



Digital Business Ecosystem

Contract n° 507953

WP31: Regional Catalyst Action

Del 31.4: User and provider SME requirements description



Project funded by the European Community under the "Information Society Technology" Programme

Contract Number: 507953
Project Acronym: DBE
Title: Digital Business Ecosystem

Deliverable N°: 31.4
Due date: 31/5/2005
Delivery Date: 7/7/2005

Partners owning: Technology Centre Hermia Ltd
Partners contributed: Technology Centre Hermia Ltd

VERSIONING		
VERSION	DATE	AUTHOR, ORGANISATION
0.1: DRAFT	31.5.2005	PETRI RÄSÄNEN & JAANA HELIÖ; TECHNOLOGY CENTRE HERMIA LTD
1.0: STABLE	4.7.2005	PETRI RÄSÄNEN & JAANA HELIÖ; TECHNOLOGY CENTRE HERMIA LTD
1.1: STABLE	5.8.2005	PETRI RÄSÄNEN & JAANA HELIÖ; TECHNOLOGY CENTRE HERMIA LTD

Quality check
1st Internal Reviewer: Tim Romberg, FZI
2nd Internal Reviewer:

Table of Contents

1	INTRODUCTION.....	4
1.1	THE FOCUS AND AIM OF THE DELIVERABLE	4
1.2	STRUCTURE OF THE DELIVERABLE	4
2	REQUIREMENTS FOR THE KNOWLEDGE MANAGEMENT IN DBE-SME SOFTWARE DEVELOPER COLLABORATION	6
2.1	INTRODUCTION	6
2.2	NEEDS FOR ONTOLOGIES	7
2.3	NEEDS FOR MODELING IN REQUIREMENTS ENGINEERING	8
2.4	EXPECTATIONS WHEN WORKING WITH OPEN SOURCE SOFTWARE	10
2.5	REQUIREMENTS ENGINEERING RECOMMENDATIONS	12
2.6	CONCLUSION.....	14
3	APPLICATION NEEDS OF USER SMES	17
4	DRIVER SME REQUIREMENTS FOR DBE PLATFORM.....	20
5	CONCLUSIONS	22
	APPENDIX I	25
	APPENDIX II.....	28
	REFERENCES.....	31

1 Introduction

1.1 The focus and aim of the deliverable

This deliverable summarises the results of work that has been carried out in the first 18 project months in WP 31 in the area of SME requirements analysis. One of the subtasks of the task B29 - Functional Analysis and Specification - has been to provide a SME-centric view to the more theoretical analysis of requirements. This has been done by bringing a small set of software-intensive SMEs and ICT experts from the Tampere Region into the DBE project and studying them as a source of requirements for the DBE system.

The results in hand have been extracted from research work and training sessions that have been conducted in 18 months period starting from November 2003 and ending by May 2005. During this period significant progress has been made in the DBE project in all the research domains. This progress is also reflected in the results presented here. During the earliest studies, most areas of DBE technologies, DBE business vision and ways of operation were not yet crystallised which becomes clearly visible also in the research findings. These results also show how the interaction with SMEs in the regions and the level of knowledge within the DBE project itself has evolved from fuzzy presentations of innovative ideas and visions towards detailed understanding, analysis and demonstration of useful technology.

This deliverable takes a general, but region specific socio-economic and managerial approach to SME needs and requirements and thus complements the more detailed and technically oriented work that has been done in the field of requirements definition (e.g. use-case analysis). The purpose of the deliverable is to provide knowledge about the business-related and non-functional SME needs and requirements and thus contribute the overall understanding of SME requirements for DBE. This knowledge is meant to be utilised in the planning and execution of the present and forthcoming project activities, especially DBE bootstrapping, SME engagement and sustainability planning.

1.2 Structure of the deliverable

In the following chapters the results of the studies on SME needs requirements are presented in three blocks. The first, most extensive chapter, summarises the result of a research work, that has been published as a Master's Thesis at the Department of Industrial Engineering and Management at Tampere University of Technology in November 2004. The aim of this study was to investigate what information flow and information management practices are needed when the small and medium sized software enterprises are developing software services for the DBE. The study is the earliest of the studies presented here and it was completed by September 2004.

The second chapter describes the software service needs of SMEs in the Tampere region. The results are extracted from the database of a two-year project within the eTampere programme that has collaborated with the DBE project and has provided us

with insights on the grass-root level needs of user SMEs. The empirical data has been collected in 2004-2005.

In the third chapter, the technological requirements of potential driver SMEs are presented. These requirements have been extracted from a SWOT analysis of the main components of the DBE platform. The analysis was made by five driver candidate SMEs at the Tampere region in a training workshop in February 2005.

2 Requirements for the knowledge management in DBE-SME software developer collaboration

2.1 Introduction

As part of the DBE research project, a study on the SME needs for information flow and information management in DBE was conducted during the first project year (see Heliö 2004) The study was performed by M.Sc. Jaana Heliö and supervised by professor Tuija Kuusisto at Tampere University of Technology.

In the following chapters only the main findings of the study are summarised. The study covered a wide range of questions – a fact that reflects the multidisciplinary, visionary and emergent nature of the DBE project and the resulting challenges in internal and external communication during the first phases of the project.

The main objective of the study was to study **what information flow and information management practices are needed when small and medium sized software enterprises are developing software services for the DBE**. The research was done from the software developer SMEs point of view in order to let them provide the DBE-project with information that will ease their involvement in the DBE. Thus, the tactical aim of the research was not only to collect requirements but also to support the formation of the engagement strategy and the execution of engagement activities.

The fundamental concept of the study is requirements engineering. For the SMEs under study, requirements engineering and management are among the key tasks in their SW engineering process. DBE was expected to introduce new approaches, methods and tools for requirement engineering and thus transform the core processes of the SW SMEs. In this context the study investigates the information management needs of SMEs and tries to analyse the match between DBE core concepts (e.f use of modelling, ontologies and open source) and the current practices and needs of SMEs. On the other hand, the concept of requirements engineering is also used in order to study SME expectation for the DBE as an Open Source project. The beliefs of SMEs toward requirements engineering in Open Source project were examined and their specific expectations towards DBE project were elaborated.

