Process Support for Agile Requirements Modeling and Maintenance of E-Projects

Ghada Alaa  
School of Information Systems-Computing & Mathematics: Brunel University

Jaco Appelman  
Faculty of Technology Policy and Management- Delft University of Technology

Guy Fitzgerald  
School of Information Systems Computing & Mathematics Brunel University

Follow this and additional works at: http://aisel.aisnet.org/amcis2006

Recommended Citation
http://aisel.aisnet.org/amcis2006/14
Process Support for Agile Requirements Modeling and Maintenance of E-Projects

Ghada Alaa
School of Information Systems, Computing & Mathematics
Brunel University
Uxbridge, Middlesex, UB8 3PH
United Kingdom
ghada.alaa@brunel.ac.uk

Jaco Appelman
Faculty of Technology, Policy and Management
Delft University of Technology
Jaffalaan 5, 2628 BX, Delft
The Netherlands
j.h.appelman@tbm.tudelft.nl

Guy Fitzgerald
School of Information Systems, Computing & Mathematics
Brunel University
Uxbridge, Middlesex, UB8 3PH
United Kingdom
guy.fitzgerald@brunel.ac.uk

ABSTRACT
Process support for the development and maintenance of e-projects in organizations is increasingly important because systems evolve and become entwined with other systems. The focus of this paper is on support that enables constant updating of requirements after an initial set of requirements has been produced. An approach termed Up-front Requirements Modeling for E-projects (URME) aims to provide a process-based approach enabling organizations to react in a timely and appropriate fashion to changing circumstances. It empowers stakeholders to maintain and improve information systems and the services that these systems offer. In particular the paper reports on modifying URME to include a Collaboration Engineering (CE) technique known as thinkLets. Inclusion of thinkLet-based process designs allows for codification and promises re-use of group processes. The paper reports on experiences of applying thinkLets in an existing local government e-project in Alexandria, Egypt. The experience suggests that collaboration engineering concepts such as thinkLets are a suitable extension to URME and can support agile information systems development and maintenance. The research packages and codifies collaboration tasks enabling the potential transfer of competencies like the design and execution of goal oriented meetings. The CE component in the URME approach is needed to empower stakeholders to undertake key collaborative processes that support the achievement of agility and emergence in information systems-based organizations.

Keywords
IS development, e-projects, changing requirements, agility, emergence, collaboration, thinkLets

INTRODUCTION
The turbulent nature of the e-business environment (Turban et al., 2000) requires organizations to react quickly and creatively to make the most of new opportunities and business models. This can only be achieved when the mental models of members vested with decision making power, pertaining to the particular place, time and circumstance of the e-project are aligned and shared. Once visions are shared and agreed it becomes possible to jointly produce changes/improvements to an information system or organization. Additionally, through sharing and collaboration an organization creates focus with, and for, its members. This constitutes one of the more enduring competitive advantages a firm can develop in fast-changing, highly-competitive environments (Nooteboom, 2000, Birkenshaw & Hagstrom, 2001).

In this paper we report on the development of the organizational competence to collaborate. It is part of a larger research project which aims to develop a methodology to make organizations more agile in their day-to-day business operations and to
accommodate the speedy definition and re-definition of requirements of the information system on which they are dependent. For this paper there is a more modest goal: to create collaborative acumen in organizations that operate in an e-commerce environment.

Learning how to collaborate and the ability to codify and transfer this knowledge is one of the skills that an organization can try to routinize. By learning, internalizing and routinizing how to collaborate and by being able to perform this process in a distributed or co-located environment a firm can transform a skill into a competence and thus achieve a competitive advantage. In other words collaboration and teamwork become part of the culture of an organization, in the sense that the practices become taken-for-granted and are as perceived as self-evident or natural. In this paper a nascent discipline and design approach for collaboration processes, termed Collaboration Engineering, is introduced. The thinkLet concept is utilized to design recurring collaborative tasks, specifically in requirements modeling and continuous adjustment and maintenance of the IT-based application. The scope of this paper is limited to reporting on the first steps of a collaboration engineering effort that has been undertaken in internalizing collaboration competencies within the upfront requirements modeling for e-projects (URME) approach, developed in an earlier study by the authors (Alaa & Fitzgerald, 2004). The first evaluation results focus therefore on acceptance and adoption of the proposed approach rather than trying to measure sustained use.

