Engineering a Method for Wide Audience Requirements Elicitation and Integrating It to Software Development

Tuure Tuunanen, Matti Rossi
Helsinki School of Economics, Information Systems Science
tuure.tuunanen@hkk.fi, mrossi@hkk.fi

Abstract

In recent years, consumer oriented information systems development has become increasingly important matter, as more and more complex information systems are targeted towards consumer markets. We argue that developing IS for non-organizational users creates new problems, which IS and requirement engineering (RE) community should attend to.

First of all, the elicitation of requirements becomes more difficult as usually consumers do not explicitly know what they want, and it is difficult for them to express their ideas. To support different views of product development, such as project management and design, the method should present requirements in a ‘rich enough’ way to avoid overloading management, but in the same time giving designers the detailed information they need. Last but not the least, the results of requirements engineering should be easy to integrate to the software development process. To this end, we have constructed a new RE method and its support environment within Metaedit+ Meta CASE tool. We based our method on Critical Success Chains (CSC) method, which supports top-down approach for planning, but also provides for wide participation of IS customers to get rich information. CSC aggregates the results of many individual interviews into meaningful graphical models of what is critically important about a potential system. In our work, CSC is extended with customer segmentation and lead user concepts from marketing.

1. Introduction

We argue that the traditional views of requirement engineering process and methods do not live up to the challenge that the information systems engineering (ISE) community is facing when dealing with the development of software and embedded products for mass market audiences. Examples of these new types of software are applications for palm top devices, Java powered phones, and JAVA enabled Digital TV sets.

The companies developing systems for these platforms have been up against the fact that many novel innovations that look good on tests do not win the acceptance of customers.

The received view (e.g. [1, 2]) believes that requirements are out there to be gathered by the requirements engineers, and firms have used managers and engineers as proxies for end-users to develop applications without knowing what the customers want or are willing to pay for [3]. The problem has been that people have thought they only have to find the right informants and use the right techniques to achieve the complete specification. Researchers [1, 2] have seen that by selecting and prioritizing the requirements into usable sets an agreement can be reached on the common goal.

In contrast, we present a view that often the end-users cannot express their needs [4], and as the end-users are scattered and outside the traditional information systems development (ISD) environment, a single organization, it is not a trivial task to reach them in the first place. Hence, questions should be raised and more focus placed on who should actually be the participant in an ISD project. We argue that new methods are needed to elicit the hidden needs of users of information systems that are targeted at a very wide audience of end-users, such as consumers. As many of these systems do not exist currently, we should also consider how to extract requirements and relationships that are not historical. In addition, we argue that these goals and specifications should then be expressed in a semi-formal language that is accessible for both end-users and business representatives. We see this as critical for integrating the more sophisticated elicitation methods with the traditional development process.

In this paper, we seek to achieve these goals with a method engineering approach by applying a method engineering tool. We present a preliminary method
construct that integrates a cognitive elicitation method, critical success chains (CSC), used in Information Systems Planning, supported by the MetaEdit+ CASE tool. In the next section, we define method engineering and then create a Meta model of the CSC method. After this, we present a method construct based on the research done earlier on the CSC method. Finally, we discuss the possibilities of the constructed integrated environment and identify possibilities for further research.

2. Wide Audience Requirements Engineering

Within the Software Engineering (SE) literature, Requirements Engineering (RE) has been focusing on the issues surrounding the problems in eliciting and managing the changing requirements. Requirements are generally specified as something that the product must do, or something that it should achieve when considering quality [5]. However, if considered from the information systems viewpoint requirements can also be defined as descriptions of how the system should behave (i.e. application domain information, constraints on the system’s operation, or specifications of a system property or attribute) [2]. In RE, the requirements elicitation has usually been done at the beginning of the SE process, or continuously during the process as is done in spiral development approaches (for example [6]). Requirements are usually categorized into different levels. The naming of these categories varies, but in this study we divide them into the following three: 1) User requirements are statements in a natural language, plus diagrams, of what services the system is expected to provide and of the constraints under which the system must operate. 2) System requirements define the system services and constrains in detail. 3) The Software Design Specification (SDS) contains functional, non-functional, and domain specific requirements of the system to be implemented.