In the theoretical part of the study, a conceptual approach was applied. In this part, a general understanding about requirements engineering collaboration between proprietary software companies and open source projects was acquired. In the empirical part, action-oriented research approach was applied. The case method was used for examining seven advanced software SMEs in the Tampere Region. All of them were proprietary software producers, which gave an opportunity to examine the state of requirements engineering practices in contemporary proprietary software business. Most of the participating companies were also actively working with the open source community, which made it possible to study the point of view of open source software collaboration as well. The requirements engineering collaboration between proprietary companies and open source projects was examined by interviewing selected software specialists and business managers of the companies.

The case studies were conducted in spring-summer 2004 in seven leading software engineering SMEs in the Tampere region. The profiles of the companies participating in the study and the interview guide used in the case studies are presented in the appendix.

In order to answer the main research question, four sub-questions were formulated to guide the study in more detail. The four sub-questions were the following:

1. Why is ontology needed by the SMEs and how is it used?
2. How do current software requirements engineering principles in practice support managing the information needed by the SMEs?
3. What are the ways of working with open source software projects for an SME?
4. What are the requirements engineering recommendations for the DBE?

These questions and the identified answers for them are discussed in the following sub-chapters. The data sources and detailed reasoning leading to the summarised results presented below can be found in the full research report that has been made available for the commission as an addendum to this deliverable.

2.2 Needs for ontologies

Based on existing literature, four general benefits for the use of ontologies in software development were anticipated. First, ontologies offer an opportunity to reduce diversity in commonly dealt information. Secondly, it enables companies to improve operational productivity by reducing errors and noise in information and thus leading to time and money savings. Thirdly, ontologies facilitates better co-operation both within and across organizational boundaries. Fourthly, it offers the basis for optimizing organizational change by the right set of tags.

When comparing these anticipated benefits with the company views on ontologies, only a few links can be pointed out. Most companies under study saw defining ontology as useful. However, it was difficult to get direct answers, why in fact ontology could be useful in their specific business cases and how it could be used, even if the company did consider ontology useful.

Based on indirect observations about the challenges the companies are facing, however, it can be noted that good reasons for defining ontology in the cases of the studied SMEs could be lessening diversity in information and reduction of error and noise. On the other hand, not a single company saw ontology as a way of managing organizational change by suitable tagging or as a way of facilitating better co-operation.

Generally, ontologies and the opportunities they yield for software engineering do not seem very familiar in software producing SMEs in the Tampere Region. The arguments about needs to use ontologies in SMEs to reduce errors and noise leading to saving time and money, and facilitating better co-operation within and across organizational boundaries, did not receive much support from the SMEs in this study.

2.3 Needs for modeling in requirements engineering

The second sub-objective was to investigate how do current software requirements engineering principles in practice support managing the information needed by the SMEs. In the first phase of the execution of the DBE project in the Tampere Region, DBE was clearly expected to introduce some new methods for the management of customer requirement in SW production. In this context the aim of this subtask was to provide a set of baseline information about the current status of requirement engineering and the needs of SMEs in this area.

Information management can be viewed as the management of network of processes that acquire, create, organize, distribute and use information as a continuous cycle. Requirements engineering, then again, includes elicitation, analysis, specification, verification and management of software requirements. The goal of information management is to transfigure information into learning, insight, and commitment to action, whereas requirements engineering aims to transfigure information first into learning and insight, then into action, the implementation of the software. Consequently, requirements engineering is a way of managing information.

According to the model of Choo (1998) information needs are discovered from problems, questions, and ambiguities through different situations and experiences in an organization. The information has to be rendered meaningful to certain individuals in certain situations; thus, the meaning of information is not the only thing concerned. Requirements engineering begins with analyzing the problem. In the first step of requirements workflow, the problem is analyzed and the aim is to gain an agreement on the statement of the problem to be solved. In addition, the first step of requirements workflow includes identification of stakeholders, persons who have a stake in the project outcome, for example different sets of users.

The stakeholder needs are elicited and captured in the second step of requirements workflow. The needs are captured as requirements with the help of use cases that are modeled with UML. The use cases are a method of describing what each set of users wants the system to do for them. Each set of users also produces several different use cases for different situations of using the system. Thus, the system requirements information described by use cases is meaningful for certain individuals in certain situations.

When comparing the model with the existing practices in SMEs, the requirements workflow is quite similar, yet more simple in some cases. The stakeholders are simply the customer companies negotiating with the SMEs. Whether a more thorough identification of stakeholders is performed within organizations of the customers or not, did not come up in the company answers. The capturing of the requirements is performed with the help of UML, sometimes yes, but more often in natural language. Traditional requirements specification document seemed to have sustained its popularity as a starting point of capturing requirements. Business modeling was not mentioned as a separate phase of requirements capturing. However, that does not mean it would not be performed, it could just be performed as an integrated part of requirements capturing.

Based on the case studies, it seems that the size of the SME and the size of the customer are important factors influencing the extent in which the requirements are modeled and documented. The bigger the company, the more predefined requirements

process it possessed and the more developed requirements modeling it used. This does not mean, however, that smaller SMEs would perform their work worse than bigger ones. Smaller companies often carry out smaller projects, which do not need to be as structured as bigger projects. In addition, both big and small companies tailored their processes and methodologies according to needs of bigger customers, when required.

In requirements engineering several sources of information are typically used. The variety of information sources is managed by selecting typically one person per a group of stakeholders that have similar job descriptions, to give the group requirements as use cases or in natural language. In many cases existing software can provide useful set of requirements.

In the SMEs under study humans are the most important source of information. Requirements are elicited, captured, and analyzed often in face-to-face meetings in collaboration between the company and the customer personnel. Occasionally, old software is not needed; sometimes it seen even as an obstacle for innovative creation of new solutions. On the other hand, new ideas can sometimes be caused by another software product, for instance by a competing software product. Anyway, in requirements engineering information is always acquired by humans from humans, with or without using current or competing software or other information technology in assistance.

