Identify new Software Quality Assurance needs for the UK e-Science community and reintroduction for the right tools to improve evolved software engineering processes

Victor Chang
OMII-UK, University of Southampton, Southampton SO17 1BJ
v.chang@omii.ac.uk

1. Introduction

Software Quality Assurance (QA) is defined as the methodology and good practices for ensuring the quality of software in development. It involves in handling bug reports, bug tracking, error investigation, verification of fixed bugs, test management, test case plan and design, as well as test case execution and records. Standards such as ISO 9001 are commonly followed for software QA, which recommends using a wide range of tools to improve the existing software engineering processes (SEP) for the organisations [1]. One identified problem is that software engineering is often a fast-evolved development that often the organisation's software QA activities and tools lag behind the actual requirements, thus affecting the measurement of real quality of software products [2,3]. Therefore, the purpose of this paper is to identify new software QA needs and reintroduce the right tools for Software QA, which has made its good impacts for our SEP. Based on the literature [1,2,3], QA experience and our ENGAGE interview records, the following areas that require more focus on Software QA, are described in next section.

2. Tools fit for each type of testing

Visualisation testing: **Cubic Test** is a platform that can be used independently or as an Eclipse plug-in. It is similar to UML or some workflow tools, and its strength is in designing test cases and helping to validate them, particularly for visualised workflow tools such as BPEL.

Virtualisation and platform-deployment testing: Six major tools are evaluated for deployment of clean test environments with different operating systems and Sun JDKs pre-installed. **VMware Server 1.0.5** and **Sun VirtualBox 1.6.0** are our main choices for deployment of server and desktop virtual platforms for testing e-Science applications such as OMII 3.4.2 and Campus Grid Toolkit 1.1. This accelerates the QA progress as both the number of bugs reported and the number of fixed bugs for verification can be doubled or tripled. It is also useful to replicate the system image, saving considerable effort in reinstalling the OS and applications. Virtualisation is useful to test on the latest systems such as Fedora 9, Windows Vista SP1 and XP SP3, as it also reduces the risks involved in installing and running new and untried platforms.

We have also investigated the use of **Altiris 2.1** for software virtualisation to capture e-Science software installations on Windows. The image can be reproduced on other systems. However, this is still in the experimental stage.

Performance testing: **AdvenNET QEngine** can simulate stress environments with many concurrent (up to 10,000) client-server requests. It is useful for testing live websites or services such as OMII 3.4.2. **Apache JMeter 2.3.1**'s functionality is similar to AdvenNET QEngine. Key elements such as hosts, IP, rmiregistry, HTTP/HTTPS requests, Java requests and JUnit requests can be edited on the script before running tests. It has been proven useful for OMII 3.4.2 as it rightly points out software's strength and weakness.

QA Management: **Google Docs** is used for test case design, test plan and test records, where results can be kept up-to-date at a specific URL. **Test Case Manager** is also used to improve the test management process. **ApTest Manager** is an integrated QA management system that allows editing test cases, recording results, checking project progress, delegating tasks, tracking outstanding bugs or requests, tracking each team member's progress and so forth. It is helpful to QA management, although its usefulness must be balanced against its high licensing fees.

3. Further Work

As an ongoing process for continuous improvements, an increasing number of tools will be evaluated, used and implemented, and more discussions will be analysed and described in the AHM 2008 paper.

4. References

[1] Course materials for British Computer Society ISEB Foundation Certificate in Software Testing, 2006.

[2] Hughes. B, Cotterell M., "Software Project Management", Second Edition, McGraw-Hill, 2001.

[3] Thompson, J B, Edwards H M, "How to Teach Practical Software Quality Assurance" An Experience Report and a Working Paper.