Most of the current RE approaches assume that the users are known, and therefore the requirements can be elicited from them, using some predefined semi-formal methods. However, in the case of new product development for wide audience end-users this is not so. We define wide audience end-users as a group of end-users that are primarily external to the organization developing the information systems (IS). These external end-users can be consumers, as in the case of the JAVA example described previously. However, the reader should not think that we exclude the internal users of the system totally. Instead, we want to emphasize the new nature of end-users who are not the members of the organization. This leads to problems in committing the end-users to participation in development and, more importantly, actually finding the end-users for the developed system. For systems like these, the traditional tools and techniques offered by SE and RE communities are not suitable, as in many cases we do not even have ways of getting at the users’ opinions.

At present, the dilemma has been approached mainly from the RE process perspective. Researchers have realized that meeting consumers’ demands is different than when developing an information system for an organization. They have pointed out that prioritization of requirements, continuous improvement of requirements, and a short period of time-to-market are vital [7]. However, RE has not been dealing heavily yet with consumer oriented products and eliciting information for these projects. These projects can easily involve something that the end-user has not even thought of being possible, like the ability to download JAVA applications to mobile phones and Digital TV sets. Some of the research on COTS processes and market driven requirements determination deals with closely related areas, but it makes stringent assumptions about the availability of the end-users and the possibility of a relatively linear RE process. However, as Orlikowski [4] has pointed out, when the products are taken into use, they are reinterpreted and innovatively applied by end-users in ways not envisioned by the developers (an example of this is SMS messages, which were intended as operator messages for end users). If so, we still have the question of how to choose the participants to be involved.

Pohl’s [1] model of three dimensions of requirements engineering has been considered a good way of structuring the RE process, and we also choose his approach as the main base structure of our method. The Specification dimension deals with the methods used to gather and organize requirements from the stakeholders. It considers the viewpoints needed to understand the needs of various players and their different method capabilities.

However, we prefer the term “elicitation” to “capture”, to avoid the suggestion that requirements are out there to be collected simply by asking the right questions [8]. The elicitation methods used are mostly unstructured, such as brainstorming, open interviews and document inspection. We suggest a structured cognitive method, i.e. critical success chains [9, 10], for elicitation of the requirements of wide audience end-users.

The Representation dimension of Pohl’s model presents the gathered requirements, using some form of either diagrammatical notation or natural language
prose. In this dimension, the important issues are such as the ease of understanding of the representation, its compactness etc. Traditionally, this dimension has been addressed methodologically by different presentation models, such as Use case and DFD diagrams. To these demands, we present an answer by means of rich presentation of end-users needs [3] and organizing them with a CASE tool in a structured way.

The third dimension, the agreement, in turn deals with the issue of reaching a common vision, or agreement on the key requirements and the goals of the system. To support this, a trend that can be distinguished among the RE community at the present time has been an ambition to integrate the complex requirements specification methods [11] with the traditional RE process and to create the RE documentation following the standards such as [12]. Additionally, researchers have argued that users prefer the use of tools that are embedded within the development environment [13].

We suggest a preliminary solution for the agreement dimension, by using the concepts of CSC to communicate the requirements to different stakeholders, but at the same time managing the evitable chaos of the changing requirements with a CASE tool. In the next section we develop an integrated support environment for the method within the MetaEdit+ tool.

3. Method Engineering

Method engineering provides methods and processes to specify, make explicit, codify, and communicate method knowledge as well as technical tools to enact such processes effectively. In order to model IS development methods we need a set of concepts during ME that can capture the content and form of any development method into a meta-model. In its simplest form, a meta-model is a conceptual model of a development method [14]. Consequently, meta-modeling can be defined as a modeling process, which takes place on one level of abstraction and logic higher than the primary modeling process [15].

During method construction, a meta-model is specified which makes method knowledge explicit. Meta-models, thus, provide a mechanism to collect and organize development experience. For the analysis step, meta-models allow the detection of those parts of the method, which are subject to further analysis.

