colored with the same color. This distinctive color is put on the left side of the table to keep track which colors have been already used and also for the purposes of making summary of the repeating patters later on. Looking for repeating patterns continues until no more repeating patters are found between items in different workshops inside a particular category.

Next step is to insert a number to each requirements frequency (F) column. This number shows in how many different workshops a similar item was found inside a category. It must be mentioned that since there were only 5 workshops the maximum number of times an item can appear is only 5 which means it has no statistical relevance at all. However if a need is expressed almost exactly in the same way in more than two workshops it definitely has some relevance. But still a detailed evaluation with the frequencies should be avoided since the participants in different workshops might see the items they mentioned from different point of views. This means that their understanding of the meaning of the item can be different if it is taken out of the context in which it was mentioned. After putting the frequencies next to the colors indicating the repeating patterns the colors and the frequencies next to them are arranged in descending order having the most frequent on top.

After arranging the colors and frequency numbers attached to them it is time put a title on each repeating pattern that was found. As mentioned before on the left side of the table there are distinctive colors presenting different repeating patterns found from the data. The items belonging to a repeating pattern are browsed through in order to look for the most important connecting factor among them. The repeating patterns are named according to this most important connecting factor. Now these repeating patterns can be called themes and their titles express their names.

At this point all the items inside the repeating patterns are in a random order and therefore for the sake of clarity the items need to be rearranged. The items are arranged based on in how many workshops they appeared. The items appearing in many workshops are put on the top of the table inside the category they belong. The items that only appear in a few workshops are put below these items and to the items which only appeared in one workshop are put on the very bottom part of the table. Also while doing this the items belonging to a theme are kept as close to each other as possible in order to aid the visual inspection of these items later on.

There are several reasons why this coloring scheme is used. It helps to distinguish items presented inside each theme very clearly and decreases the effort for visual comparison. Even though the items might be already arranged some items still might need to be reorganized if they are seen as belonging to a wrong theme. The more organized view of all the items in a category will help to see if the items truly belong under a particular theme or not. This priority in coloring scheme based on frequencies also makes the most important items stand out since they are covering most of the table area with their distinctive color.

4.5.2. Statistical analysis of the requirements

The starting point for statistical analysis is to have data sorted in a way described in section 4.4. There all the requirements are on Y-axis and on X-axis there are all the attributes these requirements contain such as workshop, frequency, and priorities. Statistical analysis in this work is based on these priorities explained in section 4.2. The used priorities are current state, frequency of use, business impact, and relevance for the end user. In practice the analysis is based on looking for features that have a priority level of either 1 or 3 in different priority categories.

Below there are different ways of defining the importance of features listed by the workshop participants. In practice the data is first filtered by workshop to see the differences between workshops. Then the data is filtered based on the first priority category, then by the 2nd priority category, and so on. Finally the data is filtered according to the sum of 2nd, 3rd, and 4th priority in order to put the priorities with highest sum on the top. After this filtering several graphical presentations can be created e.g. bar charts having the scale based on importance of the feature.

The goal is to look for contradicting requirements by examining how differently a specific item is rated in different workshops. The purpose of this is to look what kind of support should be offered to different business units and based on the results what does it mean for the plans of unifying the process automation software development efforts between different business units of the Case Company.

Rarely used ↔ Used all the time

This can mean for example the customer demand in different business units is very different and therefore affecting how much certain functionality is actually needed.

Functionality is missing ↔ Very well supported

This can mean for example that the maturity level of interface development efforts is different in different business units. Therefore the interfaces offer different level of functionalities affecting on how many new functionalities are needed.

No major Impact ↔ Real differentiator

This can reveal key differences on what clients of different business units consider as important, which is very important in selling the Case Company’s software.

Not relevant ↔ Mandatory

This can reveal for example differences in software development practices between software developers in different business units.

Average rating of the items in the category → important category ↔ not important category

For example in one business unit security can be very important for their customers and therefore its importance for the workshops participants is also higher since these are the people meeting the demand of the customers.

Requirements that were very different from others and did not repeat in other workshops

Can reveal key differences in software development between different business units.

It is also possible to look for top features, which are considered as features that have a very high priority in several priority categories. Top features will be listed for each individual workshop as well as for all the workshops combined. Basically a feature is considered a top feature if it got priority 3 from 2 or more in several priority categories.

Very well supported + Used all the time + Mandatory (or 2 pair combination of these)

This shows the features that are very well supported by the user interface development tool, are used all the time by software developers and are mandatory for the end-users. Basically it shows features that must be there while everything else is taken away.

Real differentiator + Very well supported and/or Used all the time and/or Mandatory

This shows the features that are differentiating the Case Company from its competitors when selling products for customers.

Another aspect in analyzing the priorities is to look for must take action features. These consist of features that are missing but they would be important if they existed. Also the features that are rarely used but have a high priority on other prioritization categories can raise questions that need to be answered in order to improve the operations in different business units.

Functionality is missing + (Used all the time / Real differentiator / Mandatory)

Describes the features that do not exist yet but the workshops participants would want them to be implemented in the user interface development tool in order to do their work more efficiently delivering better quality software for the customers.

Rarely used + (Very well supported / Real differentiator)

Shows the features that are rarely used but have a high priority on other prioritization categories. By highlighting these features it can reveal vital information to improve application development efforts.

It is also possible to look for features that must be included. These are the features that might not be interesting in the sense of future considerations but they are features that just need to be there for the software engineers to meet the needs of end-users.

Mandatory + (Rarely used / No major impact)

Describes features that are required by end-users to do their work but they are too common to be useful in sales situation. The features that are mandatory but are rarely used are features that must be included since they are critical in a specific phase of application development.

5. Results

In this chapter results of the work will be presented. One of the results is the process for gathering requirements in multi-site organizations, which is described in detail in chapter 4. This process is presented again in figure 25.

