

# Visual Requirement Specification In End-User Participation

Copyright 2006 IEEE, originally published in MERE-Workshop 2006

Asarnusch Rashid, David Meder, Jan Wiesenberger, Astrid Behm

*FZI Forschungszentrum Informatik*

*Research Center for Information Technologies at the University of Karlsruhe*

*{rashid, meder, wiesenberger, behm}@fzi.de*

## Abstract

*End-user participation in requirements engineering (RE) is important, but not frequently used at the moment. Reasons are the large expenditure of time for organizing and carrying out surveys as well as the time it takes to understand the users' requirements and to formulate them textually. But even textually formulated requirements are not sufficient, as users are no experts, do not have enough time beside their job to model complex requirements and describe their requirements properly. It is therefore necessary to formulate the requirements visually. Actual methods are too complex and inefficient, because they need training and can't be integrated into the day-to-day business of end-users.*

*Thus, users should be able to annotate their ideas directly on their screen and submit them to a web based collaboration platform. The main goal is to obtain complete requirements and, simultaneously, to save the time of all stakeholders in the whole development process.*

## 1. Introduction

Reasons for the failure of projects are mostly to be found in the process of requirements engineering (RE), shown by several surveys of the Standish Group [17]. Primarily, this is caused by missing or incomplete requirements. Takahashi and Anton [15] discovered that unsuccessful communication is often at the root of inadequate requirements specification. Zave and Jackson [18] recommend that the terminology used for requirement specification should match the environment in which the machine (e.g. software) is going to be built and used. When the language is peppered with jargon, i.e. technical terms, the language will be ambiguous. Requirements analysts and end-users essentially speak different languages. It could

happen that the languages share terms that are used in different ways [6].

Successful communication is hindered when only textual language is used for communication. End-users are often not able to formulate requirements detailedly by textual methods, but software engineers usually prefer detailed formal descriptions. Acclimating end-user to the formal diction (of the IT) collapses not only at the missing IT-terminology but also because of the missing time and practice in such complex tools.

The topic of this position paper are ways of enabling end-users to formulate the requirements in a visual way. Main goal is to obtain complete requirements and simultaneously to save the time of all stakeholders in the whole development process by supporting end-users and software-engineers. As end-users are busy with their day-to-day business, requirements acquisition should be integrated into their working-process with structured visual specifications to improve efficiency, completeness and accuracy of participative requirements elicitation as far as possible. Therefore, users should be able to annotate their ideas directly at their screen and save them into a web based collaboration platform.

This paper is structured as follows. First, existing approaches in methods and tools for integrating end-users in requirements acquisition are discussed. The goal is to motivate the importance of visual specifications for end-user participation. Next, an overview of existing approaches in visual requirement specification is given and faults in terms of the addressed issues are revealed. Finally, we outline our approach and illustrate a possible solution by our annotation tool.

## 2. Approaches in Requirements Acquisition

Currently, a lot of tools are available supporting the process of requirements acquisition [9]. Hoffmann et al. [8] specify requirements for these tools and accentuate the need for the feature that every

requirement can be shaped freely within the selected tool, as this is one of the few possibilities to collect all requirements of every user.

Actual methods for acquiring the requirements assume that a requirements engineer understands the problem and considers the user as a source of information. In general, the user is merely integrated in the development process in a limited way, even though numerous studies ([7], [10], [12], [17]) postulate that users should participate in the whole development process.

## **2.1 User Participation in Requirements Acquisition**

Generally, existing methods for user participation deal with face-to-face conversations and surveys [5]. Though this allows full requirements specification, it also leads to big operating expenses [4]. Reasons are the large expenditure of time for organizing and carrying out surveys as well as the time it takes to understand the requirements of the users and to formulate them textually. This keeps organizations from integrating end-users in their software projects.

Further, end-users are highly interested in tracking the development of their submitted requirements. By using web-based collaboration tools it is possible to support this kind of distributed requirements acquisition and to allow end-users to track their requirements. Current collaboration tools, however, only provide text-based input options. Normally, users are not at all experienced in software development and are fully involved in their day-to-day business. Thus they speak another language than the developers and this makes textual requirement specification insufficient.

According to the needs of end-users, most of their requirements concern modifications of the graphical user interface and of existing functions. Therefore, it appears that textual requirement specifications will not be adequate to articulate end-users requirements.

On the whole, it is obvious that participative acquisition of end-user requirements with textual specification is insufficient. Visual requirements specification methods appear well-suited.

## **2.2 Visual Specification of End-user Requirements**

There are numerous commercial tools supporting visual specification of requirements. The best-known notation methods are Use-Case diagrams of the UML. These formal techniques support primarily software engineers and presume skills in modeling languages.