LIMITATIONS OF REQUIREMENTS MODELING METHODS FOR E-PROJECTS

Vidgen et al. (2002) characterizes requirements for e-commerce projects as vague and imprecise, indeed often completely unknown, and changing frequently. They suggest that requirements evolve through understanding of the application, the circumstances and the customer perspectives, as well as through the experience gained as the e-commerce application is realized. Truex et al. (1999) concur and suggest that the traditional lengthy analysis to obtain complete and unambiguous specifications of requirements is no longer relevant in Internet-time. Instead they suggest a shift to new information systems development (ISD) values, such as dynamic requirements negotiation, continuous redevelopment, and adaptability.

These problems have been around for some time and attempts have been made to address various aspects. Martin (1991), for example, proposed Joint Application Development (JAD) sessions as a means to set up the initial requirements for rapidly-changing projects through brainstorming activities. JAD focuses on customer involvement and collaboration to accurately depict the users’ view in defining the requirements and jointly developing a solution. But the authors feel these are probably too narrowly focused as they concentrate on formal meetings that have defined rules of behavior, e.g. the use of well-defined agendas, keeping official meeting minutes, the involvement of a qualified facilitator, etc. The amount and frequency of meetings involved is substantial. Although such methods are an improvement on traditional requirements elicitation approaches they can still be too rigid and stifle agility. Furthermore the success of JAD is highly related to the facilitator and will depend on his/her qualifications, skills, experience with facilitation, etc. (Wood & Silver, 1995).

In addressing these problems, Ambler (2001) suggest agile modeling (AM) sessions that should be faster than JAD as they are highly iterative, with the agile modeler iterating back and forth quickly between identifying a requirement, analyzing it, and proposing a potential design strategy, all within a short timescale, maybe even minutes. As compared to JAD they are not restrictive and embrace change, as they follow flexible principles that are argued to leverage creativity and fast response, such as active stakeholder participation, collective ownership, displaying models publicly to enhance communication and others (Ambler, 2002). Although AM is designed to set guidelines for effective agile modeling habits it does not specify how modelers can be engaged to adopt these habits as it does not address the content or a structure that defines the modeling process itself. Also AM requires expert facilitators to ensure its success.

UPFRONT REQUIREMENTS MODELING FOR E-PROJECTS (URME) APPROACH

Adopting some agile concepts an earlier study by the authors proposed an ‘e-Commerce Issues Roadmap’ (Figure 1) (Alaa & Stockman, 2001) to identify and analyze global, business value requirements for e-commerce projects. It provides a template that categorizes the different underlying issues and concerns that typical e-projects face, e.g. business, technical, social, legal, political and economic issues. Due to the large variety of potential issues, the proposed roadmap acts as a ‘balance’ between structure and freedom by providing an initial set of topics/issues that need to be considered, thus preventing issues being missed or too much time wasted to identifying them all. The process of identifying the requirements under the categories contained in the roadmap is called Upfront Requirements Modelling for E-Projects (URME). It is based on brainstorming sessions guided by the ‘e-Commerce Issues Roadmap’. As compared to JAD and AM, URME does not concentrate on producing requirements artifacts, such as functional decomposition diagrams, data flow diagrams, etc, which would be too comprehensive and would limit modelers’ creativity and fast response. Instead URME leaves space for the modelers to be
creative as it provides only a starting point, based on the roadmap, to identify global upfront topics. There is of course a need to carry out a comprehensive analysis and to get to the level of detail needed to build the application, but this comes after URME.

