VICTORIA UNIVERSITY OF WELLINGTON

Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematical and Computing Sciences

Computer Science

Specifying Requirements: XP and the Open Source Customer

Angela Martin, Brenda Chawner, Robert Biddle, James Noble

Technical Report CS-TR-03/8
April 2003

School of Mathematical and Computing Sciences
Victoria University
PO Box 600, Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

Specifying Requirements: XP and the Open Source Customer

Angela Martin, Brenda Chawner, Robert Biddle,
James Noble

Technical Report CS-TR-03/8
April 2003

Abstract

Traditional software development relies on an up-front requirements gathering phase that results in a requirements specification. In XP, Beck suggests that we need to reassess this approach to software development, and embrace software development that involves iterative development with constant small corrections. By allowing changes in requirements, the XP approach supports meeting requirements with accuracy. We suggest that the XP approach can work in open source development toward the same result, and by using the same strategy.

Publishing Information

This paper is to be presented at Workshop on Open Source Software at XP2003, the 4th International Conference on Extreme Programming and Agile Process in Software Engineering, Genoa, Italy, May 2003.

Author Information

All the authors are at Victoria University of Wellington: Angela Martin is a PhD student, Brenda Chawner and Robert Biddle are members of the academic staff of the School of Information Management, and Robert Biddle and James Noble are members of the academic staff in the School of Mathematical and Computing Sciences

Traditional software development relies on an up-front requirements gathering phase that results in a requirements specification. The specification, at that point, becomes the sole reference for the system requirements, and we build to the specification and we manage all changes to the specification in a controlled manner. We assume the customer is able to define their requirements at the start of the process, prior to seeing working software, and that these requirements remain stable throughout the course of the project.

In XP, Beck (1999) suggests that we need to reassess this approach to software development, and embrace software development that involves iterative development with constant small corrections. Quality has many dimensions, and two that are often distinguished are precision and accuracy. Definitions for precision typically emphasise detail systematic repeatability, and definitions for accuracy typically emphasise true adherence to external verification. By allowing changes in requirements, the XP approach supports meeting requirements with accuracy.

We suggest that the XP approach can work in open source development toward the same result, and by using the same strategy. In the sections below, we first outline how XP supports accuracy in changing requirements, then describe the open source project we have been studying. Finally, we describe how, together, these suggest an answer to questions about specifying requirements in open source development.

Requirements and the Role of the Customer in XP

The requirements process in XP is different to a traditional software development approach. Central to the XP requirements process is the on-site customer role (Martin et al, 2003). The other significant differences are the informality of the requirements documentation and the iterative nature of requirements gathering.

Informal Requirements: In XP there is a clear separation of roles between the business and technical people. The business responsibilities concern the content of the system, that is what functions will be built and in what order. The technical responsibilities concern how the system will be built and how long it will take to build the system. It is also clearly recognised that the two sets of decisions cannot be made in isolation, as one has a key role in determining the other.

The on-site customer is the person responsible for making all of the business decisions. This person is expected to know the domain well, be able to make decisions, and to be on-site with the rest of the XP team. The on-site customer is responsible for writing user stories and acceptance tests. A user story is a short paragraph description of a system requirement or feature that is:

> *...understandable to customers and developers, testable, valuable to the customer and small enough so that the programmers can build half a dozen in an iteration. ...A user story is nothing more than an agreement that the customer and developers will talk together about a feature. (Beck & Fowler, 2001, p. 45-6)*

An acceptance test script proves the function works as expected by the customer.

Iterative Process: XP projects are decomposed into small public releases of software, and within each release are iterations. Releases are small, typically two to three months. Iterations are smaller, typically two to three weeks. The standard process is:

- The customer prioritises the user stories in business value order.

- The programmers signup to the user stories they will develop and estimate each user story.
- The customer confirms the final scope of the iteration or release, based on the programmer's estimates for each user story and the total amount of time available for the iteration or release. This step can include manipulating the stories, including breaking a story into parts, in order to ensure business value is delivered.

Once the scope is determined, the programmers are able to enter the coding phase.

Requirements and the Role of the Customer in Open Source Development

The library and information management domain has begun to adopt open source software and methods to develop domain-specific applications, with over 80 projects currently underway (Oss3Lib, 2003). One of these, Koha (1999), the world's first open source library management software package, provides a case study to illustrate the role of the customer in the development process. Koha was originally developed in late 1999 for the Horowhenua Library Trust, based in Levin, New Zealand, by Katipo Communications Limited, in Wellington, New Zealand. It was released as open source software under the GPL in 2000. By late 2000 there were 27 registered developers (with 84 subscribers to the development discussion list) working on new releases and fixing bugs, based in New Zealand, Canada, the United States, France, Poland, Germany, Spain, and New Caledonia. The Koha kaitiaki (project manager) is in the state of Washington, USA. Koha is currently being used by libraries in New Zealand, Australia, Canada, France, Italy, and the United States.

Since its initial release, Koha has averaged one major release each year, with additional maintenance releases as necessary. The present Koha development plan is based on a 6-9 month development cycle, with at least one major release each year. The next major release, 2.0, is currently being tested. A lead developer signs up to coordinate each release, and it is up to that developer to clarify the requirements for the features that the Koha community has agreed will be included in the release. A local customer provides detailed requirements, but in other cases, customers can hire a development company to undertake specific enhancements. For example, Philanthropy Australia (based in Melbourne), hired Katipo (in Wellington) to add an abstract field to the main catalogue record, and to integrate URLs into the catalogue (Arkles, 2002). In August 2002, the Nelsonville Public Library in Ohio issued an RFP for a developer to add support for catalogue records in MARC 21 format. The current release is being managed in France, and a library there is involved in initial testing. While the original Koha customer and developer were both in New Zealand, and worked closely together for the initial version of Koha, subsequent releases have been managed by lead developers in Canada (1.2.2) and France (1.9 to 2.0). Customers have been in New Zealand, the United States, Canada, and France. There are few formal requirements documents, and email lists and IRC are used to achieve consensus on new features and strategic directions.

Conclusions

XP supports accuracy amid changing requirements by an on-site customer being actively engaged in iterative development. In the Koha library system, we have found that library system customers are involved in iterative development, taking part by active engagement in electronic communities. Between them, developers and customers participate in managing the project. We suggest this approach shows how the XP approach works for open source requirements specification: a distributed developer community works together with a distributed customer community. In other words, open source development should involve open source customers, on-site and actively engaged, by electronic means.

References

- Arkles, Louise. 2002. Open source library software in action. InCite, 23, no. 11 (November):12.
- Beck, Kent. 1999. Extreme Programming Explained: Embrace Change. Addison-Wesley.
- Beck, Kent, Fowler, Martin. 2000. Planning Extreme Programming. Addison-Wesley.
- Koha. 1999. http://www.koha.org
- Martin, A., Biddle, R., Noble, J. 2003. Being Jane Malkovich: a Look into the World of an XP Customer. Proceedings of XP2003. Genoa, Italy. Springer-Verlag.
- Oss4Lib. 2003. http://www.ossf4lib.org