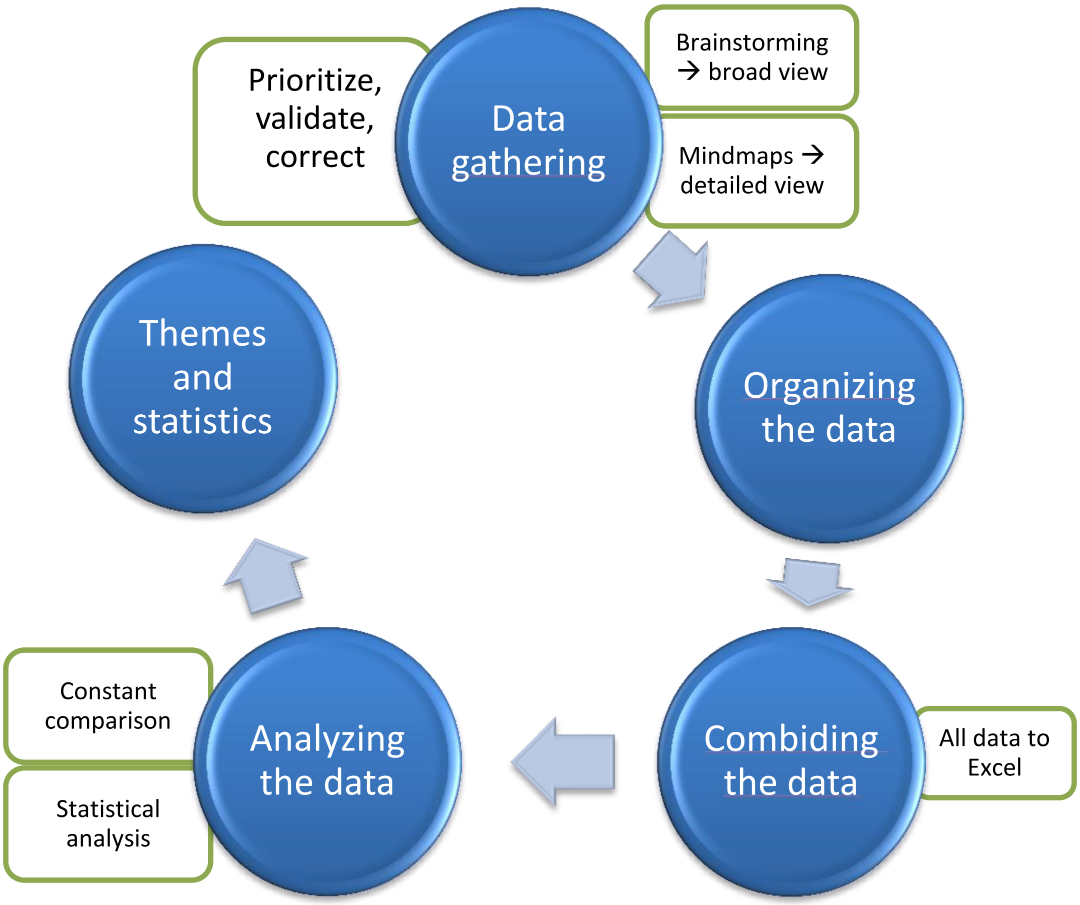


Figure 25: Whole process for gathering, organizing, combining and analyzing the requirements

In the process first the data is gathered using 2 different methods. The brainstorming workshops are used to get a broad, less detailed view to see what is actually important. Then requirements gathering workshops, involving participants creating mind maps of requirements, are used to give detailed information about the areas of highest importance discovered in brainstorming workshops. Even though the focus of these workshops is different both of them reveal important information about what is required. Therefore the results from both workshops are combined to get a bigger dataset to decrease uncertainty as well as give a more detailed view about the requirements. Before combining the results from different workshops the results must be organized to be comparable with each other using the framework offered by requirements gathering workshops (subsection 4.3.2). After that data is analyzed category by category according to the used framework. This way the individual datasets are in a more manageable size to use constant comparison methodology can be used efficiently. This resulted in themes presenting similarities in requirements between different workshops as well as statistics revealing interesting repeating patterns in the data.

Next in this chapter the process of requirements gathering described in chapter 4 is compared to the challenges presented in the chapter 3. Here it is explained how this process meets the challenges in requirements gathering. Also challenges that weren’t met the process are explained here, reasons why they weren’t met, and also what should be done about that.

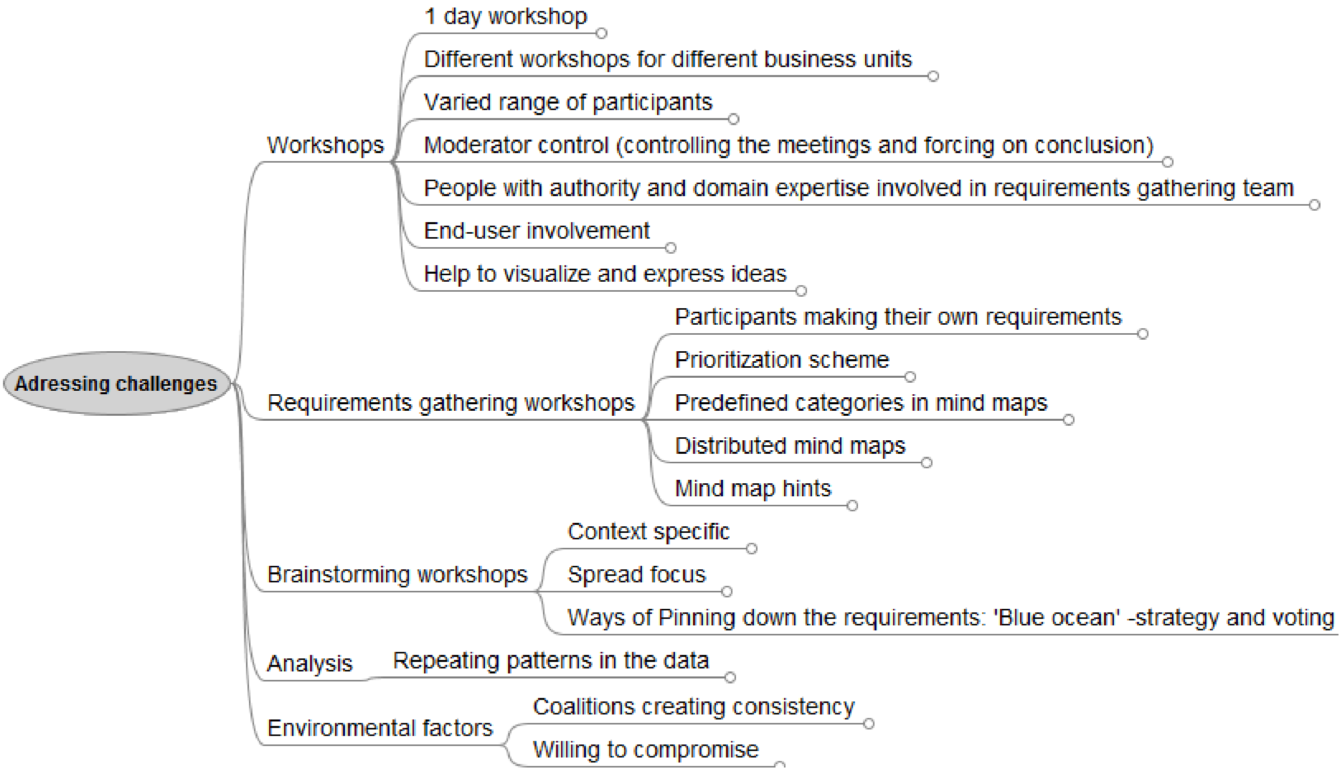

Adressing challenges
Workshops
1 day workshop
Different workshops for different business units
Varied range of participants
Moderator control (controlling the meetings and forcing on conclusion)
People with authority and domain expertise involved in requirements gathering team
End-user involvement
Help to visualize and express ideas
Requirements gathering workshops
Participants making their own requirements
Prioritization scheme
Predefined categories in mind maps
Distributed mind maps
Mind map hints
Brainstorming workshops
Context specific
Spread focus
Ways of Pinning down the requirements: 'Blue ocean' -strategy and voting
Analysis
Repeating patterns in the data
Environmental factors
Coalitions creating consistency
Willing to compromise

Figure 26: Parts of the methodology used in the process of requirements gathering addressing presented challenges

In figure 25 the answers to the challenges presented in chapter 4 are presented. The answers are divided in 5 categories **Workshops**, **Requirements gathering workshops**, **Brainstorming workshops**, **Analysis**, and **Environmental factors**. The reason for answering the challenges based on characteristics of the requirements gathering process is that one single characteristic can address several different challenges. Therefore this approach leads to less repetition. Also this leads to more detailed answers since each challenge is answered considering the viewpoint given by the characteristics of the process.

The **Workshops** -category consists of general characteristics of workshop-approach in requirements gathering and how these characteristics address the presented challenges. It applies to both requirements gathering workshops and brainstorming workshops. The **Requirements gathering workshops** –category consists of characteristics unique to requirements gathering workshops and how these characteristics address the presented challenges. The **Brainstorming workshops** –category consists of characteristics unique to brainstorming and how these characteristics address the presented challenges. The **Analysis** –category consists of characteristics of the analysis methodology used in the presented requirements gathering process and how these characteristics address the presented challenges. The **Environmental factors** –category presents characteristics of the project done for the Case Company which have an effect on the execution of the presented process. Also how these effects answer challenges presented in chapter 3 is examined.