URME has been used within a B-B (business-business) e-marketplace project to support agile requirements modeling and it was found to trigger innovative requirements and give structure to the wide range of issues encountered in typical e-projects (Alaa and Fitzgerald, 2004). However, findings from the use of the e-Commerce Issues Roadmap within URME suggest that although it is very useful it is not enough on its own. Just as with AM and JAD it does not lead to a sustainable collaborative modelling process without the use of expert facilitators. In an environment where a facilitator is absent and the time or resources are lacking to execute a highly structured process other solutions need to be found.

![e-Commerce Issues Roadmap](image)

**Figure 1. e-Commerce Issues Roadmap**

**WHY COLLABORATION ENGINEERING?**

Findings from the use of collaborative modeling approaches, such as JAD, AM and URME, as discussed above, indicate that they do not lead to the development and deployment of collaborative skills that, ultimately, culminate in a collaborative competence.

- It cannot be assumed that participants/stakeholders will perform the right or necessary collaboration activities while jointly identifying and defining the right requirements and solutions.

- Most collaborative modeling approaches typically provide a solution direction and important insights on how to develop and maintain collaboration competencies, but without being specific about which actions of individual members must be enabled or constrained. They assume that experts are still needed to facilitate the modeling sessions.

The authors also made this assumption in the first case study of the application of their approach but it was found that the URME did not provide enough support for the members of the group to initiate, execute and thus develop a collaborative
practice/competence, on their own. The URME provided a blue print but it was not clear exactly how it could or should be put into practice.

The conclusion was that the processes of collaboration needed to be addressed more directly within modeling sessions. Although the agile literature emphasizes that speed and teamwork (collaboration) are key when trying to flexibly adapt to changing circumstances and that the culture of the organization needs to be amenable, or even better, supportive of teamwork and self organization. These contributions are valuable in terms of the insights they offer but it is argued that such approaches are not sufficient to instill a collaborative practice. In the following we describe collaboration engineering concepts that were used to complement and further develop URME.

COLLABORATION ENGINEERING AND THINKLETS

“Collaboration Engineering is an approach to support the design of re-usable collaboration processes and technologies meant to engender predictable and success among practitioners of recurring mission-critical collaborative tasks. A mission-critical task is one which creates substantial value, or which reduces the risk of loss of substantial value for organizational stakeholders.” (De Vreede & Briggs, 2005). A design consists of thinkLets, which are “named, tightly scripted, processes for creating a single repeatable predictable pattern of collaboration among people working together towards a particular goal” (Briggs, et.al., 2001). In other words, a ThinkLet is a codified facilitation routine. They have only been relatively recently introduced and have been used in a number of contexts and areas, e.g. brainstorming. Collaboration engineering identifies five basic patterns of collaboration, as follows:

* **Diverge** – move from having fewer to having more concepts.
* **Converge** – move from having many concepts to a focus on, and understanding of, fewer concepts deemed worthy of more attention.
* **Organize** – move from less to more understanding of relationships among concepts.
* **Evaluate** – move from less to more understanding of consequences for choices toward attaining group goals.
* **Build Consensus** (also called Align Goals) – move from less to more agreement among a group of stakeholders; achieving more congruence between individual and group goals (Briggs, et.al., 2003).

These patterns support collaboration design efforts of engineers and facilitators involved in group-based activities. They do so by breaking-down the collaboration process into meaningful patterns of activities to produce sustainable group-based activity results, e.g. beside diverging ideas, ideas are also converged, organized, evaluated and consensus built. Thus, thinkLets promise more predictable and repeatable group-based activities, as they describe in detail how a certain collaboration activity can be realized. This has the added advantage that thinkLets can inform the design of group meetings/sessions as they codify and break down group activities into a series of elementary collaboration steps (Kolfschoten, et. al., 2004).

