

AgileTool - Managing requirements in Agile WWW projects

Mike Arvela¹, Matias Muhonen¹, Matias Piipari², Timo Poranen³, and
Zheyang Zhang³

¹ Ambientia Ltd., Visamäentie 35, FI-13100 Hämeenlinna, Finland
`mike.arvela@ambientia.fi`, `matias.muhonen@ambientia.fi`

² Wellcome Trust Sanger Institute, University of Cambridge, Hinxton, Cambridge,
CB10 1HH, United Kingdom `mp474@cam.ac.uk`

³ University of Tampere, Kanslerinrinne 1, FI-33014 University of Tampere, Finland
`tp@cs.uta.fi`, `cszhzh@cs.uta.fi`

Abstract. Active user involvement and customer collaboration are essential yet hard to achieve in software development projects. Therefore, tools that effectively support communication and collaboration between customers and developers have a real need. In this paper we describe a requirements management tool, AgileTool, which effectively supports customer collaboration in agile web application development projects. Requirements can be attached directly into web pages under development by navigating the pages with a browser. AgileTool is a communication platform that improves product quality by facilitating requirements consistency, traceability, and testability.

1 Introduction

Software development projects are frequently not completed successfully [11]. This is despite extensive research into and practice of software development methods, processes and tools in the industry. Empirical studies reveal many contributing factors to the success of a software project [13], among which user involvement and completeness of requirements are the key contributors; Typically the lack of user involvement results in incomplete and ambiguous requirements specifications. Improving user involvement during software projects unsurprisingly leads to improved user satisfaction of the final product [7].

In traditional software development projects the needs and expectations of the stake holders are initially discovered and documented into a specification during requirements elicitation. The specification comprises services that the system should provide, its constraints and any background information necessary to develop the product [9, 6]. Quality of the documentation clearly depends heavily on interaction between the developer and the client. Furthermore, communication barriers often exist in the client–developer interaction due to different working experiences, mind sets and background knowledge. Ensuring smooth communication and understanding are therefore essential when the goal is to specify

correct and complete requirements. During recent years, agile software development models that try to better meet the needs of customers have become popular. Agile models include e.g. Extreme Programming (XP) [2], Feature Driven Development (FDD) [8] and Scrum [10] that all emphasise working software, customer driven change in requirements, and face-to-face communication [3] over extensive documentation.

We believe that requirement management tools [5] can support especially agile software development projects, when applied in combination with suitable requirements elicitation techniques [14]. Most existing requirements management tools on the market essentially perform the same core functions: they support organising, documenting and tracking requirements for software systems. Some tools such as Telelogic Doors [12] provide a web-based interface or groupware support for reviewing and editing requirements in development teams that are geographically distributed. Such tools facilitate requirements management for developers, but rarely take client involvement into account or provide sufficient support for communication and collaboration.

We have developed a new requirement management solution, AgileTool, that emphasises customer's role and importance in facilitating requirements documentation and maintenance for agile web application projects. Our solution fits in to an agile development process where developers ask users what they want rather than document detailed functional requirements and specifications.

2 AgileTool

We identified two major obstacles for gathering software requirements. Firstly, using document-oriented approaches may result in the client and developer not really knowing if they are speaking the same language in terms of what exactly should be done, how the final product should work and what it should look like. Secondly, any additional work or tools that are introduced in a web development project are often perceived by project workers as something that slows down development and adds unwanted complexity.

To avoid these problems, AgileTool was created with the following principles in mind: it should be very easy to use and should embrace the IKIWISI (I Know It When I See It) principle [4]. AgileTool allows for attaching requirements directly into the view of a website under development and makes it possible to navigate the requirements of every individual page on the fly. Apart from effectively facilitating client-developer communication, we have emphasised three aspects of software requirements: *consistency*, *traceability* and *testability* [6]. Requirements are made traceable and consistent by allowing customers and developers to visually match every requirement to a view of the website. Traceability in turn supports test case design which further leads to improved quality.

AgileTool enables attaching requirements directly into the views of the developed web application. Requirements are annotated directly on top of the site while freely navigating it page-in-page. The workflow is based around *projects* which correspond to the developed web applications. A project contains one or

more *versions*, each of which is assigned a specific URL containing an actual installation or development version of the site.

When a user logs on to AgileTool, a project listing is presented. This view provides an overview of the system and also works as a central location for managing projects. Both projects and their versions may be edited and removed, new ones added, and their status changed. Projects may also be quickly filtered based on status (e.g. *'In Progress'*, *'Closed'*).