3.1. Method Engineering Environment

MetaEdit+ is a customizable CASE environment that supports both CASE and metaCASE functionality for multiple users within the same environment. It supports and integrates multiple methods and includes multiple editing tools for diagrams, matrices and tables. Architecture of MetaEdit+ is a client-server environment with the server containing a central meta-engine and object repository and various modeling tools (diagramming, matrix etc.) working as clients. The repository is implemented as a database running in a central server: clients communicate only through shared data and state in the server. All information in MetaEdit+ is stored in the Object Repository, including methods, diagrams, objects, properties, and even font selections. The adoption of full object orientation enables flexible organization and reuse of software components in the environment and a high level of interoperability between tools.

The core conceptual types of a method are defined at the repository level and can be modified by the method developers. Method engineers can change components of a method specification even while system developers are working with older versions of the method. The method can be developed and simultaneously tested in method engineers’ workstation much in the same way as described in [16]. The data continuity, (i.e. that specification data remains usable even after method schema changes), is confirmed by a number of checks and limitations to the method evolution possibilities. The idea is that the user can always be guaranteed data continuity while working with partial methods.

4. Construction of the Method

We begin the task of engineering the method by first choosing the specification language and process. Various methods have been used and are used for the elicitation and text books often mention interviews, scenario analysis, use-cases, Soft Systems Methods, observation and social analysis, ethnographic analysis, requirements reuse, and prototyping. The number of techniques and methods developed for this is almost unlimited and Lausesen [17] describes nineteen different ones. Nuseibeh and Easterbrook [18] have taken a more structured approach and they have developed a classification of methods, which divides them in six metagroups: 1) traditional techniques; 2) group elicitation; 3) prototyping, model-driven techniques; 4) cognitive techniques; 5) contextual techniques. For our paper, we have chosen a cognitive elicitation method.

The selected conceptual structure and process, critical success chains (CSC), originates from Information Systems Science and was developed by [3, 9]. The process of using the CSC is described later in
table 2 when we provide a step-by-step description of the method. However, before examining the process we must define a meta-model. Therefore, we did a simplified version of the Critical Success Chains model using GOPRR meta-modeling language [19] within MetaEdit+. The meta-modeling language is illustrated in table 1. The formal metamodel allows the analysis of the methods conceptual structure, as well as, immediate tool support for modeling using the method.

Table 1 Role based entity modeling (GOPRR)

<table>
<thead>
<tr>
<th>Meta type</th>
<th>Description</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>Consist of independent and identifiable design objects. Examples: an Entity in an Entity Relationship Diagram or a Process in a DFD.</td>
<td>Rectangle</td>
</tr>
<tr>
<td>Properties</td>
<td>Properties are attributes of objects and can only be accessed as parts of non-properties. Properties can be recursive in the sense that they can be lists of other types or references to them. Example: Process Name of a Process in DFD.</td>
<td>Oval</td>
</tr>
<tr>
<td>Relationships</td>
<td>Properties are attributes of objects and can only be accessed as parts of objects or relationships. Example: a Data Flow in a DFD.</td>
<td>Diamond</td>
</tr>
<tr>
<td>Roles</td>
<td>Roles define the ways in which objects participate in specific relationships. Appeared as the ends of Relationships (e.g. an arrowhead) have properties. Example: a Role which an Object plays in a Relationship, such as which end of a relationship is ‘to’ and which ‘from’.</td>
<td>Circle</td>
</tr>
<tr>
<td>Graphs</td>
<td>Collections that can contain objects, roles, relationships and bindings of these, have properties, and model concepts such as a whole DFD. They also hold the information about the connections between graphs.</td>
<td>Window</td>
</tr>
</tbody>
</table>

4.1. Conceptual specification of the method

Figure 1 below shows the conceptual definition of the CSC method according to GOPRR. The method definition is done by identifying the key objects of the method, connecting them by the CSC relationship types and adding properties to those. The CSC relationship types are further explained in section describing the CSC method in below. Roles define how different object types can participate in the relationships, and in our model it describes the interaction connection between the CSC meta-model and the data collect by an in-depth interviewing method stored in a spreadsheet. A complete model of a method is the composition of objects and relationships into a graph, which is one diagram type within the method. Objects can be connected by explosion/decomposition links between the diagram types. When these are defined with their graphical representations, we have a complete conceptual specification of the method, which can be used immediately as a template for new models (see [19] for details). In the end of the chapter, we tie the presented meta-model, figure 2, to the CSC method. We also present goals that can be archived with the engineered method.
4.2. Process specification of the method

The process of CSC starts by selection of participants for elicitation process. For wide audience targeted IS a lead-user concept has been used [10]. Rogers [20] has argued that if we are able to recognize lead-users, people who are among first to adopt new innovations, and what they are interested in, they can be used to forecast what the masses want. One way to select lead-users may be the traditional customer segmenting used in marketing science (for example [21]). This follows the original ideas of wide participation of IS users for reaching cross organization acceptance of IS, and finding the feature set providing the maximum positive impact [9, 10].