Requirements information organization is traditionally performed according to software functions, for example requirements for a user interface, requirements for a database table, and requirements for a background program. However, often a single requirement influences many design areas, for example functionality in a customer information user interface may affect the user interface, a background program and a customer information database. In that case, separate requirements should be created and linked together via requirements management tool or equivalent. Then, the three different requirements would not be forgotten and in case any of them changed, the possible need to change the other two would be noticed as well.

The requirements information storing and presentation can be carried out in multiple ways depending on the company practice. Traditionally requirements have been gathered in a requirements specification, which is a document written in natural language. This was the case with the interviewed companies as well. The document may also contain preliminary pictures of the user interface and preliminary models of the database and the software architecture. Use cases, both in natural language and in UML, have been gaining popularity recently as a way of describing and storing the requirements. Most companies in this study used UML too; yet, it was not the obvious choice for most of them. Only one company, the biggest one, stated that modeling the requirements with UML is more a rule than an exception.

Use cases are often an addition to the requirements specification in order to verify the working of certain key features of the software system. Sometimes, requirements are stored individually in a database with a software product meant for the purpose. This was the case in one company. Every requirement is given for instance a title, description, state, person responsible, timeline and most importantly, a link to other requirements that may be due to change if this requirement changes. With the help of these software products, the lifecycle and change management of requirements can be made easier. Recently Internet-based documentation has gained popularity but most of

the interviewed companies had their documents in local storages, although plans for taking Internet-based documentation into use had been made in few companies.

Altogether, compared with the recent models of software engineering, the requirements engineering processes and methodologies supporting the information management of the SMEs in this study seemed to be quite informal, tailored and customer driven. As a main rule, requirements were collaboratively elicited and analyzed with customers and documented as a requirements specification in natural language.

Requirements were not always modeled with UML. Mains reasons for this were that the models change quickly and UML modeling is more time consuming than using natural language. Meta-modeling the software system with MDA and MOF was not familiar to the SMEs. If meta-modeling had been familiar to the companies, it would probably still not be used regardless of its advantages, as it is even more time consuming than using UML is.

It turned out that bigger SMEs obeyed more carefully requirements engineering processes and methodologies. However, even smaller SMEs tailored their requirements processes and methodologies when needed according to the process of the customer, as bigger customers would expect them to obey their own, more developed processes.

2.4 Expectations when working with Open Source Software

The third sub-objective was to find out what are the preferred ways of working with open source software projects for an SME. DBE software environment will be published and further developed as open source software. SMEs are expected to provide services on top of this infrastructure and build their own business models in DBE. On the other hand, the driver SMEs are also expected to contribute the first releases of DBE environment. Thus for SMEs, when they consider participating DBE, it is important that the expectations about the operational practices, maturity and sustainability of DBE as an open source project are sufficiently met.

It was found that SMEs could not be treated as one group, as their experiences, expectations, and attitudes regarding open source software, are not uniform. Therefore, the seven interviewed SMEs were classified into three groups regarding their collaboration with open source community: strategic, active, and observing collaborators.

1. Strategic collaboration with the open source community indicates that the company business idea is somehow based on open source software. For instance, a company can offer different kinds of services to open source software product in addition of selling it. Strategic collaboration also indicates that the company is actively involved in the community and understands the basic idea of the open source community: if you take something from the community, it would be advisable to give something back. Additionally strategic collaborators possess enthusiastic attitude towards open source software. Two companies out of seven fell into this category.

2. Active collaboration with the open source community means that the company uses open source software as part of the company product or in the development of it, but the open source software use is not especially strategic to the company. The business idea of the company is not based on open source software. Active collaborators might give their requirements to the open source community, but would probably not participate in implementing open source software. In general, active collaborators have a positive attitude towards open source software. Three companies out of seven fell into this category.
3. Observing collaboration with the open source community indicates that the company is using open source software, typically software development tools made by the community, thus, software is taken from the community, but rarely anything is given back. The attitude towards open source community might even be a bit negative in observing collaborator's case. Two companies out of seven fell into this category.

The division of companies into the above mentioned group reflects the amount of experience in open source software. The least experienced companies came from observantly collaborating companies and the most experienced ones came from strategically collaborating companies. The division also reflects the preferences to use and collaborate open source software. The more experienced the company is with the open source community, the more positive attitude towards the community it had. Which one came first, the positive attitude towards open source software or the experiences of collaboration with the open source community was not the subject of this study. However, it can be noted that if the positive attitude existed first, it had not changed with open source community collaboration.

The above mentioned classification of open source collaboration can be compared to a division of open source software usage model: internal use, product use, shaping market situation with open source software, and open source software as a business. When comparing this division with the division presented in this research, some similarities and some differences can be found. One of the companies, categorized as an observing collaborator in this study, is using open source software internally; the other observing collaborator has integrated it as part of their product. Active collaborators could be described either to use open source software in products or to shape the market situation with it. All active collaborators in this study belonged to the group that uses open source software in their products. Strategic collaborators belong to group that has developed a business of open source software. Altogether, in this study, two different types of open source software use, internal use and product use, were classified as the same type of open source software collaboration, observing collaboration. Therefore, when comparing the two divisions, they cannot be described as identical. Although both classifications considered open source software use, additionally the collaboration division presented in this research, considered the importance of open source software in the company business, attitude towards the community, and the balance between taking from the community and giving to the community.

Another interesting classification of SMEs is that of the DBE project itself. In the deliverable D28.1, the SMEs have been grouped in four clusters: drivers, discoverers, implementers, and users. The SMEs were characterized by two dimensions: their

technological and behavioral ability to execute the DBE and their willingness to engage in the DBE.

Drivers have strong capabilities and interest to participate in the DBE at a very early stage. With a profound background in the relevant technological and business concepts, they can absorb the necessary DBE knowledge faster than other participants can. When compared to the classification presented in this study, strategic collaborators could be described to possess quite the same characteristics as drivers. However, the DBE division is not emphasizing the nature of open source software use of the company as much as the division in this study. Therefore, in addition to the two strategic collaborators, there is one more company that could be classified as a driver, thus the number is altogether three.