5.1.Analysis on how well the process met the challenges in requirements gathering

In this part of the work there is a discussion about how the challenges in requirements gathering in multi-site organizations posed in chapter 3 are met by the process presented in in chapter 4. The chapter is structured in a way that it goes through the different points of the methodology used in the process. Under each point from the methodology there is discussion about how each point of the process addresses one or several several challenges presented in chapter 3.

5.1.1. Workshops

1 day workshop

A workshop lasting for a one full working day (8 hours with lunch and coffee breaks) gives a great opportunity for full two-way-interaction between developers and users. In this kind of workshop everyone has enough time to express their point of views in full detail increasing the communication between developers and the participants. Also the workshop methods promote cooperation, understanding and teamwork among users and developers. This feeling of doing together helps to build trust among the people in the workshops. Although it must be said that only in workshops there was face-to-face communication but beyond that the exchange of information was restricted to email, phone, and online collaboration softwares. It is possible that this could have restricted the additions to the gathered requirements after the workshop had ended. If the people in the requirements gathering project are in the same office it is easy to just stop by and give feedback quite informally.

The possibility for 2-way communication offers enough time to express different viewpoints and therefore helps to manage conflicts through negotiation. Because of this face-to-face approach the feedback is immediate and therefore the time used for negotiating about the form and importance of requirements decreases. In addition in the workshops the developers had the chance to ask clarifying questions to fully understand the requirements. This way the requirements were correct and clear for both the developers and the participants of the workshops addressing the challenge of participant’s difficulty in articulating their needs in a way that was understandable for developers.

In the brainstorming workshops since the ideas are written on post-it notes at first it required the participants to crystalize the ideas of the requirements and therefore the form is very compact. The details on the other hand are captured in workshop notes by the moderators. Also the ULRC methodology requires participants to crystalize the key idea of what they need to fit it in a mind map nodes and the details are added as sub-categories. All these operations result in more clear requirements with compact form addressing the challenge of unclear requirements.

One critical aspect of these workshops is to give the participants a feeling that they have a chance to effect on the outcome and most importantly this way they get the feeling that they are part of the process. These workshops also inform different business units that a change is coming. This way the participants have plenty of time to get used to the thought of change and adapt themselves accordingly.

Different workshops for different business units

The creation of coalitions (explained in further detail in subsection 5.1.5 Environmental factors: coalitions creating consistency) unifies the opinions and ways of operating inside a business unit. These coalitions help to build trust among the members in an individual business unit which results in workshop participants being more willing to share everything relevant for the creation of requirements. Also this results in more consistent requirements from each individual business units capturing the differences. On the other hand between business units the effect is the opposite: it further differentiates the opinions about what is required in each business unit. Also it is actually possible that different business units can be competing against each other. For example Energy Management business unit can offer better solution for Pulp & Paper business units' customers for energy management than Pulp & Paper themselves. For these reasons mentioned it is the best way is to have separate workshops for each business.

The people in different business units have been operating in a specific way for a long time and therefore it is highly unlikely that they change their opinion to be aligned with some other business units' point of view during one workshop. Therefore putting all the participants from different business units to same workshops would have led to too long arguments about the relevance and the form of individual requirements for which we didn't have time for. Therefore to avoid unnecessary conflicts the only way was to have separate workshops for each business units. Also it is highly probable that common understanding about requirements during one workshop could only be achieved by having separate workshops for each business unit.

Having separate workshops also helped with traceability of requirements. When all the requirements where gathered the information from which business they came from was stored. This guaranteed some level of traceability in the requirements. To be even more accurate the name of the participant who expressed this requirement could be included in the additional information about the requirements. For the purposes of this project this wasn't seen that relevant since the focus was on high-level requirements.

Varied range of participants

In each workshop there were present sales people and managers with business know-how as well as engineers having expertise in software development. Having a varied range of participants made the study more representative taking into consideration many different points of view. On the other hand it resulted in extra work in the workshops. Because of this diversity the workshops participants had to first clarify what they actually needed among themselves and also clarify their need for the moderators. After this they had to define the right form together so it would be understandable for both business people and engineers. If there was any differences in the viewpoints they were solved here leading to a common understanding about requirements. Only after this the individual requirements could be formed and finalized for the moderators to understand and eventually to document them.

Another point is that team approaches such as the workshop methodologies presented in this work make sure that issues of scope are properly addressed by getting the appropriate people involved at the very beginning of requirements gathering. Since there are several business units inside the Case Company consisting of hundreds of people it is not possible to get a consensus otherwise than doing a workshop where only a few people are present. It is already hard enough (very time consuming) to form a consensus inside a business unit let alone forming a consensus between different business units.

Moderator control

All the meetings in this requirements gathering process were workshop –based. A workshop controlled by one or several moderators with clear roles, ensure effectiveness. All the participants were experienced experts on their area which was essential for good requirements. On the other hand they had a tendency to go very deep into the details. Therefore without clear moderator control time would have been spent in arguing about the relevance of details in individual requirements for which we didn't have time for. The moderators make sure that only relevant things are discussed based on their expertise and the mind map framework containing relevant categories.

The moderators also make sure that requirements were clearly defined. This is done by a moderator articulating how he understands the content of a requirement for everyone else when he is unsure about the meaning. Then the participants can correct his understanding which assures that the gathered requirements are clear for everyone.

People with authority and domain expertise involved in requirements gathering team

In requirements gathering social aspects such as knowing who are in the position to make changes in the organization play a key role. Without knowing this the participants might not tell all the relevant things since they might have the feeling that they can’t change things anyway. Therefore it was essential to have 1-2 IT architects present in each workshop. They actually have power and authority to make changes in the way how process automation applications are created in all the business units of the Case Company. Also this gave a lot more credibility for the requirements gathering. Therefore the participants felt that they were talking with someone who could actually make the change possible. In addition because there were people present who could affect the operations in other business units also it was easier for the participants to trust them. In other words these 2 IT architects weren’t thinking what is best of one individual business unit, instead they were thinking what is the best of whole process automation application development in the Case Company. Therefore participants were very motivated in sharing everything what they needed without leaving something out.

Another point is that the quantity of all the data that can be gathered from each 8 hour workshops is way too much to handle efficiently. Because of that only key points should be documented. At the same time a lot of important descriptive information (detailed information about different aspects of individual requirements) is only stored as common understanding between the members of development team and the participants of the workshops. Therefore by including people with power in the workshops makes sure that instead of getting this information indirectly they get it directly and there is less need for documentation. This way they also capture all the finest details of requirements which wouldn’t be possible only by reading documentations about these requirements. For these reasons it is essential that the people with power are present in as many workshops as possible. The reason for this is that these are the people who can make sure that the required changes are implemented.

Another reason for including people with authority is that they have inter-business unit understanding. Requirements were gathered from several very diverse business units having very diverse domain specific languages. In helping to understand the domain specific language of different participants there were always experienced Case Company employees present in each workshop. They had understanding about different terminology used by each business unit and also the differences in their business. The knowledge of these experts helped to avoid misunderstandings about the requirements.

Customer involvement

It is very obvious that the only way to address the problem of not having end-users in the study is to include end-users. Possibly the best way to do this is to interview customers of different business units separately to gather their requirements. If the end-users would have been included in the same workshops the effectiveness of the meeting would have suffered. Also if there was a separate workshop for all customers they most likely wouldn’t want to share their needs to other customers since they could be competitors.