Table 2 Critical Success Chains Method - Process description, slightly modified from [10]

<table>
<thead>
<tr>
<th>CSC Process</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Prestudy Preparation</td>
<td>Determine scope to manage complexity. Select participants to represent views you want to understand. May be employees at various levels, suppliers, customers, and experts. Arrange for data collection. Collect interview stimuli.</td>
</tr>
<tr>
<td>2. Data Collection</td>
<td>Ask participant to rank-order stimuli on importance. Ask series of “why would this system be important…” questions to collect consequence and value data. Ask series of “what is it about this system that makes you think it would</td>
</tr>
</tbody>
</table>
3. Analysis
Aggregate personal constructs into CSC models.
Interpret individual statements and label consistently across participants.
Cluster chains.
Map clusters into network models.

4. Ideation Workshops
Elicit feasible strategic IS from technical and business experts and customers.
Recruit workshop participants with technical and business skills.
Evaluate CSC network models and develop 'back-of-envelope-level' ideas for IS projects that satisfy the relationships implicit in the models.
Create brief system descriptions and network business value models for each idea.

After the selection process, the prospects are contacted and during, for example, a telephone conversation, stimuli for the system are collected informally. These are later used in requirement elicitation where the method utilizes a variation of in-depth interview called laddering. It has been used successfully in marketing to define features of consumer products (see [22, 23]). Laddering is based on Kelly’s work in 1930s and 1940s when he worked as practicing psychologist [24]. He argued that by understanding how people see and understand the surrounding world, one can predict their behavior. He modeled how they saw the relationships between states of the universe, the consequences of those states, and the impact of those consequences on their individual values. Hence, by using laddering we can make implicit requirements more explicit and understand what the end-users actually want.

The ladders can be collected during the interview, or in a post-process of transcribing the interviews and reconstructing the ladders. The individual ladders form chains and an example chain, on a product feature level, is illustrated in figure 3. The example is derived from previous research [3]. It expresses participants desire for the ability to pay micro payments (0.1-2 €) with his/hers mobile phone.

In the first studies [9] the ladders were written down by hand with good results. Later in [3] these were reconstructed from interview tapes and input directly to tailored software tool Ladder (please see [3] for more details on the software), which reconstructed the chains to a database. Researchers [10] have argued that chains are more coherent when recorded during interviews.

Previously, most challenges have arisen from branching of the chains. There can easily be three to five chains originating from the same starting point.

![Figure 2 Example chain collected from a participant, adapted from [3].](image)

When considering the requirement explosion that usually happens in an RE project, as well as the described coherency and branching issues of the chains, an appealing way would be using a spreadsheet program, like Microsoft Excel, to structure the ladders. This would also enable researchers to use hyperlinks between the chains that branch out. In addition, it would be a simple task to add hyperlinks of the interviews themselves to a chain with a time index referring to an MP3 file for example.

In CSC, the collected ladders are linked to critical success factors [25, 26] of an information system that are further divided to system attributes, consequences of these and goals or values of stakeholders. In analysis, these are clustered using Ward’s method [27] by researchers and the results are presented using a rich graphical presentation [3] showing the aggregated features, the consequences resulting from them, and the values or the goals that explain why the end-users want these. An example of this is presented in Figure 3 that describes mobile wallet application for 3rd generation mobile phone reported in [9]. The figure demonstrates the importance of different attributes etc. by number of incidents and by emphasizing this with the size of the sphere. It can be seen that ‘access to account’ and ‘ability to pay small payments’ are important systems features and one of the key reasons is avoiding fraud. These, in turn, result in more trust and economic security.
In the next phase, these graphical presentations, or CSC maps, are presented in an initial workshop in order to introduce the key features of the system to the client’s R&D people who evaluate the maps and identify the feasible project ideas. The ideas are developed to a 'back-of-the-envelope' standard, so that participants can identify as many ideas as possible. For each system, the participants are supposed to label the system, briefly describe its nature, its likely architecture, the resources required to develop it, its cost, likely sourcing, the magnitude of the risk, and its expected impact on the organization. Following this, researchers have suggested that the results are then returned to the R&D function of the firm, and used in the information system development process [10].

