IMPROVING OOS USABILITY OF SOFTWARE TECHNOLOGY IN HIGHER EDUCATION IN CHINA

[1]**XIANGSHENG KONG**

[1]Department of Computer & Information, Xin Xiang University, Xin Xiang, China

E-mail: [1]fallsoft@163.com

ABSTRACT

The increasing popularity and use of Open Source Software (OSS) have led to significant interests from research communities and enterprise practitioners, notably in higher education in China where this type of software offers particular benefits given the financial and human capital constraints faced. Software Technology assumes the mission of improving Intellectual Property (IP) and software innovation in China. The research looks at implications of the emerging IP regime for China and it attempts to improve software technological capabilities. The paper outlines how OSS can support the educational process of Software Technology in general and the WEB Application Development in particular. The primary objective is to identify suitable tools from the open-source landscape and then validate and prove their integration within a real-life Application Development scenario.

Keywords: *OOS, proprietary software, Intellectual Property, SDLC, CASE*

1. INTRODUCTION

Software is the soul of the computer system. Software Technology is not only the core of information technology industry, but also the important foundation of software industry and information applications [1]. The Open Source Definition is a bill of rights for the computer user. It defines certain rights that a software license must grant you to be certified as Open Source [2]. Those who don't make their programs Open Source are finding it difficult to compete with those who do, as users gain a new appreciation of rights they always should have had [3].

At most universities in China, WEB Development Technology has become the main course used for teaching programming language, Database, Software Engineering, Software Testing, WEB Server, and Network Operating System concepts in Software Technology field. It is also regularly used in coursework, practical examples, and assignments. Its objectives are improving students' vocational skills, and enhancing the students' ability of development of technology and the students' sense of responsibility and mission to improve the professionalism. Most experimental environment in China has installed Visual Studio, Jbuilder, QTP, LoadRunner, Visio, Rational Rose, PowerDesigner, Dreamweaver, Photoshop and other proprietary software such as Windows Server, IIS, SQL Server. In order to reduce software acquisition costs, most universities have to use pirated software to maintain teaching environment. But it is a violation of IP, and erodes the professionalism of the students, and plants a time bomb in the development of information technology on the road. The students in Software Technology field as the founder of the software do not respect proprietary software, thus their own software will not be respected.

Now Software Technology has focused on the use of proprietary software and teaching materials have become a proprietary software brochure, the protection of IP exists in name only in China. Piracies always destroy the professionalism of the students. Domestic software is unable to grow up because of double squeeze between university and foreign software. Domestic software vendors can not provide enough jobs for university graduates. A vicious circle has been formed. How to solve this problem is becoming increasingly important.

2. OUTLET

A primary strength of OSS is its leverage of outside innovation. All are free to take open source software and use it, evaluate it, repair it, and add new capabilities. A key advantage of OSS is that its source code is available, and therefore it can be customized to meet a user's special needs. Cost advantage, flexibility, and better performance are the keys to its success. Governments, business,

education and other institutions around the world have benefited from OSS [4].

Why not OSS? Do any open-source alternatives to the above-mentioned software products exist?

The answer is yes! Thanks to the immense amount of activity in open source software development over the last couple of years, open source versions now exist for most software development technologies.

Table 1. OOS Vs Proprietary Software

Module	Proprietary SDT	OOS SDT
Object-oriented Analysis and Design	Rational Rose or Microsoft Visio	Enterprise Architect
Database Modeling	PowerDesigner	MySQL Workbench
Programming language	.NET	PHP or Java EE
Operating System	Windows Server	Linux
WEB Server	Microsoft IIS	Apache
RDMBS	SQL Server	MySQL
IDE	Visual Studio	Eclipse or NetBeans
WEB Browser	Microsoft IE	FireFox or Chrome
Page Designer	Photoshop, Dreamweaver	GIMP, Bluefish
Functional testing	QTP	Selenium
Performance Testing	LoadRunner	OpenSTA

WEB Development Technology is divided into several modules. Table 1 shows these modules and the software development tools (SDT) of each module. These open source alternatives are all as effective if not better than their commercial counterparts. In fact compared with proprietary software, OSS has most advantages for teachers and students [6] in the teaching process:

1. No restrictions on how software can be used. OSS can save the cost of teaching environment. Costs of procurement and software upgrade are almost zero.

2. Can provide better performance, better stability and better security than proprietary software [5]. Security holes can be found quickly, tested rigorously.

3. Can provide students with an opportunity to study high-quality source and train students in Software Technology innovation.

4. Can cultivate students' ability to write high-quality source, rather than "tools used" capability.

5. Can get a wider range support of platforms and servers.

6. No need to worry about piracy. OSS Can put an end to piracy and helps to protect the IP.

So there's no real reason not to use OOS for software development technology - especially in the teaching process of WEB Development Technology. OOS therefore presents a very valuable resource for educational institutions, which often don't have the financial resources to purchase commercial products.

