



**University of
Sunderland**

Hagan, David, Watson, O and Barron, K (2008) Development Experience with Open Source Software. In: 13th Annual UK Academy for International Information Systems Conference.

Downloaded from: <http://sure.sunderland.ac.uk/id/eprint/4130/>

Usage guidelines

Please refer to the usage guidelines at <http://sure.sunderland.ac.uk/policies.html> or alternatively contact sure@sunderland.ac.uk.

DEVELOPMENT EXPERIENCE WITH OPEN SOURCE SOFTWARE AT THE UNIVERSITY OF BOLTON

Dave Hagan (Senior Research Fellow, University of Bolton),

Owen Watson (PhD candidate, University of Bolton),

Kirk Barron (PhD candidate, University of Bolton)

Dave Hagan, Deane Campus, The University of Bolton, Deane Road, Bolton, BL3 5AB

01207 903 796 or 0785 082 4069

d.hagan@bolton.ac.uk

Word count: 1914

Introduction

Over the past eighteen months, the software development team based within the Business, Logistics, Innovation and Systems (BLIS) centre at the University of Bolton have been trialling the development of software using Open Source Software (OSS) tools and techniques. This is a direct result of the research being undertaken within the department with regard to OSS development, project management and Human Computer Interaction (HCI)[1].

Throughout this period, the team has been involved in a number of internal projects including a web-based CV management and Research Assessment Exercise (RAE) compilation tool [1] and a web-based business locating utility. During this time, the team has also been involved with several commercial projects and within the past six months has been involved in projects with several organisations - both utilising software originating from a single software house.

In response to the work undertaken by the team, there has been a proposal for a joint venture between the University of Bolton and a software house, giving the University of Bolton and the

development team within BLIS responsibility for clients in the North West. This subsequently relieves the burden of travel for developers of the software house and consequently allows the University of Bolton to establish greater links with businesses in the region.

The report will briefly examine previous work undertaken by the software development team, the joint venture and the problems associated with using OSS. It will conclude with a summary and reflection upon the findings and ‘lessons learned’ from the work undertaken so far.

Existing Projects

Within the university, several projects have already been undertaken and include the aforementioned CV and RAE management application and the business-location tool [1]. The CV management tool, known as ‘*Tabula*’, was the first major software project undertaken. The decision was made to use this as a research project, allowing the team members to explore methods of project management and new paradigms for design and prototyping. This led to the development of what is referred to as ‘Design-centric Development’, whereby the member of the team researching into HCI would lead the development by designing prototype screens with which the software developers would implement the required functionality. As part of the research, the team also chose to use open source development tools and frameworks with the goal of proving the usefulness of utilising open source frameworks on which to develop enterprise-class applications.

Despite the noble goals of the project, a number of problems were encountered; the greatest of which were related to stakeholder involvement. With minimal communication, the difficulty of agreeing specifications for the application was intensified. This was further compounded by a lack of contact with the proposed users of the software, severely limiting their input and thus diminishing the capability of the developers to successfully employ focus groups whose purpose would be to test the software and analyse layout, feature and functionality.

As with the majority of software developers, a contributing impediment encountered was the volume of work given to the team during the development of the software. This determined the degree to which focus could be allotted to the design-centric approach of development. The pressure to deliver tangible work resulted in an ‘as and when’ project administration with no real project management structure allowed to prevail. The consequence of such hectic software development was the unclear direction and understanding of projects and their respective components, and an uncertainty of project importance.

Joint Venture

As briefly mentioned in the introduction, the University of Bolton will be undertaking a joint venture with a commercial software house, with the University accepting a degree of responsibility for clients within the North West region. The successful management of this work

has the potential to stem future growth within the North West of England, while continuing to contribute toward the strengthening of university links with industry and partners throughout the region.

The initial project to be undertaken by the joint venture involves several universities around the region. They will be collaborating to achieve a unified view of continued professional development with the aid of a software application that will allow potential clients to search for courses that enhance their existing skill set.

This project involves the University of Bolton due in part to the geographical location and proximity to the client, and also due to the client having been exposed to work previously undertaken by the development team. It stands as a test bed for future joint ventures and if successful, could help prove the effectiveness of open source tools and the team's development style and the methodologies utilised.

From the perspective of BLIS, the goal of the project is to refine the usability of the application, increasing client satisfaction with the general appearance through both expectation management and through minor modifications to the application itself.

Within the software development team there is a range of skills including an HCI practitioner. These skills help facilitate a base from which to establish stakeholder requirements and should allow for the improved management of expectations. Should changes be required they can be understood and rationalised within the context of the application, allowing for greater stakeholder satisfaction.

There is a feeling this project has suffered from a lack of understanding on the client's part as to the complexity of the project and the amount of time required to complete to their satisfaction. There is a suggestion there has been client involvement at inappropriate levels of development, with the client contacting developers directly and changes being made which were not part of the original design specification.

It is hoped the collaboration with the University of Bolton will allow, as previously mentioned, greater stakeholder satisfaction through greater management of user expectations and requirements.