Involving customers would have increased the awareness of the development team about local work context and real problems that end-users have with the Case Company’s process automation solutions’ user interface. This knowledge would have been also useful when there are conflicts about the importance and form of specific requirements. Customer point of view would give confirmation about the true nature of requirements in situations faced by uncertainty. In addition involving customers would have also increased the representativeness of this study Then again it must be remember that the real end-users in this project are application engineers in the Case Company not the people in working for customers using the Case Company’s process automation solutions interfaces.

In this project I actually did a visit to a customer who was producing pulp & paper products. This one visit offered me a lot of knowledge about this particular domain and I could really see the use of the Case Company’s process automation solutions in action. Unfortunately there weren’t enough time or resources to arrange such customer visits for each business unit.

Help to visualize and express ideas

In brainstorming workshops the use of post-it notes and flip charts helps to visualize ideas in a quick and easy way. Also the use of mind maps in requirements gathering workshops help to see what categories are already covered and what kind of requirements have already been mentioned. These approaches make the communication more efficient and clear. In addition especially in the requirements gathering workshops using a graphical presentation such as a mind map makes the requirements easy to read for a wide range of users. [12]

It must be noted that the visualization tools could have been even more advanced though since the experts in the workshops are experts in what they do and not in explaining what they do. For example they could have been instructed to bring some context specific artifacts or software with them. This would have helped to explain their ideas more concretely and clearly was well as helped others to understand the ideas.

5.1.2. Requirements gathering workshops with ULRC methodology

Participants making their own requirements

In my work I had only basic knowledge about each business unit and their operations. I understood this problem before I started the requirements gathering and took it into account in planning of the process. Because of this it was good to have a mindset of an apprentice, [24] which minimized the effect of anchor and adjustment bias. With apprentice approach the requirements gathering team is less likely to have overconfidence about their knowledge leading to wrong conclusions about the nature of requirements. Also because of my lack of domain knowledge I chose the user-led construction of requirements methodology. This way the participants decided the form and language of the requirements. It greatly weakened the researcher's effect which could have resulted in wrong interpretations about the requirements leading to invalid requirements. After participants formed the requirements I could verbalize my understanding about the requirements and then the participants could correct me or add additional information if required. This negotiation helps to achieve a common understanding about the requirements as well as make sure that the requirements are in a clear form. Also since users were the ones controlling the content it results to requirements where the local working context is taken into consideration as far as participants understanding of it goes. In addition the requirements gathering team is forced to consider alternative hypothesis since it can't ignore documenting contradictory evidence for hypothesis since the users are controlling the content.

Also since both in requirements gathering workshops and in brainstorming workshops the participants are the ones forming the requirements their opinion is never overlooked. This answers the problem of participants having difficulties articulating their needs in a way that it would be understandable for the requirements gathering team. If the requirements gatherers just note down their requirements they can entirely missed some requirement but fortunately in the ULCR -methodology this is not possible. Furthermore since the participants have a chance to affect the outcome and they feel included it is more likely that they will convey a more positive image about the change to come to their colleagues in their business unit. Therefore if the Case Company decides to form a new road map for process automation application development, based on this and further research, the different business units will more likely accept the change and conform to it. The reason for this is that they have been part of the process from the beginning and had a chance to effect on the outcome.

Prioritization scheme

In requirements gathering workshops the clear and simple prioritization scheme presented in chapter 4.2 helps to define which requirements are the most important for each business unit addressing the problem of pinning down requirements. Also because the requirements have clear priorities it was a lot easier to manage a high number of requirements since they could be filtered in an excel sheet based on their priority. This also helped a lot to determine which requirements would require the most immediate actions.

Another clear benefit of this prioritization scheme came apparent in the situation of having a high diversity of business units involved in the study. These priorities helped to define how differently business units valued certain requirements. This was very important information in analyzing the similarities in process automation application development in different business units. Another way how priorities help in the analysis phase is when trade-offs must be made. There the value of requirements can be compared based on priorities giving basis for arguments in the analysis phase. Also the priorities diminish the effect of anchor and adjustment bias. The reason for his is that when a hypothesis is formed about a unifying factor between different business units and new information is gained from consecutive workshops the priorities help to define how much this hypothesis needs to be adjusted. If the priorities of new information show high contradiction with the hypothesis it may lead to formulation of alternative hypothesis addressing the problem of confirmation bias. In confirmation bias alternative hypothesis are not sought for.

Predefine categories in mind maps

The framework in user-led construction of requirements methodology is quite generic. More detailed framework directs users thinking too much and creation of their own view is greatly affected by the framework. By having a generic framework the given categories are working as umbrella concepts for everything the users might say. This way it is easier for users to map their whole domain without forgetting important aspects. Therefore the probability of capturing higher amount of relevant requirements increases addressing the challenge of limitations in long-term memory and missing requirements. In addition since the limitation of working memory has less effect the participants are able to collect their thoughts better giving valuable information about the details of requirements. Also this way the participants can rely on the mind map to act as a tool to help in processing their ideas. With the help of predefined categories also the topics that are not currently concerning the participants are being revealed addressing the availability bias.

When the requirements started to pile up the vast quantity and diversity of requirements was controlled by pre-defined categories in user-led construction of requirements methodology. Each individual requirement was categorized while they were gathered making the management of high quantity of requirements a lot easier addressing the challenge of knowledge management. Also these categories helped in managing the diversity of requirements by dividing the requirements from each workshop to the same framework of categories. The fact that all the requirements were filtered to fit this framework also made them more consistent and therefore made them easier to understand for the requirements gathering team and participants as well. These things also address the problem of consistency and common understanding. Because of the predefined categories the requirements were directly comparable inside a category among different workshops. This made it easy for looking for repeating patterns, duplicate requirements, conflicting requirements, and to determine the importance of individual requirements for each business unit.

In requirements gathering workshops the use of predefined categories in mind maps ensured that time wasn’t wasted on thinking what important requirement areas were covered and which were left to be uncovered. During the workshops only a short glance on the screen revealed what was left to be discussed. This made the requirements workshops a lot more efficient answering the challenge of effective meetings.

Another benefit of the predefined categories is that the mind maps store essential information about the requirements in a hierarchical easy to read visual form. Therefore the mind maps act as clear communication tool between the participants and the instructor of the session making forming and communicating the ideas a lot easier. Also because of improved communication it made achieving common understanding of requirements easier. These things mentioned address the challenge of inadequate communication.

Distributed mind maps

In order to make sure that the gathered requirements are relevant for the whole business unit after each workshop the built mind map was distributed to be commented and modified by the people working for the particular business unit. This helped the people inside a business unit to get their opinion heard. This addressed the representativeness bias by actually having a more representative view of a business unit. Also since the study is more representative having more diverse participants it can increase the scope of the requirements. In addition distributing the mind maps offered a chance to form a consensus inside the whole business unit about the requirements addressing the challenge of common understanding. If there was a common understanding then the people in the business unit that weren't in the workshop simply just confirmed the requirements otherwise they modified them. This addressed the challenge of invalid requirements making sure that the gathered requirements also were requirements according to other people in the business unit. In addition distributing the mind maps inside a business unit increased the communication about requirements addressing the challenge of inadequate communication. Also since the awareness of requirements gathering increases it prepares people to expect changes in the future. This knowledge about possible change early on lets people to adapt to the change accordingly. This way the challenge of managing change was addressed.

After distributing the mind maps if the requirements are either confirmed or slightly modified it is ok but the situation become problematic when new requirements need to be added. When new requirements are added communicating about them most likely won't happen face-to-face. This brings up the same problems that the workshops in this requirements gathering process tried to solve. Of course using advanced communication and collaboration software helps but still many of the same problems remain.