Implementers have similar technological capabilities as drivers that allow them to develop components and contribute to the DBE. However, this larger fraction of developer SME is expected to show a less steep learning curve and reduced engagement when it comes to building the DBE community than the drivers. Implementers could be described as active collaborators, when compared to classification presented above. Thus, there were two of them in this study.

Discoverers have limited technological capabilities and other business interests than technology development. Therefore, they are mainly able to use DBE services and are not expected to contribute with components. However, they are strongly engaged and are expected to see a significant individual case in using the DBE. Discoverers are a group that no SME was classified to in this research. Accordingly, it is no wonder that the category was not found either. However, the category is important and needs to be taken into account when planning training activities of the DBE project. Users have similar initial capabilities to discoverers. However, they are expected to be more focused on individual application and personal benefits and be mainly passive users of the community. This category is comparable to the observing collaborator group presented earlier. Thus, two SMEs can be classified in this category.

Altogether, the classification presented in the DBE training strategy and the one presented in this study seem to be quite identical. Although the reasons for different categories were a bit different, the outcomes are still the same.

2.5 Requirements engineering recommendations

The fourth and last sub-objective of the research was to identify requirements engineering recommendations for the DBE.

In the context of DBE business modelling has been seen as a way to identify the business problem to be solved with software, and its results as an important source of the first requirements for software. According to the references studied and found for this research, it was argued that business modeling is rarely performed in the open source community. In the interviews, it was found out that the SMEs do not perform business modeling either; instead, they use the traditional requirements specification as a starting point for capturing requirements.

Thus, the SMEs do not expect open source projects to perform formal business modeling. In case the SMEs had requirements of their own for open source software, they were willing to perform business modeling themselves when needed, and to give the results to the open source project to be further developed. However, the companies needed the open source project to communicate back as well. The open source project should somehow inform the original producer of a requirement of how is the requirement evolving and when it is going to be published.

Requirements capturing, analysis, and design have been identified as the next phases of requirements engineering. The capturing phase includes listing candidate requirements, understanding the system context and capturing both functional and non-functional requirements, which are then refined in analysis and design phases for the implementation of software.

Based on existing literature it was assumed that in open source projects requirements are more informally discussed and modeled than in proprietary software companies. However, the interviewed companies had requirements engineering practices that were not as far from open source practices as expected. It seemed that the SMEs had quite informal requirements practices, which were suitable considering the size of the companies and the size of their customer projects. Additionally, all companies tailored their ways of working if needed in case of a bigger customer that requested a more formal process.

What is formal enough then? The companies stated that documentation in open source software projects, in general, is quite good, although it is not as thorough as proprietary software documentation. Even some UML-models had been found, at least in some bigger open source projects. The most important documentation, the companies thought open source software projects should always provide, is a description of the software architecture and interfaces.

Again, the companies were willing to give their requirements to open source projects, and even to perform the analysis and design phases as well, in case they considered it beneficial and they thought the open source project way of performing the job was too informal. However, once again, the SMEs needed open source projects to inform the evolution of requirements somehow back to the requirements origin. Additionally, managing the requirements more openly and clearly was requested to ease independent information searching of open source projects.

Requirements management, decision-making on requirements priority and changes is performed throughout the requirements engineering process. Usually customers make the decisions, based on timeline and work estimates given by software companies. In the interviews, the active role of software producing companies came up. Active role in these cases means giving the customers advice, occasionally strong recommendations, on what would be the best possible solution for the customer, even if the best solution were not to implement the requirement at all. Additionally, the companies transformed the feedback given by customers into software requirements, only few companies stated that their feedback is refined enough to be implemented into software as such. Companies made independently decisions about implementation order and timelines as well. Thus, the role of customer as a sole decision-maker is not as strong as it used to be, the companies have taken a more active role in the process with their expertise.

When comparing that development with the fact that open source projects are typically seen as dictatorial communities, where a maintainer or a maintainer group makes all the decisions, a conflict situation could be near. The SMEs do not consider this kind of arrangement preferable, which came up several times in the interviews. When collaborating with open source projects, the SMEs expect to be taken into account because of their general activity, good ideas, and documents. From the SMEs point of view that should be enough for gaining respect and decision power over an open source project. Based on these case studies, participating in implementing the software, which usually has been nominated as the way of gaining power, should not be a prerequisite for SMEs to have an influence.

2.6 Conclusion

In the DBE engagement and training strategy SMEs have been classified into four groups according to their ability and willingness of collaboration with the DBE: drivers, implementers, discoverers, and users. This study clearly confirms the usability of this classification.

The expectations and needs of information flow and information management practices within these three groups are a bit different. Therefore, although the research problem is answered mostly concerning the SMEs as one group, some aspects are explained more thoroughly group by group.

The **communication facilities** of any project are of major importance. The collaboration between the DBE and proprietary SMEs does not make an exception. Communication arrangements should enable fluent information flows both from SMEs to the DBE and from the DBE to the SMEs.

Traditionally open source projects have communicated via Web-based forums, like web sites, discussion forums, and mailing lists. According to the SMEs under study, the web-based forums are acceptable for the collaboration with the DBE; however, some adjustments can be stated by covering each collaboration group individually.

First, drivers have made a business of open source software and they are actively involved in open source projects. Thus, they are experienced web-based forum users. Additionally, they are experienced in searching information of projects independently. Therefore, drivers might be satisfied with the situation as it is.

Implementers have some experience in using the web-based communication forums of open source projects as well, but since it is not their actual business, they are not experts in it. Implementers need the DBE to reorganize the traditional open source way of communicating in the Web. Information has to be more organized and structured as it now is to make information searching easier and the project way of working more formal.

Users agree with implementers on the need of reorganization and restructuring of web-forums, but in addition, they want more. Users needed face-to-face meetings to be arranged at least in strategically important phases of the DBE..

The future **decision making** principles of the DBE project are even more important than communication practices. Traditionally a single maintainer or a group of

maintainers have made all the decisions in open source projects. People, who have wanted to participate in decision making, have basically no other way to gain decision power than contributing quality software code and preferably often. If the SMEs were collaborating with the DBE open source project, those characteristics can, once again, be considered via those four SME categories.