A number of thinkLets are typically executed in sequence to complete a collaboration process. For example in the approach adopted here a number of thinkLets that relate to the different collaboration patterns have been coupled to design the collaboration process of the Upfront Requirements Modelling for e-Projects (URME). A thinkLet is usually defined in terms of the GSS tools used, the configuration of those tools and a script to be followed during the facilitation intervention (Briggs et al., 2003). Essentially a script contains instructions on what to say to a group in order to enable them to perform a collaboration activity independently. ThinkLets have become one of the cornerstones of the discipline of Collaboration Engineering.

RESEARCH & COLLABORATION DESIGN APPROACH

Research Approach

The steps of the collaborative modeling approach based on thinkLets have been undertaken and refined during an action research engagement within an e-government project in the Communal Service Centre1 in Alexandria, Egypt. The application

---

1 “Communal Service Centre” is equivalent to a Local Authority in UK
automates and links the e-government online services and it had already been developed before the fieldwork was carried out. The fieldwork of this study was initially part of an evaluation of the project as well as a way to track and manage ongoing and evolving requirements. This involved planning for the next phase, which is an expansion of the application to cover all municipalities in Alexandria.

An ‘action research’ approach was adopted because it enabled the researchers to observe and create effective organizational change with the use of thinkLets. Action research is considered a very appropriate strategy to study new or changed systems analysis & design methodologies (Baskerville & Wood-Harper, 1996). The need to intervene in some way to inject the new thinkLets-based URME approach into the practitioner environment make pure ‘case study’ research (which is ‘non-interventionist’) inappropriate (Baskerville & Wood-Harper, 1996, Cavaye, 1996). But the URME (Upfront Requirements Modeling for E-projects) had been developed during a previous action research study within a B-B (business to business) e-marketplace application in the pharmaceutical industry in Egypt. The URME was introduced to address the modeling of agile, upfront requirements that were missing in other requirements techniques as discussed earlier. But a couple of drawbacks were identified: first, a problem of information overload was experienced, due to the large number of issues/requirements produced by the “e-Commerce Issues Roadmap”, and secondly, the lack of co-ordination efforts and control over modeling outcomes without the use of expert facilitators. Based on the drawbacks identified insights from collaboration engineering are used in this study to codify and break-down URME into rigorous collaborative activities. In this way the first action research findings were used to inform the second one, on which the designs of the collaboration approach were based.

During three months of fieldwork the researcher introduced and mediated the use of URME based on thinkLets in the e-government project. The first step (modeling activities) were performed asynchronously by all the different stakeholders in the project that covered the Communal Service Centre, with its different departments, citizens, ministerial representatives, system analysts, software manager, business consultant and project manager. This resulted in a total of 15 hours of modeling activities. The second step took the ideas produced by the stakeholders as input to a group of decision makers and problem owners, including the project manager, the business consultant, and the software manager. This involved a total of 10 hours of synchronous sessions. At the end of this step interviews were carried out with the project manager, the software manager and the business consultant in order to assess the activities and the user acceptance of the approach. The evaluation covered criteria such as, ease of use, usefulness, cost, satisfaction and transferability.

Collaboration Design Approach & Process Description

To design a collaboration approach the researchers made use of the classification of thinkLets that codify the patterns of collaboration they create. Our aim was to identify issues under the different categories of the e-Commerce Roadmap, find solutions/requirements of the system to deal with those different issues, organize similar solutions/requirements together, prioritize them, then take the first steps towards implementing the highly-ranked ones. This yielded the following thinkLets design (Figure 2): Dealers Choice, Leafhopper, Themeseeker, Pop-corn Sort, Bucketwalk, Check-mark and One-up (these are names of different types of thinkLets as follows):