Another problem is that in practice since requirements gathering wasn't part of the employees core activities they it is possible that they just agreed on what was said without thoroughly checking the mind maps since no changes were made on the mind maps after the workshops. Better way would have been that the requirements gathering team would have presented that there will be a workshop in which system architects of the Case Company's process automation applications were also present. This way the whole requirements gathering process would have been more visible inside each business unit also providing more credibility. This would have probably resulted in higher level of participation from people inside business units.

Mind map “hints” subcategory

The moderators in the workshops have an already filled framework for each category linked to the main categories. This helps the moderators to keep track of all the relevant things in the domain that have been considered and discussed by the participants. In my case it helped to cope with the lack of domain understanding. By using the “hints” subcategory the moderators can start new topics based on the hints subcategory. This makes the workshops more efficient without wasting too much time on thinking about new discussion topics. This way the participants can also recall less recent incidents that could work as a basis for requirements addressing the availability bias. The “tips” -subcategory help the participants to understand what should be included in a category.

Unfortunately the use of hints category can also lead to invalid requirements since the participants can pick some requirements directly from the hints without thinking thoroughly if they actually really require that. Fortunately when the participants where prioritizing the requirements less relevant requirements were ruled out.

5.1.3. Brainstorming workshops with Linja methodology

Context specific

The fundamental idea behind the used brainstorming methodology is to understand the user and the use context. Ideation around the use contexts helps participants to recall real life cases where they had actual problems with their work in a real context working with software or customers. This helps to cope up with the challenge of limitations in long-term memory. Furthermore when participants are telling stories about their work the information they deliver is in a form that is easy for participants to remember since it is structured by the flow of the story. This helps to overcome the limitations of working memory. Also because of the context specific approach the participants could focus on innovating ideas that can solve real problems resulting in more valid requirements. This addressed the challenge of invalid requirements.

Because of the real life examples there were a lot detailed descriptive information about the nature of requirements. This information can reveal important aspects about the local working context. Furthermore when this detailed information from different workshops is compared it can reveal important aspects about the differences in their organizational culture and business. In addition because of greater level of details in ideas it helps to understand the given ideas from many different viewpoints. This results in more clear and complete requirements. Bigger amount of descriptive information about requirements also helped to avoid ambiguousness in requirements. This resulted in more clear requirements addressing the challenge of unclear requirements.

Since the requirements were almost always presented through real life examples it minimized the chance requirements gathering team ignoring or misunderstanding the requirements. This addresses the challenge of participants' difficulty in articulating their needs. Furthermore these real life examples helped the participants to understand better what they actually required since these examples consisted of real problems. This answered the challenge of participant's being unsure about their needs.

Unfortunately participants telling a story about their work in a meeting room is very limited compared to participants telling about their work while they are doing it. When experienced people are doing their work it consists of many routines they are not aware of. Therefore they are not actually aware of all the things they do. When people are doing their work each step helps them to remember the next step and each action reminds them about the last time they did it. This reveals a lot of details about their work which is valuable in building quality requirements. When people are talking about work they tend to generalize their explanations leaving out relevant details that could be crucial in making quality requirements. When observing others' work it also reveals structure in the work. This structure reveals strategies about why certain actions are taken. It is obviously very important to know the fundamental reasons why certain actions are taken in order to have an optimal user interface for certain context. [24] Unfortunately in the project for the Case Company the time and the vast diversity of participants in the project didn't allow such approach. On the other hand the purpose was to gather high-level requirements and for that case the chosen approach was adequate.

Wide-spread focus

One important thing in brainstorming workshop methodology is that it helps to distribute the focus throughout the whole domain. It tries to make sure that very different areas of the whole domain are being explored by differentiating the areas of interests on the abstract use context level. Context of use acts as an umbrella concept for benefits and enablers and therefore by choosing very different contexts the ideation is differentiated quite broadly resulting in very different benefits and enablers, in other words very different ideas addressing the challenge of scope. Furthermore the confirmation bias is also taken into consideration since the purpose Linja Brainstorming methodology is to brainstorm as diverse ideas as possible around the relevant contexts which can lead to formulation of alternative hypothesis.

Ways of pinning down the requirements: 'Blue ocean' –strategy and voting

In brainstorming workshops the participants are asked to focus on the things that they see as most relevant for their work. This is accomplished by voting for most important contexts, benefits and enablers. Voting helps to see what are the most important requirement groups while the requirements are being gathered. In practice this leaves out many ideas but makes sure that the most important ideas for the participants are highlighted through voting.

Also in the brainstorming workshops the goal was not to make the best possible trade-offs. The goal was rather to look for new innovative ideas that could open a whole new market according to the idea of 'blue ocean strategy' explained in subsection 4.3.1. Because of this the analysis was driven by what requirements should be strategically left out in order to clarify what is truly necessary in order to find this 'blue ocean strategy'. This approach helped in pinning down the requirements to get a clearer idea how to unify process automation application development operations in the Case Company.

5.1.4. Analysis

Repeating patterns in the data

In the synthesis phase the true needs connecting different business units are found based on constant comparison analysis presented in subsection 4.5.1. The constant comparison analysis help to reveal repeating patters from the data gathered from different workshops revealing underlying hidden needs. These underlying needs are still true needs and usually even more important than the needs participants can actually articulate since they reveal insights relevant for all the business units. Also there is plenty of data backing up the findings making the highly relevant. This addressed the challenges of participants being unsure about their needs and difficulties in articulating needs.

Although it must be mentioned that the forming of hypothesis was driven by formulation of similarities among different business unit and this affected the results. It is possible that after some clear repeating patterns were found among business units it might have led to ignoring some less obvious contradicting factors among business units. This is exactly the problem called confirmation bias presented in section 3.2. Fortunately statistical analysis reveals these contradictions since it is not affected by evaluators' opinions.

Statistical analysis

Statistical analysis helps to get a bigger picture of requirements on more detailed level based on priorities set by workshops participants according to prioritization scheme presented in section 3.2. Finding repeating patters only reveals larger themes from the data which do not tell anything about how different business units of the Case Company value individual requirements. Statistical analysis also works as a tool to check that also individual requirements are aligned with the themes found with constant comparison analysis.

Presenting data via charts generated with statistical analysis is a good way to convey understanding about the results of the requirements gathering project. This answers the challenge of knowledge management. Also seeing what requirements are most valued by different business units helps to see what requirements are most important in general. This answers the challenge of pinning down requirements.

Another good thing in statistical analysis is that it is free from all the biases presented in subsection 3.1.3 such as availability, anchoring and adjustment, overconfidence, and confirmatory evidence.

Also since the participants where the one prioritizing the requirements it gives a bit deeper understand what participants actually want based on the priorities of current state, frequency of use, business impact, and relevance for end-user. In the analysis the most interesting priority combinations were requirements that were mandatory, used all the time, and very well supported since these were the top requirements. Also requirements that were real differentiators were interesting considering the future of application development in the Case Company. In the developer point of view the requirements that were missing but were said to be either mandatory or used all the time if they would implemented were interesting.

5.1.5. Environmental factors

Coalitions creating consistency

In each business unit of the Case Company the members have daily interactions with each other which have a natural effect on creating coalitions. It means that a certain part of a multi-site organization has in time developed its own organizational culture and ways of operating. This creates consistency among the viewpoints presented by the workshop participants. Therefore it is more likely that the participants present in the workshop will also be able to take into consideration their colleagues needs. Which means that the needs of local working context will also be most likely considered in the requirements as well. Also this way the requirements will be more representative since more viewpoints are being considered at once.

Unfortunately creation of coalitions also creates problems since in each business unit there are separate coalitions with different goals and viewpoints which leads to difficulties in forming a synthesis among the results from all workshops held for different business units. Although this was the only possible approach since if people from all the business units were included in the workshop session at once most of the time would have been spend on arguing and nothing could be actually decided. Also it would be impractical since there would be too many people present at once to run a workshop efficiently.