Drivers know how open source projects work; the decision-making hierarchy of open source projects is not a problem to them. They are also willing to participate in the implementation phase if it is seen economically beneficial. Implementers are not as approving. They want to be taken into account because of their good ideas and requirements. Implementation of software should not be needed to gain decision power. The power should be based on other activity performed for the project or some other solution, for example an open organization, could be formed. Users do not accept the traditional decision making ways of open source projects at all. Their arguments are the same as implementers had. Additionally, users emphasized the limited resources of SMEs, they do not have personnel to participate in open source software implementation and donating money in order to achieve decision power is out of the question as well.

Altogether, the DBE project should notice that SMEs especially need to participate in decision-making concerning software requirements they have requested from the DBE.

Based on the theoretical part of this study, a gap between proprietary companies and open source projects, concerning the **formality of software requirements modeling and documentation**, was expected. However, it barely existed. Nicely working software, produced by open source projects, was more appreciated than formal documentation. Usually, when SMEs are collaborating with an open source project, information flows from an SME to an open source project. The information may consist of for example requirements, comments, and in general, analytical discussion. Those principles were quite acceptable to the SMEs. The drivers and implementers, even one of the users, were willing to model their requirements themselves and pass the information on to the DBE, if they considered the DBE way of modeling requirements too informal.

The fact that DBE utilises advanced modeling and meta-modeling practices, can become a challenge for the adoption of DBE in SMEs. Formal requirements modeling is not typically part of the software process of the SME, but is a standard procedure in DBE. Challenged may emerge especially in the case when an SME has legacy systems that need to be modeled in the DBE and the SME is not used to perform structural modeling with for example UML or MOF.

Traditionally, an open source project communicates back by putting information to be visible in the project web site or equivalent. However, the SMEs needed the DBE to do much more than that. The SMEs, especially implementers and users, wanted the DBE to produce information about that their requirement has been noticed, how the requirement is evolving, and what comments has it received. The DBE should also inform the SMEs about which requirements are going to be included in which release and when are the next versions going to be released.

Information management practices are in an important role, when wishing for a successful collaboration between the DBE and the SMEs. What kinds of information

management practices are needed and when, depends quite a lot on what kind of an SME is at issue. Drivers already possess the enthusiasm and abilities to engage in the DBE. Therefore, to maximize the possibility of successful initiation phase, drivers should be the first group of SMEs to collaborate with the DBE. Drivers should be informed about the DBE in general, but especially about the business potentials and practices, the DBE generates. The drivers can also be the first companies to implement services in the DBE, thus, information about the DBE service development practices would be appropriate. In addition of drivers, discoverers should be taken into account in the initiation phase as well. The discoverers might not possess technical abilities, but their willingness to participate in the DBE is more important. Their information needs are quite the same as the needs of drivers, basic DBE information and information about the business possibilities. In the second phase of the DBE project, the implementers and users should be encouraged to engage in the DBE. As the implementers possess more abilities to engage in the DBE, they should be provided with information about DBE services development, in addition of basic DBE introduction.

Once the DBE is up and running, the web sites, mailing lists, and discussion forums are a good start for information sharing and management, however, most SMEs need something more. Based on their experiences with current open source projects, the SMEs demanded the DBE to organize information in a more structured and clearer fashion. That could mean, for example, creating a suitable, focused ontology to ease the categorizing of information. Within ontology defined tags, web sites and discussion forums can be organized in a way that enables also novice SMEs to search for information autonomously. Easy access to information, openness in information sharing, and decision-making policies are ways of adducing the will of especially implementers and users towards DBE collaboration.

3 Application needs of user SMEs

In the Tampere Region the DBE project has been operated as part of the eTampere programme. eTampere (www.etampere.fi) is a large regional information society initiative aiming at development of information society of businesses and citizens and making Tampere a global leader in the research, development and application of issues related to the Knowledge Society.

Within the eTampere framework, the DBE project has collaborated extensively with eBusiness Research Centre in various research tasks. The eBusiness Service project (<http://www.etampere.fi/eyrityspalvelu>), in turn, has been the major partner when planning the actual implementation of DBE in different user groups and industries.

In the eBusiness Service project the ICT needs of over 400 SME have been analysed in 2003-2005 by a group of qualified and independent ICT consultants. In the analyses the business needs of SMEs have been identified and a set of possible service providers delivering suitable solutions for the specific business case of the SME in question has been recommended by the consultants. The composition of the companies that have participated in the project reflects quite well the overall industrial structure of the Tampere Region (see tables 1 and 2 below).

Table 1. Breakdown of SMEs participating the eSME project by sector (N= 326).

• Industry	65 (20 %)
metal industry 32	
• Telecommunications, ICT	49
• Consultation services for enterprises	39
advertising, engineering, legal offices	
• Retail and wholesale	37
• Building	25
• Events producers, theaters, concerts, ..	23
• Financial services	21
• Other services for enterprises	18
• Other services for consumers	16
• Medical services	16
• Accommodation, hotels, restaurants, ...	11
• Transport, shipping	5
• Others	2

Table 2. Breakdown of SMEs participating the eSME project by number of employees (N= 326).

• 1 employee	51
• 2-5 employees	107
• 6-10	67
• 11-20	39
• 21-30	27
• 31-100	19
• 101-200	8
• over 200	8
 • 1-10 employees	 69 %

The objective of the project has been to take an extremely realistic and customer oriented approach to the ICT needs of SMEs. The mission has not been to introduce new technologies or technology opportunities but to encourage SMEs to implement proven ICT solutions and thus improve their business operations. The consultants have selected from a group of long term professionals in the field and they have a good track record in SME sector. Consultants have been trained and guided to recommend solutions that are able to deliver clear customer value in short time and that are economically feasible. Consultants have also been independent from software vendors. Based on this work and resulting database, the real life needs of local SME can be analysed better than has been the case before.

The results of the analyses have been collected into a shared database that has been made available for the DBE project as well. The need profiles of the participating SMEs are summarised in the table below. For the details of the methodology and project evaluation, see Penttilä (2005).