- **Dealers Choice** is a “divergence” thinkLet that codifies brainstorming of a group on multiple topics (known as buckets in thinkLet terminology) according to the expertise of the participants. The name comes from the dealer that distributes playing cards to the other players, so the facilitator of the session “deals” out the topics for the team to address.
- **Leafhopper** is another “divergence” thinkLet that codifies brainstorming of a team on several topics at once, so the participants jump from topic-to-topic, contributing as they are inspired, then quickly moving on, like the leafhopper.
- **Themeseeker** is an “organization” thinkLet that categorises issues based on their meaning or relationship.
- **Popcorn sort** is another “organization” thinkLet that quickly organizes unstructured issues into pre-defined themes or clusters, it is usually used after a Themeseeker.
- **Bucketwalk** is an “evaluation” thinkLet that allows a group to walk through the items that have been put together under a certain bucket/theme/category to ensure that they belong to that bucket.
- **Checkmark** is another “evaluation” thinkLet that let participants select items according to pre-established or personal criteria from a list of items.
- **One-up** is a “convergence” thinkLet that focuses on reformulating and improving ideas, participants identify better ones than the ones that are under discussion. Thus they “one up” with ideas that must be better than the previous ones, with arguments as to why they are better. Evaluation criteria for those ideas also evolve. For a more complete description of thinkLets see De Vreede & Briggs, 2005.
A description of the application of the design is now provided. In the first part of the sessions the researcher arranged meetings with the client (representatives from the communal service centre and citizens), the project owner (project manager from the ministry of telecommunications), the business consultant, and the software manager from the outsourcing company. Participants were left with cue cards to execute the process, or parts of it, at their own time. This part included a total of 15 hours asynchronous modeling activities, as indicated in Table 1. For the second part of the design participants were selected that had appropriate decision-making power and expertise, these included the project manager, software manager and business consultant (Table 1). They could alter the requirements and initiate changes to the information system. They were fed with input from earlier meetings where the first four thinkLets served as the design (Figure 2). A total of 10 hours were spent in the second part (synchronous).

The implementation of the design is now discussed (it is described as if it were a continuous meeting for the sake of brevity). Participants were provided with the general substantive problem description and an introduction to the roadmap. The researcher then told the participants to start brainstorming issues in the category of the roadmap they had most expertise in (Dealers choice). Each time the researcher gave the prompt (read the script) associated with a particular thinkLet adjusted to the circumstance. Brainstorming with the Dealers choice thinkLet stimulates the quick production of a large number of issues at the beginning of an activity, later on the participants were allowed to contribute to other categories. The Dealers choice thinkLet then changed into a Leafhopper where participants found solutions or created ideas on how to deal with a particular issue. They cycled between brainstorming on issues and finding solutions. The different issues and solutions were written on notelets (Post-it type) and were attached to large sheets of paper. These were then reviewed by means of the Themeseeker thinkLet, which was iterated, producing a project hierarchy. The issues and solutions were then transferred to the new categories/themes identified via a Pop-corn sort. All actionable items were reviewed and the redundant ones plus any duplicates removed in each category by means of the Bucketwalk thinkLet. This is where URME was most useful, it provided categories to support the sorting of the items. Then the solutions were Checkmarked and rated. Participants could evaluate solutions in each category according to the criteria of: Urgency (red checkmark), Easiness/Quick wins (green checkmark) and Usefulness, but not urgent nor easy to implement (yellow checkmark). Each idea was marked by each participant, and they were only allowed one mark per item. Then the group made a final ranking by reviewing urgent and quick win items and assessing which solutions should be implemented first. At the end participants focused on producing better ideas to implement ranked items.

Different parts of this process were performed with different groups. Some of them just brainstormed issues and used a leafhopper and a bucketwalk and sometimes a checkmark thinkLet. The final decision makers went through the whole process. The last meeting produced a list of prioritized actions to be undertaken and requirements that needed to be met, linked to different parts of the project. It produced an implementation plan based on the collaboration processes and the set of thinkLets of Figure 2. In the next section some of the results will be shown. Detailed results of the outcomes of the sessions can be obtained from Alaa (2006).