Furthermore, there are many approaches of alternative specification techniques in order to achieve end-user participation. Reeves and Shipman [16] describe a combination of visual and textual elements. Li et al. [11] develop a solution, which uses Natural Language Processing. Video techniques are used in the scenario "Software-Cinema" of Bruegge et al. [3], which addresses requirements acquisition in mobile environments.

Mockups [14] and Rapid prototyping techniques [2] are commonly used in early phases of software projects. They do allow software engineers to sketch screens with low functionality as well as to run first usability tests. Tools supporting these techniques are applied by engineers and analysts but not by the user. Users are merely enabled to return feedback in the usual way.

According to our topic, Moore [13] offers an interesting approach, which is based on requirements acquisition using GUI elements without functionality. End-users 'will create mock user interface constructions augmented with textual argumentation that will act as communication to software requirements engineers' [9, p. 2]. The idea is to allow users to construct their own user interfaces. The disadvantage of this idea is that actual software are very complex and end-users will not have enough time to construct a whole software system.

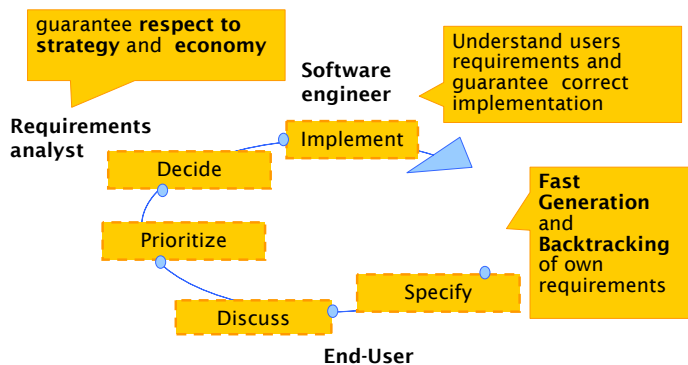
For this purpose, the tool Annotate Pro [1] provides functions to draw annotations directly on the users' screen by using snapshots in combination with ordinary picture editing functionality. This enables end-users to draw comments on their active applications, without time-consuming trainings and preparations. The commented snapshots serve as requirement specification and can easily be sent to the software engineers by email. As this tool neither contains a method, which follows a well-structured plan nor provides a formal notation language there does not exist any common language either. Furthermore, there is no possibility for end-users of tracking their submitted requirements.

In summary, the problem is that none of the present tools and approaches provides all essential functionality to support end-users in an adequate way. All of them are lacking either usability, efficiency or structuring.

## **3. Main Theses**

In the previous chapters, we have shown, that end-user participation with visual requirement elements provides high potential benefits for RE. In order to

avoid disadvantages in efficiency and usability, a new way of requirements acquisition is necessary.



**Figure 1: Workflow in requirements engineering with end-user participation**

According to the need for complete requirements specifications end-users have to be able to specify their requirements generally during their work, beyond of sporadic surveys in some software projects. Textual requirement specifications are not enough, as users do not have the time to get deeply involved in the system and therefore need to be assisted during the specification of their requirements. Visual requirements specification seems to be well-suited for this challenge. Further, tools have to be quickly and intuitively usable without time-consuming trainings. From the software engineers' point of view end-users have to specify their requirements in a well-structured way.

This requires a tool with which the requirements can be formulated, administrated and used during day-to-day business in a short period of time. There are two ways end-users can be integrated into the traditional requirements acquisition workflow. First, whenever end-users feel unhappy about their application they can submit their requirement specification in the form of a suggestion. Secondly, software engineers and requirements analyst invite end-users for participation on actual projects. The working processes (Figure 1) between the stakeholders are simple: End-users sketch and discuss their requirements and the requirements analyst checks the relevance of each requirement. All of them can take part in the discussion regarding the requirements. This makes it feasible for software engineers to explain technical possibilities and for requirements analysts to track the progress of the submitted requirements.