Table 2. Distribution of ICT needs of SMEs using the eBusiness Service (N= 428)

Web pages, intranet, extranet	70 %
Telecommunications and data security	53 %
Customer relationship management	47 %
Financial administration, for example electronic invoicing	41 %
Infrastructure and equipment	38 %
Exchange of information	23 %
Enterprise resource planning	22 %
Project management, workflow	21 %
Electronic market places	13 %
Office systems and tools	13 %
Education	11 %
Consulting and architectural planning	10 %

To sum up, based on the results of the eSME project, software services that are most urgently needed are expected to:

- Provide cost and time savings in the daily operations
- Improve customer relationships management: services that make the sales, marketing and customer care processes and interaction with customers more effective
- Improve internal communication: services such as intranet, collaborative team work, remote work and project management
- Enable exchange of information between companies: services such as electronic invoices, technical and sales documents
- Be based on low-cost IT-infrastructure that is easy to use and to maintain. There is also a growing need for local IT-caretaker services that would provide services such as installation, upgradings and trouble-shooting
- Allow seamless cooperation between large and small operators, governments and businesses, allowing full interoperability.

4 Driver SME requirements for DBE platform

In spring- winter 2005 a driver-centric strategy for DBE bootstrapping was introduced in all the DBE regions. The strategy aims at integrating to most willing and capable SMEs first to DBE at thus receiving the acceptance and critical mass of SMEs as soon possible. In spring- summer 2005 the effort of regions in SME recruitment will be sharply focused on these few companies.

One tool to be used in the integration process of driver SMEs has been extensive training and consultation of these companies and the most relevant interest groups. Instead of producing training for masses of SMEs, the strategy has been to produce tailored workshops for selected target companies and their stakeholders. One of the most successful and important events in this area so far has been the DBE Technology Workshop organised in the Tampere Region in February 2005.

In the workshop the achieved results and future plans of the development of DBE platform components (Execution Environment, Service Factory and Evolutionary Environment) were presented to SMEs by the main architects of each of these components. In addition, several other presentations from the same technology domains were presented by external well-known technology experts. The aim was to provide SMEs with knowledge that would enable them to position DBE technology offering and value proposition in a wider setting.

As the final step of this two day work a SWOT –analysis on the DBE Platform Technologies was conducted by the participating SMEs. In this analysis the SMEs identified, working as a group, the main strengths, weaknesses, opportunities and threats of the DBE platform components. In the analysis no explicit ranking was given and the companies were expected to reach a consensus report. The existing strengths and future opportunities can be interpreted as (potentially) fulfilled requirement. On the other hand the weaknesses and threats can be interpreted as (potentially) not fulfilled requirement.

From the SWOT analysis the following requirements for the platform of Digital Ecosystems can be derived:

1. No single point of failure and control

Digital ecosystems should not be dependent upon any single instance or actor. From the technology point-of-view this refers especially to the utilisation of P2P-technologies and from the organisational perspective to balanced and decentralised governance models.

2. Commitment to Open Source and Open Standards

The platform of a digital ecosystem should be based on open source technologies and open standards. Open standards mean open access to the specifications and free usage of the standard.

3. Long-term credibility and attractive brand

When aiming at piloting new technology with real business, the sustainability of the technology in use is a central success factor. Long-term credibility is crucial for adoption by SMEs and can be enhanced by several measures, including EC

instruments, but especially by having support from large players in the software industry, from large established development communities and/or from standardisation organisations

4. Utilisation of proven technologies

The platform should reuse whenever possible standards and technologies and should build on top of previous successful research and development. However, in the areas where new technologies are deemed necessary, they should provide enough competitive advantage against existing integration solutions and emerging proprietary products to justify the cost of switching.

5. Simple on the surface, performant technology underneath

The use of the technology should be simple and easily integrated into the daily operating mode of the SMEs; the underlying technology should enable the SMEs to interact efficiently with other, bigger systems (interoperability) and provide the savings the SMEs are looking for.

6. Sufficient trust and identity management and data security

The major current unresolved questions seem to be in the area of trust and security. The absence of reliable and robust solutions to be applied within a distributed P2P architecture could greatly slow down the diffusion and prevent the full realisation of the benefits of the concept of digital business ecosystem.

7. Proven business cases and benefits for service providers and service users

The bootstrap of services in a digital business ecosystem needs to be based on the attractive business cases for service providers and consumers. The business benefits should be clearly demonstrated and widely communicated to the SMEs in different regions and opportunity spaces.

After the workshops these requirements have been communicated to the technical team of DBE. Together with other more informal input from driver SMEs in spring 2005 they have already had some impact on the design decisions of DBE platform.

5 Conclusions

The rich set of SME needs and requirements described above provides a fruitful basis for operational and technical design decisions in the DBE project. In this concluding chapter we take a short look at some possible implications that can be considered. However, it must be noticed, that in this report we can discuss the practical conclusions only tentatively, as the final design and operative decisions can be made by the authorised people in dedicated tasks and work packages. Also it is worth mentioning that as the progress of the DBE project is currently very fast and some of the findings have already been communicated within the DBE project, some of the conclusions have already been taken into account in DBE operations and technology. In the following sections the conclusions are made from several points of views: firstly from the DBE platform, then from the DBE services and finally from DBE operations point of view.

SME needs and requirements seem to have somewhat different links with the main components of the DBE platform. Execution Environment (ExE) consists of the machinery that allows individual services to be dynamically registered, looked up, and accessed over the network. It can be concluded that basically there seems to be clear need for the benefits provided by ExE. SMEs are eager to deploy for solutions that make them automatically and seamlessly connected with their customers, partners and public authorities. With the ExE-enabled connectivity SMEs can achieve concrete improvements in their daily operations. However, it must be kept in mind that in this technology area DBE will face strong competition. There are several competing and/or alternative technologies and products that can provide the same or most of the functionalities that has been planned for the ExE of DBE. Only if ExE is successful in providing some extra features on top of mainstream technologies it can be expected to meet the SME requirements better than its rivalries.

