

Supporting Participatory Requirement Engineering in an ERP Software Community

Harris Wu¹

¹ Old Dominion University
Norfolk, VA, 23529
hwu@odu.edu

Abstract. The paper presents a prototype¹ aiming to widen user participation in requirement engineering for an open source ERP software company. The prototype utilizes a set of templates in a wiki system to allow community participation, and yet maintain necessary processes and models in the software development. The prototype informs developers about the ongoing work of others, in part to avoid conflicts. An XML-based infrastructure is being developed to improve interoperability with other tools such as discussion forums, issue tracking and documentation systems, to further utilize community efforts in software development.

Keywords: requirement engineering, participatory, community, collaboration.

1 Introduction

Participatory and collaborative software engineering has drawn much attention in recent years [1]. In fact, requirement engineering is participatory by its nature. Requirements engineering consists of the cohesive collection of all tasks performed by various stakeholders related to the identification, analysis, specification, and management of requirements [2]. Requirements engineering is the key bridge between users, system analysts, developers and other stakeholders of a software system. As software becomes the core of business processes, and software development becomes an integral part of the global economy, the roles of users, system analysts and other stakeholders are continuously blended.

With the increasing use of packaged software and industry acceptance of inter-organizational collaboration, the “participatory” need of requirement engineering has to be stressed. Compared to proprietary systems, off-the-shelf software packages have much larger stakeholder communities. Information sharing, reuse and web collaboration have gained industry acceptance and even become part of the industry culture, as evident from active user groups and discussion forums in many commercial software communities and the success of open source software. However, the existing collaboration tools for requirement engineering do not always scale to support larger stakeholder communities. The rapid advance of technology (such as

¹ Supported in part by the United States National Science Foundation, Award No. 0725277.

Unified Modeling Language) and the deployment cost of software tools often raise the barrier of entry for participation.

This paper presents a prototype developed to support participatory requirement engineering for an open source ERP software company. The prototype aims to widen and utilize community participation. In the following sections, I first review a research roadmap for collaboration in software engineering, which will be used as a framework in subsequent discussion. Then I describe the software company and its needs for collaboration tools when trying to widen community participation in requirement engineering. I will present the prototype and preliminary results. The paper is concluded with a discussion of the next steps.

2 A Roadmap for Collaboration in Software Engineering

In a seminal roadmap [1] outlined at the recent “Future of Software Engineering” conference, tools developed specifically to support collaboration in software engineering fall into four broad categories: *model-based*, *process oriented*, *awareness support*, and *infrastructure*. Table 1 provides a summary of this roadmap.

Table 1. Four categories of collaboration tools for software engineering

Tools	Support
Model-based	Allow engineers to collaborate in the context of a specific representation of the software, such as a UML diagram.
Process support	Represent all or part of a software development process. Systems using explicit process representations permit software process modeling and enactment. In contrast, tools using an implicit representation of software process embed a specific tool-centric work process, such as the check-out, edit, or check-in process of most SCM tools.
Awareness	Inform developers about the ongoing work of others, in part to avoid conflicts.
Infrastructure	Improve interoperability among collaboration tools, and focuses primarily on their data and control integration

3 Requirement Engineering Challenges to an Open Source Software Company

xTuple is an Enterprise Software company, author of the open source ERP solution Postbooks (<http://sourceforge.net/projects/postbooks>). The company gains revenue by selling training and support for Postbooks implementations. xTuple also sells OpenMFG, a manufacturing-enhanced ERP solution with a community code model built on the same open source code base. In 2007 xTuple was a finalist in eWeek Excellence awards and received a 5 star rating from the Channel Web Network (crn.com).

xTuple receives requirements for its software through two online communities. The Postbooks project ranks in the top 10 on sourceforge.net, had over 100,000

downloads in the past six months and has about a dozen regular voluntary contributors. The commercial OpenMFG community (openmfg.com) has around twenty implementation partners and a hundred licensed customers. xTuple product support and development teams discuss requirements with users in both communities mainly through emails and discussion forum threads, often hundreds of them a day. xTuple hopes to utilize community efforts in capturing, reviewing, organizing, prioritizing, negotiating, cleansing and documenting the requirements using Web 2.0 concepts. To support requirement engineering among a large, diverse software community, xTuple has a number of needs summarized in Table 2:

Table 2. xTuple’s needs for collaboration tools in requirement engineering

Tools	Needs and Challenges
Model-based	As an enterprise software company, xTuple needs diagramming tools and structured documents to support modeling of complex enterprise business processes. However, the tool should not require advanced technical knowledge or commercial software licenses.
Process support	As a commercial software company with a sizable base of paying customers, xTuple needs a rigorous process to manage how requirements are prioritized, priced, negotiated, approved and implemented. However, the cyclic and the parallel nature of its software development and the blurred role definitions (users versus developers) render most workflow tools too restrictive.
Awareness	With a large number of users involved in creating, modifying or discussing the requirements, xTuple needs to be aware, and also keep its community informed, of the constant requirement changes coming from both open source and commercial software communities. Furthermore, it needs requirement provenance, i.e. to figure out why a certain requirement was added or modified.
Infrastructure	Improve interoperability among collaboration tools, including the existing forum, bug tracker and documentation tools

4 Prototype

The prototype builds upon a number of open source tools (Figure 1) to fulfill the needs in Table 2. It also attempts to keep the current tools used by the community intact. The center component is MoinMoin, a popular wiki system. Besides wysiwyg editing, MoinMoin supports editing of diagrams, section editing and conversion of documents into/from XML. Templates for a growing set of requirement categories are being developed, including new module requests, enhancement requests, and bug fixes. The templates are further refined such as by client/server (user interface or application logic) and functional modules (accounting, customer relationship, etc). Users can create a wiki document by answering a simple questionnaire. Or, users can create a new wiki document through a link from the discussion forum (phpBB), and the discussion will be copied to the new wiki document. Various checks are performed to prevent duplication of the same requirement. Once a requirement document is created, it can be re-cast to other templates. The whole requirement

workflow including requirement creation, negotiation, pricing and approval is carried out through fleshing out of the templates by different stakeholders. Different templates result in different workflows. Awareness is supported by email notifications to stakeholders based on the template. Users can register for notifications by modifying the Notifications section of the document. Users can trace the requirements by inspecting the revisions and various metadata on the wiki document/template. The wiki system is integrated with Joomla!, a popular content management system, which provides single sign-on and access to a mysql database. XML is used for moving content in and out of MoinMoin.

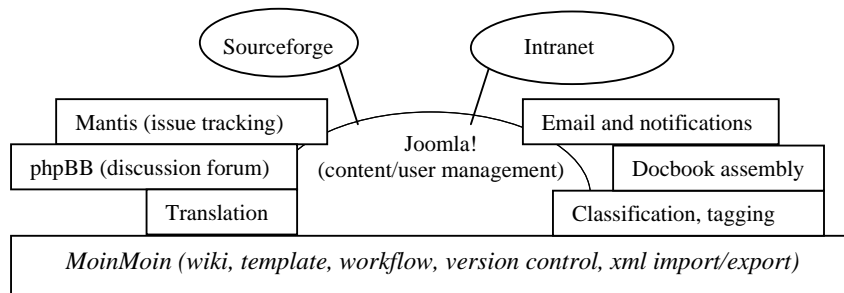


Fig. 1. Overview of the prototype.

5 Preliminary Results and Next Steps

The prototype has learned from the wiki tool in [3]. The wiki approach has gained initial organizational acceptance by xTuple, which is a considerable feat given that several key developers had doubts about using wiki to manage requirements. The effectiveness of the prototype will be assessed by examining the quality and quantity of requirements created, negotiated, cleansed or organized in the system. The prototype is being integrated with several existing tools supporting the xTuple community. The immediate enhancement tasks include classification and tagging for requirement reuse, dynamic translation (using Google), and requirement prioritization (using Mantis).

References

1. Whitehead, J. (2007). Collaboration in Software Engineering: A Roadmap. Future of Software Engineering (FOSE) 2007.
2. IT toolboxwiki, http://wiki.ittoolbox.com/index.php/Requirements_engineering.
3. Decker, B. et al. Wiki-based stakeholder participation in Requirements Engineering, IEEE Software, March/April 2007.