Social aspects

In requirements gathering in addition to technical aspects the social aspects are important, or even more important, than the technical aspects. Even though the requirements gathering team tries to be as neutral and as technical as possible it is usually operating upon assumptions rather than objective facts. Requirements gatherers have the view that knowledge is objective, unchanging and precise. This is definitely not true since the political considerations are playing a key role usually overruling the technical ones.

Workshop participants are motivated to invest in effort for finding a consensus above all things. Participants are also more aware of the distribution of power and its key role in controlling and legitimating organizational views. [12]

In addition to that in this project, at least according to workshop participants, different business units were willing to compromise in order to improve their current operations in process automation application development. This willingness to compromise is mostly due to the factors presented in the section 1.1 which explains the motivation for this project done for the Case Company.

5.2.Summary of the results

The goal of this work was to find a process that could solve or at least take into consideration the challenges posed in requirements gathering in a multi-site organization. The challenges that were only partially addressed a short explanation is given what could have been done differently. For those challenges that weren’t addressed at all an alternative way of doing things is offered. The analysis in the results is based on a project done for the Case Company. Here only the key aspects of the workshop methodology in previous section are presented.

Workshops

One day workshops give enough time for everyone to express their point of views increasing communication between the requirements gathering team and the participants. In the workshops 2-way communication offers enough time to express different viewpoints which helps to manage conflicts. Furthermore because of immediate feedback the time used to discuss about the form and the content of requirements decreases. Also this way the requirements gatherers have enough time to ask clarifying questions to fully understand the requirements. The workshops also force the participants to crystalize their ideas resulting in compact and clear requirements. On a bigger scale the workshops give the participants a feeling that a change is coming and therefore they have time to get used to it lowering the possible change resistance.

It was essential to have separate workshops for each business unit. Creation of coalitions in each business units unifies the viewpoints inside a unit but on the other hand further differentiates the opinions between business units resulting in conflicts. On the other hand having varied range of business unit was essential to guarantee the representativeness and also the scope of requirements.

It was necessary to have moderator control in the workshops. Having one or several moderators controlling the flow of the workshops guarantees the efficiency of the workshop. Moderators made sure that only relevant things were discussed. The moderators made sure that the requirements were validated.

Understanding the social aspects in requirements gathering is essential for the success of the whole project. Having people of power present alleviates trust since they are not just thinking the best of one business unit. Also these people have the power to make actual changes. Therefore the participants are more willing to share all possible information about the requirements. By including people with power there is less need for documentation. Also these people have inter-business unit understanding that is critical in understanding the domain specific language and social aspects behind the requirements.

If the end-users were included it would have offered valuable information about the local working context. Probably the best way to do this would have been to interview different people in a customer site while they were doing their work using the Case Company’s products.

In order to visualize ideas in a quick and easy way post-it notes and flip charts were used. Also the use of mind maps in requirements gathering help to visualize what is left to be discussed and also to distribute the understanding or requirements in a simple graphical form.

Requirements gathering workshops with ULRC methodology

The key point in ULRC methodology is that the participants are making the requirements by themselves while being instructed by moderators. This way the participants’ opinion was heard and the local working context is taken into consideration. This approach minimizes the researchers’ effect on the study and to further minimize it the role of apprentice was also used. This helped to avoid making wrong interpretation about the requirements. After participants’ articulated their needs the form of requirements was negotiated with everyone.

This resulted in common understanding as well as to more valid and clear requirements. On a bigger scale when the participants are creating the requirements they have a feeling that they can truly affect the outcome. This way they will react a lot more positively to a possible change.

Having a simple and clear prioritization scheme showed how different business unit of the Case Company valued individual requirements. Since the priorities highlighted the differences in viewpoints of different business units it made analysis and management of requirements a lot easier.

Predefined categories in the ULCR –methodology were used to categorize the requirements when they were gathered making the management of the requirements easier. The categories made easier for the experienced participants to manage their vast knowledge resulting in larger amount of diverse high quality requirements. Because all the requirements where in categories the requirements from same category they were comparable between workshops. This made it a lot easier to look for repeating patterns, duplicate requirements, conflicting requirements, and to determine the importance of individual requirements for each business unit. In the workshop the categories made sure that no time was wasted in figuring out was already said and what was left to be discussed. The simple visual appearance of mind maps also helped other people of the business unit who weren’t part of the workshop to understand the requirements more easily.

Distributing the mind maps made sure that the gathered requirements were relevant for the whole business unit and not only for the workshop participants. It also helped to gather more diverse requirements by having more people participating in the project. Distributing mind maps also increased the awareness of the requirements gathering project. This prepares people to expect changes in the future and therefore making them more adaptive to change.

The mind maps had a “hints” subcategory. It is an already filled framework of requirements only seen by the moderators of the workshops. It helps moderators to keep track that all the relevant things are said in the workshops. Moderators can start new topics based on these “hints” making the participants more aware of different aspects of the requirements.

Brainstorming workshops with Linja methodology

The fundamental idea behind this brainstorming methodology is to understand the user and the use context. Ideation around real contexts helps participants to recall real life examples. It is easy for participants to tell about incidents that happened in their work. Because of real life examples there is a lot of detailed information available about the requirements. This helps the developers to fully understand the idea and also it helps to understand the local working context and the differences in business making the requirements more complete and clear.

The brainstorming methodology helped to distribute the focus of brainstorming throughout the whole innovation space. The context of use acts as an umbrella concept for benefits and enables and therefore by choosing very different contexts the ideation is differentiated quite broadly.

In the brainstorming workshops the participants were asked to focus on the things that they see as most relevant for their work context. This was accomplished by voting in the workshops. Also the goal in the workshops was not the make the best possible trade-offs. The goal was rather to look for new innovative ideas that could open a whole new market according the idea of ‘blue ocean’ strategy. These approaches helped to pin down requirements while they were gathered.

Analysis: looking for repeating patters in the data

In the synthesis phase the true needs connecting different business units are found based on constant comparison analysis. This helped to reveal underlying needs that the participants failed to articulate or requirements gatherers failed to understand. Statistical analysis on the other hand is free from biases presented in subsection 3.1.3. Also it gives an opportunity to compare statistical and constant comparison analysis results to see how credible the results are.

Environmental factors

In each business unit of the Case Company the members have daily interactions with each other which have a natural effect on creating coalitions. Because of this the workshop participants are more likely to take into consideration their colleagues needs as well.

Therefore the requirements will more accurately present the view of the whole business unit of the Case Company.

Social aspects should be taken into consideration in requirements gathering. Even though the requirements gatherers try to be as neutral and as technical as possible they are usually operating upon assumptions rather than objective facts. Therefore political considerations are playing a key role usually overruling the technical ones. In order to understand the requirements the social aspects must be understood as well.

5.3.Reliability of the study

The requirements gathering project conducted for the Case Company, on this work is based on, included many challenges which lead to taking short cuts in decisions leading to less academic approach of handling the requirements gathering.

It is a big challenge to gather requirements from each individual business units only through workshops. Because of busy schedules of people and the project there were only a few key persons in the workshop whose job was to represent the whole business unit. This becomes difficult since a few people can’t be expected to verbalize all the necessary requirements for their business unit leaving some key points out of the discussion. In addition because of the schedule some key users weren’t able to make it to the workshop meetings. Of course gathering all the requirements into a mind map and distributing the mind map to other members of the business unit helped to guarantee the representativeness. On the other hand there wasn’t a guarantee that the workshop participants actively required other members of the business unit to review the mind map in order to see if they agree on its content.