Service Factory (SF) has been seen as a set of development tools for defining and implementing services on top of the DBE platform. It makes use of various modeling and meta-modeling technologies and ontologies. In the period of conducting the research above, it was assumed among the regional players that for SMEs SF would mean a major leap in their requirements engineering practices: a move from mainly informal use of natural languages and textual requirements documents towards the use of metamodeling and formal languages. In this respect the potential benefits of SF would have been quite distant for software SMEs. Even the most advanced SW developer SMEs do not list SF technologies – modelling and ontologies- among their top priorities and interest areas. For example the use of ontologies as part of the software development process is not currently at the scope of SW developer companies, although it is recognized as an issue in SW engineering in general. It might be the case that ontologies are seen as interesting but not yet usable tool for SMEs. From the perspective of DBE adoption this fact would have presented an important challenge.

However, during the course of the project, the role of SF in the DBE system and for the SMEs has become somewhat different. The Service Factory, as understood at the moment and described in the architecture scoping document, does not provide, support or automate the requirements engineering or implementation process for software. The Service Factory can at best be used to create DBE conformant interfaces and their stubs. BML manifests can be created to make a service discoverable and to describe it, but it does not represent a requirements model of the

kind which is produced in software engineering. Thus, there seems to have been a partial internal misunderstanding created by frequent references made to MDA (which is a software engineering methodology) in the context of DBE. DBE does not propose the use of ontologies in the software engineering process of SMEs; they are rather used to identify existing components to be reused and to identify services. In their current role, modelling and ontologies are meant to be mainly invisible for software companies and thus do not represent any significant requirements for the software engineering methodologies and competencies of participating SMEs. However, in the early stages of DBE bootstrapping, the driver companies may have to be able to work closely with these hidden features of SF as well when creating the first integrated service chains.

Evolutionary environment (EvE) represents the most future oriented part of the DBE platform. The purpose of this main component is to dynamically combine various services into a combined super-service with the best possible fit with business requirements. EvE would basically reuse the outputs of SF and automatically hook up individual services to perform an optimized set of actions. Again, from the current perspective of the SME in this study, the vision seems rather distant. The changing customer requirements are widely considered as an important challenge of software business, but there seems to be only little faith in any automated solutions. The needs to get adaptive and dynamic software solutions are not strongly present in the needs of user SME either. It seems that generally SMEs have not yet faced these challenges, are at the lower levels of ICT adoption or are ignorant about the possibilities how to utilise dynamic information systems in their dynamic business environments.

The DBE platform is basically an empty infrastructure that becomes usable for businesses only when services are integrated into it. Thus finally, the extent to which DBE will meet SME business needs and requirements depends on the quality and amount of services that become available. From the results of the analyses above it has turned out that SME needs are mainly in the area of business applications that help them to execute their core customer processes more efficiently. It should be clear that in the bootstrapping and recruitment activities the services and companies working in this area will be the top priority.

On the other hand, if DBE is about to serve these needs and be used in business critical applications, many fundamental requirements for the platform must be met without exceptions. Especially central in this respect will be the requirements for the quality of service and the implementation and management of security, trust and identity.

The software stack of the DBE platform will be licensed under open source licences and the sustainability of DBE will be dependent upon the viability of DBE open source community or communities. The operational model of the community will be an important factor in the sustainability planning and will include e.g. the definition of DBE software process. Procedures for requirements engineering will be part of that process definition. From the software developer SMEs' point of view there seems to be no specific expectations for the requirements engineering. For SMEs, working and well-featured software with sufficient basic documentation is the main thing and there is no significant gap between the requirements engineer practices in SMEs and open source projects in general. The DBE open source project is not expected to make full use of sophisticated and formalised procedures of requirements engineering.

While the SME needs for the requirements engineering in DBE are not especially high, the opposite is the case when considering the expectations for communication and decision making practices. As developers and users SMEs need easy to use and well-designed information sources. For developers www forums and sites can be the main sources of information, but users need also face-to-face communication.

In order to become involved in DBE development, SMEs expect to be appreciated as contributors. As for SMEs even a small contribution to an open development effort is a relatively big investment this expectation can be seen as justified. To be appreciated means that SMEs are involved in the communications loops of DBE community and they have clear visibility and influence to the decision making processes of the community. The traditional centralised leadership model of open source is acceptable only for those SMEs that have a long history in dealing with open source. Here some training needs in the area of open source development and management models can be identified. These issues can also be discussed when defining the governance model of the DBE open source community.

Finally, it is important to realise that SMEs, even the software developer SMEs, can not be treated as a single coherent group. The results of the studies above verify that SMEs will have different interests and agendas when participating DBE. The findings of these studies strongly support the strategy that has been developed in the DBE project in order to have a more segmented approach and to provide different roles for different kinds of SMEs. Drivers, implementers, discoverers and users really exist in the real world and they have different priorities in their needs and requirements towards DBE.

Appendix I

Companies participating the study on requirements for the knowledge management in DBE- SME software developer collaboration

The companies were all software producers from Tampere. The number of employees in the companies varied from two to eighty people, being most often around ten to fifteen people. All companies were growing or were planning and carrying out actions to grow.

Company A provides electronic services for order, supply, and invoicing routines, product data sharing and company's internal network management. It makes the customer processes more efficient by electrifying the connections and thereby minimizing the errors. The company is growing rapidly and has multiple private owners. There is a two-layer management in the company. Interviewee A works as development manager and is responsible for technologies, product development and the coordination of implementation. His work concerns customer requirements and he has good knowledge of open source software.

Company B creates and adds customer value by increasing the productivity of knowledge intensive work and the rapidity of adopting knowledge in organizations. It is a start-up-company with multiple private owners. The company has one management level. Interviewee B brings the market signals in the product development. Moreover, he works with the customer as a consultant. His job includes scanning the customer needs on a general level and it is not related to open source software.

Company C provides services that ease the customers to manage their processes, especially concerning schedule designing and cost controlling. The company has recently begun to grow rapidly, consequently, the number of employees has increased as well. The company has multiple private owners and one management level and it is profitable. Interviewee C works as manager for the employees that actually implement company services. Technical architecture of the company services includes in his responsibilities too, so does sales, as he has good connections with the field. Requirements engineering and open source software belong essentially in his job description.