<table>
<thead>
<tr>
<th>Sessions</th>
<th>Participants</th>
<th>Total Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Part</td>
<td>Representatives from the Communal Service Centre Citizens Software manager Business consultant Project manager</td>
<td>Total of 15 hours (Asynchronous)</td>
</tr>
<tr>
<td>Second Part</td>
<td>Software manager Business consultant Project manager</td>
<td>Total of 10 hours (Synchronous)</td>
</tr>
</tbody>
</table>

Table 1. Participants and Total Hours of Sessions
FINDINGS

The research set out to design and execute a process whose components are easily transferable, could be executed with simple means, at low cost and that was re-usable. These were conditions that needed to be fulfilled in order to give a process design a chance of becoming sustainable in the context of a low-tech resource environment. The objective was to open up the possibility to instill organizational processes and skills in routines that, if used repetitively, could become organizational competences.

It was found that groups could easily execute the different thinkLets to good effect and none of the participants expressed unease with the process. As the primary field researcher stated: “I just told the participants what they needed to do and off they went. I did not intervene, just observed, and found they could do it themselves…”. The only knowledge that has been transferred is the essential information needed to execute a thinkLet and after the session the participants only kept the outcomes and cue-cards used. This seems to indicate that thinkLets can be executed without a professional process facilitator, except for the researcher giving the prompt and explaining the substantive model. Further, none of the participants expressed dissatisfaction with the process and outcomes, even when pressed. They indicated, for example, that “The process is useful as it made it possible to incrementally specify the project’s requirements and highlight the most important ones that need to be dealt with immediately”. Participants were pleased with the outcomes. When compared with the earlier use of URME, before introducing thinkLets, too much unstructured information was produced that did not lead to decisions on what actions would improve or change a system in a desired direction (Alaa, 2006). The fact that they now had a list of things that needed to be done and the recognition that they shared a similar kind of urgency was, for the group, a big achievement. Participants expressed interest in the process and indicated that they will continue using it in the future.
CONCLUSIONS

It is argued that requirements modeling using thinkLets fills some of the gaps and problems identified in other techniques, such as JAD and AM. Such methods provide guidelines but lack mechanisms for breaking down and codifying the collaboration process. Further, the execution of thinkLets can be done without the presence of a professional facilitator. This does not mean that there is not a role for someone to trigger the thinkLet by means of reading the script and the specification of the action(s). The use of thinkLets in this context is not completely new. They have previously been used to support requirements elicitation and negotiation for services and information systems (Hengst et.al., 2004, Grunbacher and Briggs, 2001) but what is new is that thinkLets were used within an existing IT-based development framework/methodology and that relatively little time was spent on preparation and training of the participants.

In conclusion it is suggested that a Collaboration Engineering approach allows the packaging and transfer of design and collaborative competencies that enables relatively inexperienced participants to execute goal oriented meetings/group tasks and group process designers to design such meetings. In this sense CE seems to be living up to its promise. The participants liked the process design very much, they found it easy to undertake, and they produced good results which shows that thinkLets are capable of supporting process design for relatively unskilled process designers. The stakeholders indicated that they intended to use it again, although this has to be proven, it is an indication that the designs are perceived to be easily transferable and sustainable. We cannot say with certainty that we have created a collaborative acumen in the organization under study. However the conditions of low cost, easy understandability (low conceptual load), easy execution with simple techniques (in this case pencil, paper and Post-it notes), were met.

This approach has been applied to a requirements modeling method, i.e. Up-front Requirements Modeling for e-Projects (URME) in a case study, and whilst one case is inadequate for making any great claims, it is suggested that it is an interesting and potentially beneficial approach. Further, Collaboration Engineering bridges the gap between agile methods and effective collaboration self-organization, and emergence. By specification of individual activities that members of a group have to perform to obtain results it becomes possible to instill collaborative acumen in organizations. If the collaborative efforts support the organizational goals of flexibility and agile (re-)development of information systems such a method is a welcome and an important addition to the current URME-approach. It is of course not limited to URME and any methodology that supports agile information systems development and maintenance where design and, probably more importantly, execution, of collaborative and emergent processes is required is likely to benefit from the inclusion of thinkLet-based process designs.

REFERENCES