Open Source Software

The vulnerability of using open source tools and the inherent risks associated with creating software with these technologies adversely affects the progression of open source development [2]. The evolving nature of open source tools and the uncertainty of their future denote that practitioners must pursue caution before their endeavours begin in earnest. The team were mindful in their choice of technologies; those backed by corporate sponsors [5], a large

developer community [6] or technologies with a mature evolution. Despite this prudent approach, a key component of the team's technological repertoire has been dropped by its primary supporter and has since been merged with a similar technology. The result of this collapse and subsequent merger has concluded with the team's compulsory migration of technologies and reworking of project structures.

One of the risks of using open source tools and frameworks is that the projects can become dormant and unsupported [2]. In general, choosing tools from well known organisations like the Apache Software Foundation will reduce - but not eliminate - the risk [6]. The web framework software that the team chose to use is Struts Shale, so called because it is composed of many layers of software that can be swapped in or out at will, for example Spring MVC and MyFaces (an OS implementation of JSF – Sun's Java Server Faces). Recently, the Struts Shale project has become a victim of its own success. Many of the ideas raised in the project have become incorporated into the new Java Server Faces (JSF2) standard and interest has now focused on the Struts2 project which is plug-in based. Consequently, the number of Subversion (SVN) revisions has plummeted in recent months (a measure of how active the Struts Shale community is) and the news groups are buzzing with the idea that Struts Shale will be subsumed into the MyFaces project [3]. Both Struts Shale & MyFaces are Apache projects and Apache will guarantee that Struts Shale is still supported but possibly not developed further. This is a problem for the team as a heavy reliance is placed on Shale Clay to deliver the design led approach which is a cornerstone of the development methodology. They have even discussed taking over the maintenance of Clay to ensure its continuation.

Problems and Solutions

The development of many software applications lacks a key ingredient. This component has often been a prescribed set of requirements, but the unique development paradigm of most open source applications has meant that this is not necessarily the case. With the growth in the open source sector [4], one of the increasingly important considerations has been the interaction with stakeholders – particularly potential users of software. Obstacles encountered by the team as developers solely for the University of Bolton has been a distinct lack of user involvement and interestingly, minimal access to other important stakeholders. This has meant an unclear set of needs and an unconfirmed list of user requirements. In contrast, the new partnership with a commercial software house has resulted in greater access to potential user base and concluded with an improved understanding of requirements, and greater clarity of direction. The effect has been an accelerated rate of development and a reduction in necessary reworking.

Past labours have included the delivery of not only technical solutions but also content, as many stakeholders simply do not know what they want, or cannot agree on what important aspects necessitate inclusion, and which require further consideration. However, the need to improve

stakeholder involvement as well as user input will be ever more important to fully understand what is required to develop a successful product. Many clients are unaware of the complexity of what appears to be a relatively simple product, therefore it is important to apprise them of the product intricacies. Important areas to address are: flow of problem identification, stakeholder involvement, solution proposal (at a high level), user involvement (which are mapped onto stakeholder requirements), process analysis with rapid interface prototyping, a persistence layer such as a database and subsequently the business logic that provides the functionality of the application.

Conclusion

Irrespective of the maturity of many open source tools and the promise of support by organisation such as the Apache Software Foundation, it is apparent there is a great risk associated with the use of such tools. Developers quickly find new, more interesting projects in which to invest their time, rather than maintaining older, yet stable code-bases.

The transition from a university based software development team to a team more closely associated with industry brings its own share of challenges. While the projects within the university at times lacked focus and drive, newer, external projects are found to be in sharp contrast, with stakeholders requiring more immediate deliverables. Additionally, the need for expectation management of stakeholders has increased, and with it the requirement for greater communication between them and the developers of software. With the experience gained both while developing projects internal to the university, and with the experience of examining external projects, it is clear there needs to be greater stakeholder involvement which, among many businesses, is considered to be one of the biggest risks associated with new software development.

In terms of the internal projects, there was little involvement from the client, and when the software was delivered to the expected users, take-up was severely limited. As previously discussed, this can be attributed to a number of key reasons; the two most important factors being the client saying what they want and how they want it to look, and the need for greater involvement of users during the development cycle. Essentially, user-acceptance testing is often overlooked or rather, yields misleading results because of time constraints.

Further investigation should focus how developers can manage stakeholders within the constraints of an open source development paradigm, and how they can safely adopt open source tools for successful application development.

References

1. Hagan, D. Watson, O. Barron, K. (2007) Ascending into Order: A Reflective Analysis from a Small Open Source Development Team. *International Journal of Information Management* 27 (2007) 397-405
2. Polancie, G., Herieko, M., & Horvat, R. V. (2000). Open source software usage implications in the context of software development. *Informatica*, 29, 483–490.
3. Merging Shale into MyFaces (2007) Available at: <http://www.mail-archive.com/dev@shale.apache.org/msg01519.html> (Accessed: 22 February 2008).
4. Ruffin, M., & Ebert, C. (2004). Using open source software in product development: A primer. *Institute Electrical Electronics Engineers software*, 21(1), 82
5. Feller, J., & Fitzgerald, B. (2000). A framework analysis of the Open Source Software development paradigm. *21st international conference on information systems, Brisbane, Australia*.
6. Iannacci, F. (2005). Coordination processes in open source software development: The Linux case study. *UK E:CO Issue*, 7(2), 21–31.