Company D provides customers services that require special knowledge, which the customers do not have or it is not profitable to have, the company also performs some consulting. The company is owned and managed by one person. Interviewee D does all kinds of tasks from implementing to consulting. His work involves requirements engineering a lot and he is quite familiar with open source software as well.

Company E provides tools and services for competence management, the three areas it is involved with are: education and consulting, content production, and open source software technology services. The company has multiple private owners, one management level and it is profitable. Interviewee E is the managing director of the company. Requirements engineering includes in his responsibilities and he has a profound understanding of the open source software community.

Company F provides services and tools for events. The services and tools accelerate the customer work and increase the quality by diminishing errors. The company is owned by multiple private owners, it has one management level and it is profitable. Interviewee F works as marketing manager. Sales, requirements gathering and management, and supervision of the implementation are included in the job description. Open source software is not very familiar to interviewee F.

Company G provides consulting, for example, for building a business intelligence strategy or an information management strategy. In addition, the company is experienced in manufacturing large-scale software systems based on advanced technological architectures. The company is growing rapidly and has multiple private owners. There is a two-layer management in the company. Interviewee G works as project manager with development of methodologies as one of his responsibilities. He is experienced in requirements engineering and he has become familiar with open source projects, as finding suitable open source components is another responsibility of his.

The DBE project models and classifies small and medium sized enterprise (SME) types in order to relate business processes and needs to identifiable types of business structures. The classification system is based on three parameters: (1) how the company generates or creates value, (2) what stage of growth/maturity the company is at, and (3) what ownership and management structure the company has.

In the interviews, the companies were asked those three questions and additionally some specifying questions when needed. Based on their answers, the classification of the companies can be seen in table 2.

Table 3. Classification of the interviewed companies.

Company	Value C4(M)	Value C4(S)	Value S1	Value S3	Growth	Personality
Company A		X	X		0	H2
Company B	X				0	H2
Company C			X		2	H2
Company D			X		2	L1
Company E	X		X		2	H2
Company F	X		X		2	H2
Company G			X		3	H2

The C4(M) as a value code means that the company manufactures intellectual products, in this case software, according to the company designs. The value code C4(S) means that the company makes software to customer specification.

All of the companies fell into either of these two categories. Most companies also provided intellectual services, which are marked by value code S1. Intellectual

services include for example consulting, designing and educating. Growth code 0 means the company is a start-up company in a way that it is not completely profitable yet. Growth code 2 means the company is profitable and it has a one-level management. Growth code 3 means the company is profitable and it has more than one management levels. Personality H2 means a high-growth company with multiple private owners. Personality L1 reflects a one-owner private company that does not possess high growth characteristics.

Appendix II

Interview guides and questionnaires used in the study on requirements for the knowledge management in DBE- SME software developer collaboration

Working Profile of the Interviewee

1. In which company you are working in?
2. How does the company generate or add value? (Classification!)
3. What stage of growth/maturity is the company at? (Regarding itself)
4. What ownership structure does the company have?
5. What management structure does the company have?
6. What is your role in the organization?
7. In what way is your work related to requirements engineering?
8. In what way is your work related to open source software?

Information needed by the SMEs

1. How do you elicit requirements of your software (Business Modeling, other)?
From whom?
2. How are you requirements analyzed, designed (UML, RUP, MDA, MOF, other) and documented (e. g. locally, Web-based)?
3. How are requirements chosen for implementation (decision-making, prioritization)?
4. How do you get feedback of the software use and how is the feedback transformed into requirements?
5. Have you had any misunderstandings of terms or concepts with you customer and if you have, what kind?
6. Have you defined the terms of your customer domain? Would you consider that useful and how?

Co-operation with the Open Source Community

7. Have you had experiences of co-operation with an Open Source Community, if yes, what kind?
8. What would you expect of developing software together with an Open Source Community?

Ideas of the Requirements Engineering Recommendation

Finally, assess the characteristics for the communal requirements engineering recommendation of proprietary company and open source software project in the list below. They are ideas of what characteristics the co-operation could consist of.

1. Mark the characteristics that you agree to be the current characteristics of open source development. Comment why or why not if you like.
2. Mark the characteristics that are acceptable for you when co-operating with an open source project. Comment why or why not if you like.
3. Add some necessary characteristics/improvements of co-operation if they are missing in the list.

The characteristics of open source software development / collaboration:

1. Open source project communicates through Web-based forum(s) (discussion forum, mailing list, and newsgroup).
2. A (group of) maintainer(s) makes the decisions of the open source project. This means a proprietary company does not automatically have power over the project.
3. Power over an open source project is achieved gradually, by active working with the project (e.g. making a lot of quality code). For SME, this means assigning employee(s) to work in the project and/or contributing the community actively as a company (e.g. like IBM).
4. An open source project is released for production when the maintainers think its ready, not because of a timeline or a need of the SME.
5. Business Modeling is not performed in an open source project. SME may perform it itself and document it in the Web-based archives of the project.
6. The requirements in an open source project are informally discussed and documented in Web-based archives. If SME needs formal modeling, it may perform it itself and document it in the Web archives.

7. Analysis and design of the requirements in an open source project are performed and documented informally Web-based archives. If SME needs formal modeling and documentation, it may perform and document them itself in the Web archives.
8. If SME wants to participate in the decision making of the project, it has to participate actively also in the implementation phase, since contributing quality code is what counts the most in gaining respect and finally power in an open source project.

References

Choo, C. W. 1998. *The Knowing Organization How Organizations Use Information to Construct Meaning, Create Knowledge, and Make Decisions*. New York, Oxford Press. 298 p.

Heliö, J 2004. *Requirements Engineering When Collaborating with Open Source Projects*. Master's Thesis. Tampere University of Technology, Department of Industrial Engineering and Management. 123 p.

Penttilä, L. 2005. *eBusiness Service Case: eBusiness Consultants Help SMEs in IT*. Paper to be presented in eChallenge 2005 conference in 19-21 October 2005 in Ljubljana, Slovenia.