3. USING OSS IN THE TEACHING PROCESS

Now that we've looked at the role of OSS in the teaching process, let's focus in on how OSS products can be used in this area. In order to better achieve the course objectives, teachers use the "Project-Driven" method and Java EE to develop Web content management (WCM) as a development team. This section examines the capabilities of OSS from Software Development Life Cycle (SDLC) perspective. WCM development relies on a set of tools and technologies to optimize the development process. A typical scenario would be as follows:

1. Requirements management

2. Version control

3. Schedule planning

4. Modeling

5. Coding

6. Testing and issue tracking

3.1 Requirements Management

Most requirements are currently in text-based documents. The team has recently begun representing requirements as elements in a single

model using Computer-Aided Software Engineering (CASE) tools.

1. Manage and deliver large amounts of unstructured material in multiple media.

2. Provide a consistent and predictable information structure, user interface, and navigational mechanism.

3. Enable linking of related materials.

4. Ensure information is up-to-date.

5. Support well-defined roles, responsibilities, and access control for various stakeholders in various departments.

6. Enable workflow between authors, product managers, content administrators, editors, attorneys, and system administrators.

7. Enable the composing and publishing of different views of marketing information for different audiences: financial consultants, clients, and the public.

8. Provide version control to support regulatory requirements.

9. Provide a locking or concurrency control mechanism to prevent two people from simultaneously updating the same content.

10. Enable searching and retrieval of content using the predefined business characteristics of products and services.
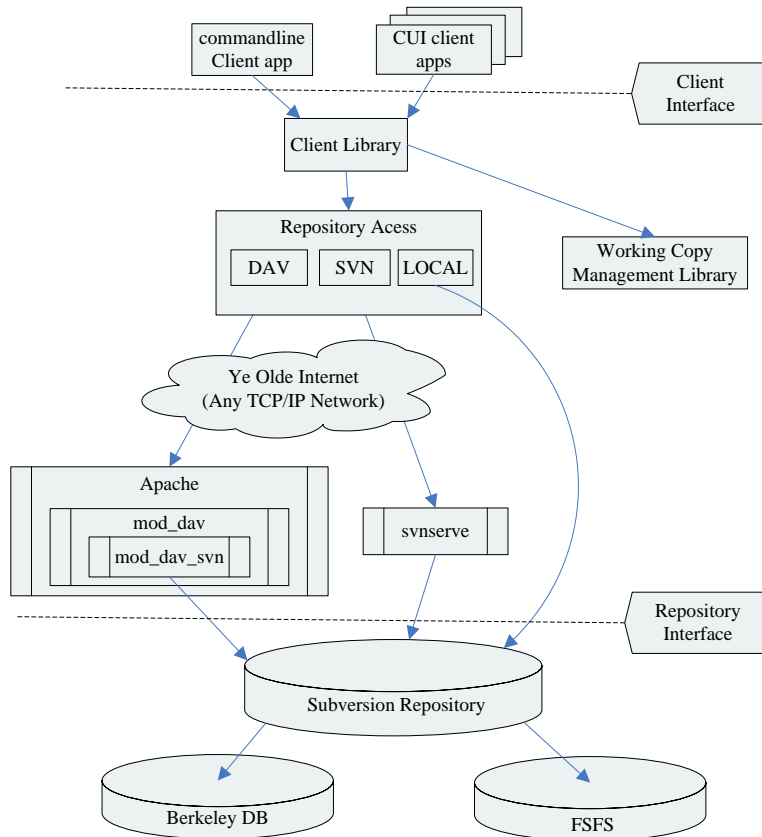

Figure 1. OOS Usability In Teaching Process --Subversion

3.2 Version Control

Version control tracks changes to source code or any other files. A good version control system can tell you what was changed, who changed it, and when it was changed. It allows a software developer to undo any changes to the code, going back to any prior version, release, or date. This can

be particularly helpful when a researcher is trying to reproduce results from an earlier paper or report and merely requires documentation of the version number [7].

Version control also provides a mechanism for incorporating changes (see Figure 1). Subversion [8] is the leading successor to CVS. Its features

include essentially all of CVS's features, with several significant enhancements: it has a cleaner, more reliable, and more scalable implementation; it is based on the existing WebDAV standard; it replaces CVS's concepts of branches and tags with simple naming conventions; and, it has stronger support for disconnected use. RapidSVN and TortoiseSVN are two of several available Subversion clients. ViewCVS can browse Subversion repositories as well as CVS repositories. Also, Subversion repositories can be browsed with any standard web browser and many other applications, due to the use of the standard WebDAV protocol. The team uses Subversion for version control and the TortoiseSVN [9] client for most purposes.