In our Meta model of the CSC, (see figure 1), we have presented a preliminary approach to integrate the CSC elicitation method and its philosophy of representation of the requirements into the Metaedit+ Method engineering environment. We present slight changes to the original way of the representation by adding three specific features: 1) segment number identifying the requirement (an individual ladder); 2) Chain list object for handling of branching chains; 3) Tape reference connected into each chain. We argue that by these modifications we can tackle most of the difficulties faced by practitioners when considering losing the traceability of requirements to the source [13].

In figure 4 below, we demonstrate the engineered method with a selected section of the examples given in figures 2 and 3. We created an example model that constitutes a part of the CSC map, and the ladders are mapped back to this specific diagram. The model includes objects (attributes, consequences, and values) with references to the individual segments i.e. the ladders in the chain. The objects are arranged according the roles specified in figure 3. In the figure, we demonstrate how easy it is to add segments to the chains in the model. Additionally, we show how it is possible to enable hyperlinking to the original spreadsheet containing the actual interview data. The given example chain also includes a hyperlink into the interview recording that is indexed according to the chain’s starting time.

Figure 3 Example CSC map 'mobile wallet', slightly modified from [10]
4.3. Implications of the constructed method

Our theoretical framework is argued to enable practitioners to avoid three important issues faced daily in the RE: how to handle changing requirements coherently, how to avoid losing the traceability connection to the original needs of the end-users, and how to create RE documents expected by the engineering with no extra effort from the RE analyst.

We believe that our method and tool contribute by integrating complex requirement elicitation methods into the standard RE process, while avoiding the traceability dilemma. However, even more importantly the extended CSC method can be used to identify critical requirements for the project and needs of end-users, which are not explicit. This is an essential area of research considering radically innovative information systems for wide audience end-users. On the practical side, the developed tools allow for immediate tool support for the method and they allow us to change the method easily to accommodate for changed development needs.

5. Discussion and Conclusion

In this paper, we have sought to develop a support environment for a theoretically grounded requirements gathering and organization method. The method is based on the age old theory of personal constructs by Kelly [24] and critical success factors originating from IS [26]. With our theoretical method, we seek to fill the gap in the current literature of RE: the lack of support for the information systems developed for wide audience end-users.

The novelty of the method is its integrated computer support environment for the whole gathering and organization process, which is seen essential for example by Pohl’s framework [1]. We present a sophisticated way for specification (elicitation) of the requirements with a cognitive approach. The approach facilitates extracting requirements and relationships that are not historical i.e. do not have an existing system to compare to. Additionally, we provide a way to reach the external end-users of wide audience IS by using a lead-user [20] concept originating from marketing science. For the representation dimension, we selected to use the basis developed with the CSC.
but extended this with ours. We believe that this kind of support is essential for reaching the agreement [1], and for the tracking of the decisions and their consequences, in other words the traceability of the decisions to the requirements and vice versa (e.g. [13]). A novel approach was to be able to drill down from the high level maps into the original comments in the interviews in the source tapes. This allows the analyst to easily recall the critical requirements for the project, but in the same time it provides the designer more detailed information for design work.

A limitation is acknowledged in the current version of the method. The agreement dimension of the method lacks the validation of the requirements by the end-users of the system. To this end, we aim to create a technique that would produce easy tools for practitioners for prioritizing the requirements with a simple process, and additionally a little effort from the potential wide audience end-users.

In the future, we seek to test the constructed support environment in real world settings. The method is currently in use in a few case organizations in a manual form. In the next phase, we seek to deploy the support environment into a case organization developing new e-business systems. This should lead into refinements in both the method and the support environment.

6. References


J. F. Rockart, "The changing role of the information systems executive: a critical success