In order to focus this study I had to put the emphasis of the study on software developers in different business units of the Case Company who are creating the applications for end-users in different industries. Therefore the role of end-users in this study was completely ignored. This had an effect especially on the requirements gathered concerning end-users needs.

Because of busy schedules of people and tight schedule of the project the methodology had to be simplified since there wouldn’t have been time to teach how to use for example a complex and descriptive model to guarantee the unambiguousness and comparability of requirements. Because of this the full details of requirements weren’t documented it made finding similarities among the requirements from different workshops possibly easier leading to unsure conclusions about the results in some cases.

Another thing that affects the reliability of this study is that I had limited authority and knowledge who would be the key users for these workshops. Therefore the selection was based on the knowledge of my colleagues. Because of this I can’t be fully assured that the people where the best possible representatives of each business unit.

It was also challenging to define adequate level of understanding in order to understand the deeper meaning of the requirements gathered and how do they interact with each other in the work of application engineers. The fact is that for me it is not possible to have the same deep understanding about the Case Company’s user interface development tools and application development as my colleagues had. Therefore instead of complete understanding I had to operate with an adequate level of abstraction about the domain and it must have had effect on the results also.

Fortunately in each workshop there was at least one person from the Case Company present as a part of the requirements gathering team. Their domain expertise significantly alleviated the uncertainties about the way how requirements were understood by other members of the team with less domain expertise. Furthermore after revealing the results of the requirements gathering project to representatives of several business units they mostly agreed on the results. They said the results indicated something they already suspected but they weren’t sure of. Also they agreed on more surprising results based on repeating patterns after seeing graphically how the results were formed using an Excel sheet with color coding, with information traceable to its source, which was presented in subsection 4.5.1.

6. Conclusions

6.1.Conclusions

The requirements gathering process presented in this work answers several challenges presented in work. The strengths of this process are shown in requirements gathering projects in multi-site organizations where the requirements are highly diverse, the amount of requirements is large, and it is essential to find the most important requirements quickly.

In this section the way how well the work answers the research questions presented in chapter 2 is evaluated. The main research question for this work was: “How to gather requirements from very different business units in a way that they are still comparable with each other?” This question is answered by using pre-defined categories acting as a framework for gathered requirements as well as a prioritization scheme used in prioritizing the requirements while they were gathered.

The pre-defined categories (subsection 4.3.2. and Appendix I) used in requirements gathering workshops acted as a basis for managing the requirements. When the requirements started to pile up the vast quantity and diversity of requirements was controlled by pre-defined categories in user-led construction of requirements methodology. Each individual requirement from different workshops was organized based on the same framework making the management of high quantity of requirements a lot easier. Also since all the requirements were filtered to fit this framework also made them more consistent and therefore made them easier to understand for requirements gatherers and participants as well. Because of the predefined categories and consistency the requirements were directly comparable inside a category among different workshops. This made it easy to look for repeating patterns, duplicate requirements, conflicting requirements, and to determine the importance of individual requirements for each business unit.

The prioritization scheme in requirements gathering workshops presented in section 4.2 helped to define which requirements were most important for each business unit. Also the clear priorities made it a lot easier to manage a high number of requirements since they could be filtered in an excel sheet based on their priority. These priorities helped to define how differently business units valued certain requirements. This was very important information in analyzing the similarities between process automation application development in different business units.

Another challenge for the process of requirements gathering was the time restrictions set by the project done for the Case Company. To make the process more efficient requirements were categorized and prioritized while they were gathered. Furthermore “hints” subcategory was used to start new topics in the workshops around relevant domains to save time. Also the workshop approach offered immediate feedback minimizing the time required for negotiation. Furthermore arranging separate workshops for each business unit decreased the time for possible arguments about the requirements. And finally moderator control ensured efficiency of the workshops making sure the discussion was always about relevant topics.

The main research questions also opens a discussion for more general question “What kind of process for gathering requirements should be designed to meet the challenges in requirements gathering in multi-site organization in general?” The process described in chapter 4 answers several challenged posed in chapter 3. How the challenges are answered is further explained in chapter 5. The key aspect of this process is coping with high diversity of requirements using a mind map framework and a prioritizing scheme as explained before. This keeps the requirements comparable between workshops. Also the contexts specific approach in brainstorming workshops and participants creating their own requirements in requirements gathering workshops ensure that the requirements gathered are correct and relevant. In addition moderator control and having people with authority and domain expertise present ensure that the discussion is around the most important topics saving considerable amount of time. Also this way the participants know they are talking to people who can actually change things having the best interest of the whole division in mind therefore revealing all the relevant information.

6.2.Future steps and recommendations

At this point all the high-level requirements are gathered. Unfortunately it is quite limited what can be said about the actual results of analyzing the requirements due to confidentiality issues. What can be said that the analysis of these requirements lead to finding several repeating patters in the needs of business units. These repeating patterns acted as basis for a concept to unify the operations in the business units of the Case Company. The concept was presented to each business unit of the Case Company who were involved in the requirements gathering project. The next steps after this is to further develop the concept to make it detailed enough to bring the discussion on practical level considering about issues related to implementing the concept.

In order to get clarity to the issues related to implementing the concept a further study must be conducted. This study will be focused on the feasibility of the concept. For example finding out, what is the payback time of the concept or what kind of support different business units of the Case Company need to implement the concept in their operations. This feasibility study will be focused on the technical aspects of implementing the concept, for example choosing the technologies which clearly have a future considering the long lifecycle of the Case Company's products. Findings of the feasibility study will finally reveal how feasible it is actually to implement this concept in practice.

As mentioned throughout this work the scope of the requirements gathering project done for the Case Company was very wide which created additional challenges for managing the diversity of the requirements for the proposed process. Even though the process described for this work was designed to be suitable to handle this diversity it must be said that the scope of the requirements gathering project was slightly too wide. To get the most of this process described in this work the scope of the project should be narrower. The key advantage of this process is to reveal the most important requirements when the amount and diversity of requirements is moderately large.

References

1. Young R.R. The Requirements Engineering Handbook. 1st ed. Norwood, Massachusetts, USA: Artech House, Inch, 2004. 264 p. ISBN 1-58053-266-7.
2. ABB homepage, Products & Services, Collaborative Production Management. Available at: http://www.abb.com/product/us/9AAC133385.aspx. [Cited 9.6.2011]
3. ABB Oy. Industrial IT cpmPlus System Version 1.0 Visualization. 2010. 582 p. Document number: 9ARD000007R1001.
4. Sinclair, J.T & Merkow, M. Thin Clients Clearly Explained. 1st ed. San Francisco, California, USA: Morgan Kaufmann Publishers Inc, 1999. 350 p. ISBN 0-12645-535-X.
5. P. Jalote. An Integrated Approach to Software Engineering. 2nd ed. Springer-Verlag New York, Inc. 502 p. 1997. ISBN 0-387-94899-6.
6. Damian, D.E & Zowghi, D. The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. IEEE Joint International Conference. Essen, Germany. 13-13.12.2002. p. 319-328. [Cited 9.6.2011]. ISSN 1090-705X. (DOI: 10.1109/ICRE.2002.1048545)
7. Sabahat, N & Iqbal, F & Azam, F & Javed M.Y. An Iterative Approach for Global Requirements Elicitation: A Case Study Analysis. 2010 International Conference on Electronics and Information Engineering (ICEIE 2010). Vol 1. 2010. 6 p. [Cited 29.9.2011]. DOI: 10.1109/ICEIE.2010.5559859.
8. Raffo, D & Setanit, S. Simulation Model for Globarl Software Development Project. Portland State University, Portland, Oregon, USA. 7 p. Available at http://www.sba.pdx.edu/faculty/davidr/draccess/WEB/publications/JOURNAL/ProSim’05-GDS.pdf. [Cited 29.9.2011].
9. ABB Oy. Industrial IT User Interface Release 4.2 User Interface User’s Reference Manual. 2009. 145 p. Document number: 3aFA000519_I_EN.
10. ABB homepage, The ABB Group, Facts and figures, Our Businesses. Available at: http://www.abb.com/cawp/abbzh252/a92797a76354298bc1256aea00487bdb.aspx. [Cited 9.6.2011].
11. ABB homepage, Products and services, ABB Product Guide. Available at: http://www.abb.com/product/us/9AAC910002.aspx?v=9AAC910002&e=us&m=9F2. [Cited 9.6.2011]
12. Flynn D.J & Jazi M.D. Constructing user requirements: a social process for social context. Info Systems Journal. Vol. 8:1 1998. 53-83 p. [Cited 9.6.2011]. DOI: 10.1046/j.1365-

2575.1998.00004.x.