3.3 Schedule Planning

The team utilizes GanttProject [10] for some aspects of planning. Recently the team has adopted the use of the Atlassian Greenhopper JIRA plug-in

to help us monitor our schedule more closely [11]. the following advantages would be as follows:

1. Integration with JIRA

2. Ability to log work

3. Simple interface

4. Ability to track iteration progress and burn-down rate

3.4 Modeling

The team uses Enterprise Architect (EA) [12] for UML modeling and Database modeling applied to the following tasks:

1. Develop the UML profiles (see Figure 2)

2. Develop the Entity-Relationship diagram (ERD) profiles

3. Create the Model Driven Generation (MDG) Technology
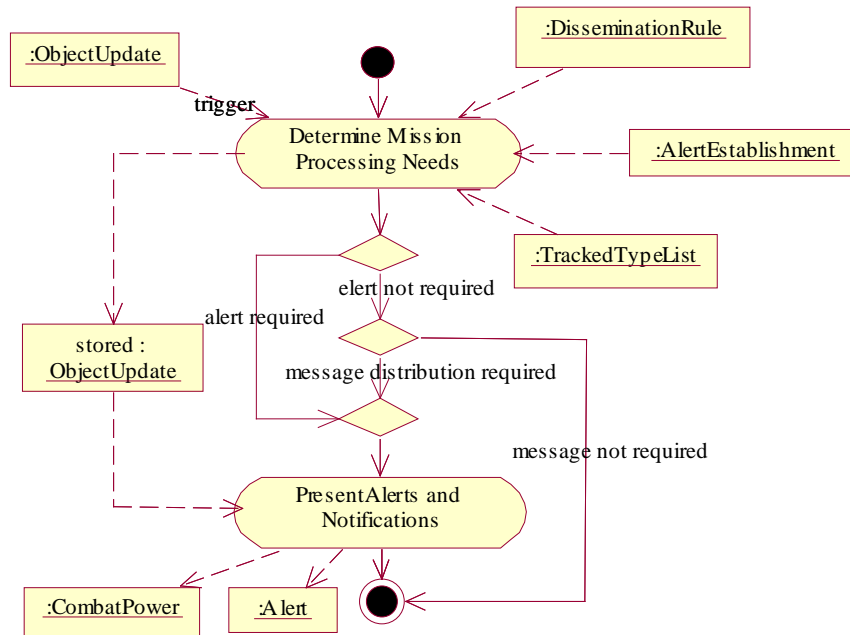
4. Deploy the MDG Technology

Figure 2. OOS Usability in teaching process --EA

3.5 Coding

The team uses EA of Forward Code Engineering to generate Java code, and then uses Eclipse to edit it (Omitted here).

3.6 Testing and Issue Tracking

The team uses TestLink to execute test procedures and to store and report on test execution

results and uses Mantis Bug Tracker for issue tracking .The team creates requirements and develops a test plan (using the structured scenario feature) in Enterprise Architect, then imports the requirements and test procedures into TestLink. TestLink facilitates the execution of test procedures via execution sets, linking test results to

requirements to complete the traceability path [13]. TestLink generates a verification matrix that shows the current status of each requirement (see Figure 3:a), and it can link to Mantis to generate issues when tests fail [14].The test cycle was based on the following activities:
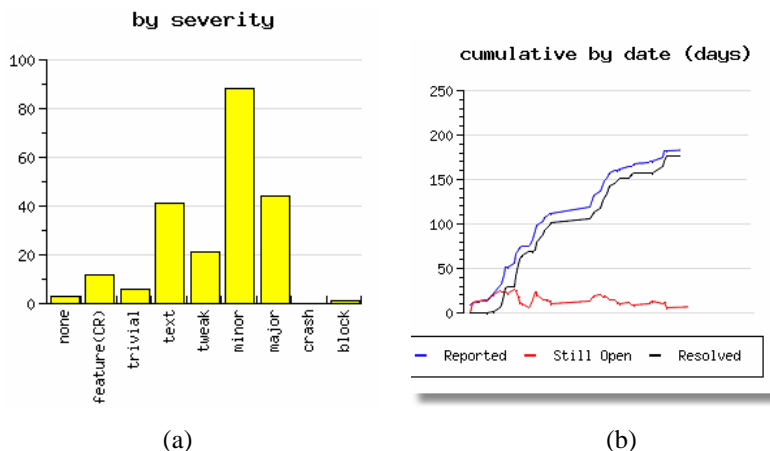

(a) (b)

Figure 3. OOS Usability in teaching process --Issue tracking

1. Plan the test cases for the stories according to the acceptance criteria and emphasizing test cases creation for exceptions.

2. Automated test scripts creation. The test scripts were implemented in the browser Mozilla Firefox, where the Selenium IDE works are recording all the tester actions and turning them into HTML scripts. These scripts were edited and gathered in a test suite, and reused in other test cases.