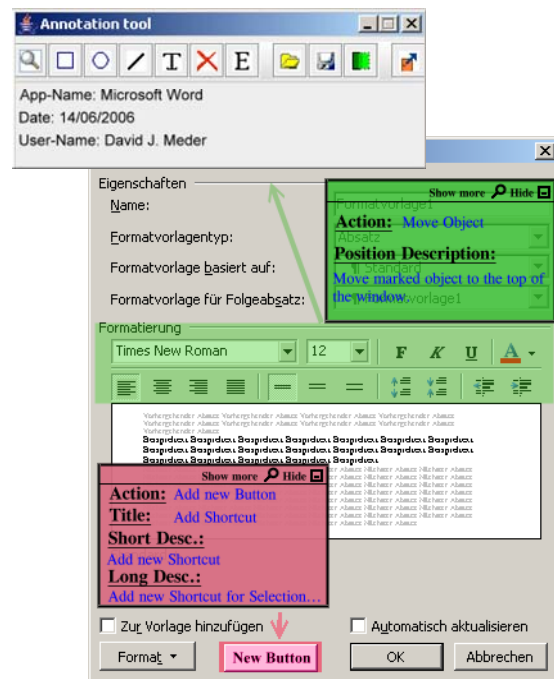
In the annotation tool end-users are able to draw comments on their screen, independently from their applications. Using the annotation tool, requirements need to be specified with a few actions of the user.

Moreover, the visual annotations must be structured and important information must be received through dialogs. Additionally, window information has to be gathered automatically (like in Revelation<sup>1</sup>) and added to the annotation. The annotated requirements are transferred in a web-based collaboration environment, providing access to the submitted requirements and enabling discussions and traceability.

In the following chapter we describe the tool that has been implemented by the Research Center for Information Technology at the University of Karlsruhe.

#### 4. Possible Approach: Annotation Tool

By starting the program, users see a toolbar (Figure 2) with which they can place drawing-objects and textboxes on the surfaces of their active applications and construct visual specifications of their requirements. In case of the software development from the scratch, where no application exists before, mock-up screens can be used instead.



**Figure 2: Drawing the requirements 'Modify position of font formatting' and 'Add new button (Define Shortcut)'**

<sup>1</sup> Snadboy Software, Revelation: <http://www.snadboy.com>

The visual specifications consist of a set of atomic specifications. For example, if users want to modify the position of a button, their specification contains the two atomic specifications 'Mark the desired object' and 'Point the new position'. Thus, it is possible to model a large set of (normally very different) specifications with the help of a small set of atomic specifications.

In order to acquire specifications following a well-structured plan, end-users will be assisted by special specification templates. There are specifications elements which belong to nearly all types of changes, like moving, resizing, adding or removing a GUI-Element. But there are also special specifications like adding a button, altering the navigation of an application, changing the sorting for a list or the padding of cells in a table. Every specification template guides the end-user through single steps and gathers important input data from the end-user by a dialog. If, for example the user wants to change the position of an element, he is asked at the beginning to mark the desired element and to mark the new position for the element thereafter. An additional effect is that unexperienced users are facilitated by the assisting dialog.

When finished with sketching, the visual requirement specification is saved in the issue tracker on the requirements platform (Figure 3). Additional information like username, application title or command-line parameters is extracted automatically and will be saved in the database of the platform.

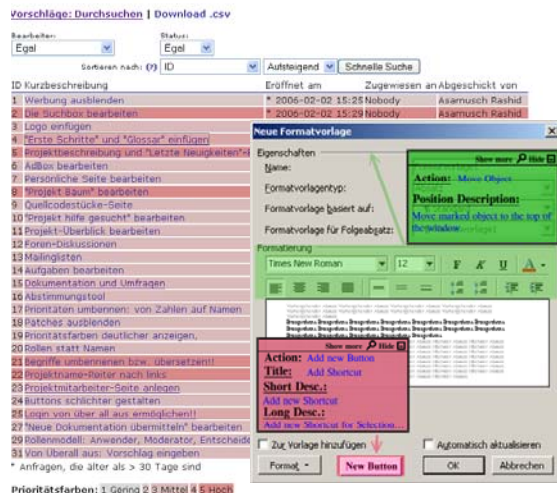


Figure 3: View submitted requirements

The requirements platform is based on the Open Source project GForge<sup>2</sup>, whereat enhancements in end-user usability of the platform were performed. Furthermore the supply of functions is (target-group oriented) reduced to a necessary quantity. Requirements can be added and discussed in the form of an issue tracker.

## 5. Outlook

The evaluation of the benefits by carrying out case studies will be an important basis for future work. Connections between employee satisfaction with regard to the direct influence on the design of their own used software, and the acceptance of the above described annotation tool are the decisive factor for further developments.

In particular, case studies have to point out how much actual effort will have to be spent by using the system described above and in which amount requirements will have to be specified. The degree of quality in the end-users' specification is another important key figure, which defines the applicability of the submitted requirements and determines whether the users' specification are effectively helpful. Further, it is not yet clear how far end-users have to be supported by specification templates without restricting the users' creativity and, simultaneously, understanding the users' drafts with a reasonable effort.