13. Goldsmith, R.F. Discovering Real Business Requirements for Software Project Success. Massachusetts, USA: Artech House, INC, 2004. 207 p. ISBN 1-58053-770-7.
14. Pitts, M.G & Browne G.J. Improved Requirements Elicitation: An Empirical Investigation of Procedural Prompts. Information Systems Journal. Vol. 17:1. 2007. P. 89-110. [Cited 9.6.2011]. DOI: 10.1111/j.1356-2575.2006.00240.x
15. Mayon-White, B & Mabey, C. Managing change 2nd Edition. Chapter 6. 1993. Wiltshire, UK: The Cromwell Press Ltd. 2004. P. 218. ISBN 1-85396-226-0.
16. Young, R.R. Twelve Requirements Basics for Project Success. CrossTalk:The Journal of Defense Software Engineering. Vol 12:1. 2006. P. 8. Available at: http://www.crosstalkonline.org/storage/issue-archives/2006/200612/200612-Young.pdf [Cited 9.6.2011].
17. ABB homepage, Products & services, ABB Product Guide, cpmPlus History. Available at: http://www.abb.com/product/db0003db004001/bb5f32b48947b56dc125777e003d384d.aspx?productLanguage=fi&country=FI. [Cited 9.6.2011].
18. ABB homepage, industries. Available at: http://www.abb.com/industries/ [cited 4.7.2011]
19. Holtzblatt K & Beyer H.R. Requirements gathering: the human factor. Communications of the ACM. Vol 38:5. 1995. P. 3. DOI: 10.1145/203356.203361.
20. Linja Oy. ABB UI Platform, Design driver workshop [Power Point slides]. [Cited 9.6.2011].
21. Mead. N.R. Requirements prioritization introduction. Carnegie Mellon University Pennsylvania, Pittsburg, USA. 2005 Available at: https://buildsecurityin.us-cert.gov/bsi/articles/best-practices/requirements/545-BSI.html. [Cited 9.6.2011].
22. Linja homepage http://www.linjadesign.fi/web/guest. [Cited 9.6.2011].
23. Kim W. C & Mauborgne R. Blue Ocean Strategy: How to Create Uncontested Market Space and Make Competition Irrelevant. Massachusetts, USA: Harvard Business School Press, 2005. 239 P. ISBN 1-59139-619-0.
24. Beyer, H.R & Holtzblatt, K. Apprenticing with the customer. Communications of the ACM. Vol 38:5. P 8. DOI: 10.1145/20356.203356.
25. Seaman, C.B. Qualitative Methods in Empirical Studies of Software Engineering. IEEE Transaction of software engineering. Vol 25:4. P 557-572. DOI 10.1109/32.799955.

Appendix I

Below there is an explanation of each of the predefined categories in the generic mind map category.

Use context

Defines different **use context** for applications used by customers. In practice it contains the most common tasks or job descriptions of end users who use the Case Company’s applications as a part of their work.

Contains also **characteristics of the environment** where applications are used for example in oil platforms the operations are divided to on-shore and off-shore operations

Can contain items describing the **nature of work** in each context for example time pressure: certain things must be done immediately at a certain time

Navigation to information

Describes the ways how application engineers **can access the information** relevant for creating applications through the provided interface

Examples of ways for accessing relevant information: context sensitive links, navigation tree or browse/search/filter data

Displays

Describes all the required ways of **presenting data inputted in the system** and also the **graphical presentation of the user interface** itself

Examples of data visualization: trends, lists, portal displays, reports

Integration with other systems

Describes the **most common system integrated with the Case Company’s systems**

Explains all the **relevant operations connected to integration with other systems** such as data integration and integration in the graphical user interface perspective. As an example there might be a need to drag and drop data components from system to another.

Customizing online systems

Describes the **customization needs of end users** for their applications. For example the user might want to change the layout of the applications.

Security

Security requirements for the Case Company's process automation applications. It can contain security related technologies such as Kerberos or security strategies such as having hierarchical access rights.

Performance

Rough **estimation of performance** requirements for the Case Company's applications from the developers and end-users perspective

Project engineering

Consists of all the operations and properties required from the interface that are related to **configuration and installations of the Case Company's applications** to customer sites for example configuration wizard to help in configuration.

Product development

Consists of all the operations and properties required from the interface that are related to **creating a graphical user interface** for an application requested by the customer. Such operations are related to constructing a user interface from graphical components, coding, help documentation etc.

Software technology

Describes the **software technologies that must be supported** in order to support the software that is currently used and also what might be the software technologies to be used in the future.

Cool stuff in the future

Describes very freely what will be the **trends and wow-factors** in process automation applications and application development in the future. This is a chance for the participants to freely express ideas that the discussion during the workshop aided to generate.

Prioritization

Describes the **prioritizing framework** used in requirements gathering frameworks. It is thoroughly explained in section 4.2.

Appendix II

Here is an example of a modified version of constant comparison methodology used in this work. The numbers on the top of the table indicate individual workshops and the letters indicate individual items.

Phase 1: Unsorted data

1	2	3	4	5
A	A	C	B	C
A	D	A	C	B
C	F	C	A	B
A	H	E	B	J
F	R	H	W	E

Phase 2: Color items having similarities with same color

1	2	3	4	5
A	A	C	B	C
A	D	A	C	A
C	F	C	A	B
A	H	E	B	J
F	R	B	W	E

🟦
🟥
🟨
🟩
🟢

Phase 3: Insert frequency numbers

1	2	3	4	5
A	A	C	B	C
A	D	A	C	A
C	F	C	A	B
A	H	E	B	J
F	R	B	W	E

🟦	4
🟥	5
🟨	2
🟩	2
🟢	3

Phase 4: Arrange the repeating patterns based on frequencies

1	2	3	4	5
A	A	C	B	C
A	D	A	C	A
C	F	C	A	B

🟥	5
🟦	4
🟢	3
🟨	2

A	H	E	B	J
F	R	B	W	E

🟩	2

Phase 5: Name the repeating patterns (themes are formed)

1	2	3	4	5
A	A	C	B	C
A	D	A	C	A
C	F	C	A	B
A	H	E	B	J
F	R	B	W	E

🟦	4	The C's
🟥	5	The A's
🟧	2	The F's
🟩	2	The E's
🟢	3	The B's

Phase 6: Arrange the items

1	2	3	4	5
A	A	A	A	A
A	F	C	C	C
A	D	C	B	B
C	H	B	B	E
F	R	E	W	J

🟥	5	The A's
🟦	4	The C's
🟢	3	The B's
🟧	2	The F's
🟩	2	The E's