3. In the Selenium Core tool, the test scripts were executed in Firefox and other browsers. The test cases execution identified application defects and the tester registered them in the TestLink tool. At Mantis, the defects were also registered, described and assigned to the team responsible who was notified about the defect by e-mail.

4. The Reports were automatically generated (see Figure 3:b). The TestLink offered several options to analyze the results based on test cases execution.

5. When the team signaled in Mantis that defects had been fixed, the entire test suite was executed by Selenium validating defects and doing the regression test [15], ensuring that another part of the system was not affected by code changes.

All of the products used in this scenario are open sources, so there is no need for the teachers and students to pay for software licenses. The integration of OSS greatly enhances the educational cycle for WEB Development Technology. This makes it much easier for both the teachers and students to get up and running. There are also many other ways that OOS can be incorporated into the educational cycle. The only limits are the bounds of your imagination [16]!

What's more, advanced students can immediately put their programming skills to use by helping to develop Eclipse, Apache etc. Participation in open source projects also gives students valuable real-world experience with large-scale programming projects.

Some obstacles in the teaching process have to be mentioned when using OOS, mainly in the following areas.

1. Teacher's own lack of capacity constraints the development of OSS in the classroom.

2. Compared with proprietary software, OOS learning materials are difficult to obtain.

3. Lack of OOS certification exam makes lacked motivation of teaching OOS possible.

4. OOS is seriously lagging behind social training module.

4. CONCLUSIONS

This article shows the power of OOS in the educational area. Using OOS for Software Technology is only one example of the many possibilities. What we have to do now is help support open source software development and integration in order to keep open source on the cutting edge. Of course, the use of open source software is a systems engineering in the teaching process, proprietary software can not be achieved overnight and transformed into OOS in China. However, in order to maintain the industry's

capacity for sustainable development, the use of OOS should be earlier access to higher education in China.

REFRENCES:

[1] Robert D Macredie, Kabiru Mijinyawa. "A theory-grounded framework of Open Source Software adoption in SMEs", *European Journal of Information Systems*, Volume 20, 2011, pp. 237-250.

[2] Mingqing Xing. "The Quantity Competition between Open Source and Proprietary Software", *Information Management, 2010 International Conference*, pp. 184-187.

[3] Barry, B.I.A. "Using Open Source Software in Education in Developing Countries: The Sudan as an Example", *Computational Intelligence and Software Engineering, 2009 International Conference*, pp. 1-4.

[4] Margit Osterloh, Sandra Rota. "Open source software development—Just another case of collective invention?", *Research Policy*, Volume 36, 2007, pp. 157-171.

[5] R. Subramanyan, M.S. Krishnan. "Empirical Analysis of CK Metrics for Object-Oriented Design Complexity: Implications for Software Defects", *IEEE Trans Software Eng*, volume 29, 2003, pp. 297-310.

[6] Alfonso Fuggetta. "Open source software—an evaluation", *Journal of Systems and Software*, Volume 66, 2003, pp. 77-90.

[7] Louis Glassy. "Using version control to observe student software development processes", *Journal of Computing Sciences in Colleges*, Volume 21 Issue 3, 2006, pp. 99-106.

[8] Domenico Tortorella, Benjamin E. Gewurz, Margo H. Furman, Danny J. Schust, and Hidde L. Ploegh ."Viral Subversion of the Immune System", *Annual Review of Immunology*, Volume 18, 2000, pp. 861-926.

[9] Thomas, P. "Open Source and Java EE development", *Power Engineering Society General Meeting*, 2007, pp. 1-7.

[10] Luyin Zhao,Fadi P. Deek. "Improving Open Source Software Usability", *AMCIS 2005 Proceedings*, pp. 430-433.

[11] Gopinath, Sreejith. "A DESKTOP CLIENT INTERFACE FOR JIRA", *Partial Fulfillment of the Requirements for the Degree Master of Science in Computer Science*, 2010.

[12] Carolyn Strano, Qamar Rehmani. "The role of the enterprise architect", *Information Systems and E-Business Management*, Volume 5, 2007, pp. 379-396.

[13] Luyin Zhaoa, Sebastian Elbaum. "Quality assurance under the open source development model", *Journal of Systems and Software*, Volume 66 Issue 1, 2003, pp. 65- 75.

[14] Jin Li Xu, Long Wang, Shao Jun Han, Zhao Fu Li, Bin Zhou. "The Design and Simulation of Motor Steering Link Test Bench", *2010 Electrical and Control Engineering*, pp. 3063 – 3065.

[15] Holmes, A., Kellogg, M."Automating functional tests using Selenium", *Agile Conference*, 2006, pp.277-275.

[16] Richard E. Hawkins. "The economics of open source software for a competitive firm Why give it away for free?" , *NETNOMICS*, Volume 6, 2004, pp. 103 - 117.