There are several challenges when trying to integrate such a requirements acquisition system in an organisation. Motivating end-users to take part is of vital importance. Further, any formalities like data privacy have to be complied with. For example, if the annotation tool is used in an application which contains private data, it must be clear how to handle the annotated screenshots. Furthermore, software engineers and requirements analysts have to familiarize themselves with responding to end-user requirements more often. The integration of specifications annotated by the end-users into software engineers' methods and tools is another challenge. For example, the enhancement of UML could be useful.

On the whole, all depends on the acceptance and efficiency of the integration of such a system. By 'all-time' acquiring of end-users requirements organisations can make themselves more flexible and improve their long-period planing, as future tendencies of end-users needs are recognized in a shorter time. It should be investigated, if GUI-dependent and workflow concerning requirements could be specified

<sup>2</sup> GForge: <http://www.gforge.org>

structured by end-users with easy usable tools. In the center of interest are audio-visual tools as well as the effort of assistants, which should supply the end-user by modeling his requirements.

## 6. References

- [1] Annotate Pro, <http://www.annotatepro.com/>, viewed on 13.06.2006.
- [2] P. Beynon-Davies, S. Holmes: Integrating rapid application development and participatory design, In: IEEE Software, Volume 145, Issue 4, 1998, pp. 105-112.
- [3] B. Bruegge, O. Creighton, M. Purvis, Software Cinema, CHI Workshop on Identifying Gaps between HCI, Software Engineering and Design, and Boundary Objects to Bridge Them, Vienna, Austria, 2004.
- [4] D.E. H. Damian, A. Eberlein, M.L.G. Shaw, B.R. Gaines, Using different communication media in requirements negotiation, IEEE Software Volume 17, Issue 3, May-June, 2000, pp. 28-36.
- [5] J.M. Drake, W.W. Xie, W.T. Tsai, I.A. Zualkernan, Approach and case study of requirement analysis where end users take an active role, Proceedings of the 15th International Conference on Software Engineering,, Baltimore, USA, 1993, pp. 177 – 186.
- [6] P. Ehn, Playing the Language-Games of Design and Use-on Skill and Participation, ACM Conference on Supporting Group Work, Palo Alto, USA, 1988, pp. 142-157.
- [7] H.F. Hofmann, F. Lehner, Requirements: Engineering as a Success Factor in Software Projects, IEEE Software Volume 19, Issue 4, Regensburg, Germany, 2001, pp. 58-66.
- [8] M. Hoffmann, N. Kühn, M. Weber, M. Bittner, Requirements for Requirements Management Tools, Proceedings of the 13th IEEE International Conference on Requirements Engineering, 2004, Berlin, Germany, 2004, pp. 301-308.
- [9] C. Hood, A. Kress, R. Stevenson, G. Versteegen, R. Wiebel, ix Studie zum Thema Anforderungsmanagement – Methoden und Techniken Einführungsszenarien und Werkzeuge im Vergleich, Heise, Hanover, Germany, 2005.
- [10] N. Juristo, A.M. Moreno, A. Silva, Is the European Industry Moving toward Solving Requirements Engineering Problems?, IEEE Software Volume 19, Issue 6, Madrid, Spain, 2002, pp. 70-77.
- [11] K. Li, R.G. Dewar, R.J. Pooley, Computer-Assisted and Customer Oriented Requirements Elicitation, Proceedings of the 13th IEEE International Conference on Requirements Engineering, 2005, Edinburgh, UK, 2005, pp. 479- 480.
- [12] L. Macaulay, Requirements for Requirements Engineering Technique, Second International Conference on Requirements Engineering, (ICRE'96), Colorado Springs, USA, 1996, p. 157.
- [13] J.M. Moore, Communicating Requirements Using End-User GUI Constructions with Argumentation, Proceedings of the 18th IEEE International Conference on Automated Software Engineering ASE'03, Montreal, Canada, 2003, pp. 360 - 363.
- [14] J. Nielsen: Usability Engineering, Elsevier Academic Press, San Diego, USA, 1994.
- [15] C. Potts, K. Takahashi, and A.I. Anton, Inquiry Based Requirements Analysis, IEEE Software, Volume 11, Issue 2, 1994, pp. 21-32.
- [16] B. Reeves, F. Shipman, Supporting Communication between Designers with Artifact-Centred Evolving Information Spaces, Proceedings of the CSCW '92, Toronto, Canada, 1992, pp. 394-401.
- [17] Standish Group, Chaos Reports 2003: <http://www.standishgroup.com>.
- [18] P. Zave,, M. Jackson, Four Dark Corners of Requirements Engineering, ACM Transactions on Software Engineering and Methodology, Volume 6, Issue 1, 1997, pp. 1-30.