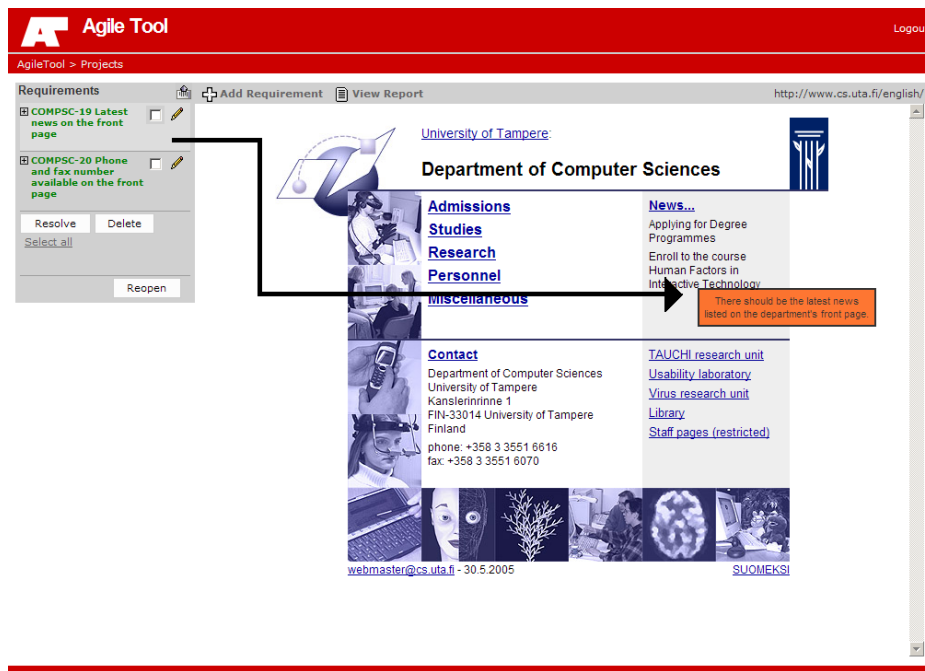


Fig. 1. *The AgileTool Requirements Management View.* The layout is divided into two areas – the web site under development is shown on the right, and the requirements pane showing a list of the requirements is on the left. Note that the web site in question can be browsed freely page-in-page with AgileTool.

When a version of a project is selected, a *Requirements View* is shown (Fig. 1). This is the main window of AgileTool. The layout is divided into two areas, the web page display (on the right side) and the requirements pane (on the left side). The web page display illustrates a web page under development. The *Requirements Pane* presents a requirements list attached to the display on the right. The developed web application can be navigated freely in the web page display area and the requirements list is kept updated to reflect the requirements

relevant for the current page of the application. Furthermore, each requirement can be positioned, and its contents can be tagged on the view of the web page after selecting it. As shown in Fig. 1, the arrow indicates that the content of requirement COMPSC-19 (in orange background) is visually displayed on the web page at the predefined position.

AgileTool provides developers and on-site customers a straightforward view of the web application and the traceability link from individual requirements to the applications under development. The view of the web page demonstrates the features and interactions. It ensures clients and developers have a common understanding of the requirements. Furthermore, it provides good cues for requirements recognition and reminds them of the fine grained functions which are often otherwise overlooked, which facilitates the small release and iterative features of agile projects.

Each requirement is uniquely identified and a description can be included. They are also prioritized and listed in the order of priority. Once selected, they can be marked as either deleted or resolved. After being resolved, requirements can also be reopened if needed.

At any time, the user may choose to view a requirements report for the version. The report is a printable list containing all the views with their attached requirements. The views are shown as thumbnails next to the requirements in order to make the relationship between the textual requirements and the views evident. This view allows exporting the requirements as a paper handout.

3 Conclusions and further work

AgileTool was created with user experience and ease of use as top priorities. We believe that the novel page-in-page requirements editing workflow described above is a rewarding and intuitive way to work for both the end user of the software and the developer.

Given that AgileTool is in its first release iteration, much work still remains. The concept of *use cases* could be introduced to the system and the user interface is also going to be improved to respond better to end user needs.

Currently the software supports only web page -related requirements, the software should allow in future also global requirements and grouping of requirements into different categories.

It is not necessary to use the system only to gather requirements. Any expert (test engineer, usability expert, etc.) can use the system and add when needed special kind of requirements (bug reports, usability requirements, etc.).

A future pilot study has been planned in order to gather user experience to guide the future development of the application. We strongly believe introducing AgileTool to production use will prove beneficial in agile web development projects to come.

Further information on the project may be found at <http://www.agiletool.org/>.

References

1. Ambientia Ltd., <http://www.ambientia.net/>.
2. Beck, K.: *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 1999.
3. Beck, K. et al.: *Manifesto for Agile Software Development*, <http://agilemanifesto.org/>.
4. Boehm, B.: Requirements that handle IKIWISI, COTS, and rapid change, *Computer*, **33**(7), 2000, pp. 99-102.
5. INCOSE, INCOSE Requirements Management Tools Survey <http://www.paper-review.com/tools/rms/read.php>.
6. Kotonya, G. and Sommerville, I.: *Requirements Engineering*, Wiley, 1997.
7. McKeen, J.D., Guimaraes, T. and Wetherbe, C.J.: The Relationship Between User Participation and User Satisfaction: An investigation of Four Contingency Factors, User Participation and Satisfaction, *MIS Quarterly*/Dec., pp. 427-447, 1994.
8. Palmer, S. and Felsing, J.: *A Practical Guide to Feature-Driven Development*, Prentice Hall, 2002.
9. Pohl, K.: "Requirements Engineering: An Overview", Informatik V, RWTH Aachen, Germany, 1996.
10. Schwaber, K.: and Beedle, M.: *Agile Software Development with Scrum*, Prentice Hall, 2001.
11. The Standish Group, "Chaos Chronicles Version 3.0.", 2003. <http://www.standishgroup.com/chaos/>.
12. Telelogic DOORS, <http://www.telelogic.com/products/doors/>.
13. Whittaker, B.: What went wrong? Unsuccessful information technology projects, *Information Management & Computer Security*, **7**(1), pp. 23-29, 1999. <http://www.telelogic.com/products/doors/>.
14. Zhang, Z.: Effective Requirements Development - A Comparison of Requirements Elicitation Techniques., To appear in Proc. *SQM and INSPIRE combined International Conferences*, Tampere, Finland, 1-2 August, 2007.