

## User Acceptance Testing of Software Application, thought paper

This paper is thought paper instead of white paper, because the content and questions that are addressed are more from emerging thoughts out of experience and user's attitude towards quality of software. Besides, it will leave us with some thoughts and questions to be answered....

Let us demystify the subject we will cover in few paragraphs now. We will spare thoughts on User acceptance testing of software (UAT). I wish to highlight three aspects, one software testing (UAT) is critically essential for quality of software, second it is a continuous process, and third it requires something beyond just testing and that is what I call management of quality. Let me touch up on some myths about software testing, especially from end users' point of view. Academician might have created thousands of pages of write up on software testing, but end users have their own perception of software testing. Rather, I got inspired to write this article from ground level experiences and queries of users with whom I interact.

**Myth:** It is responsibility of software vendor to give quality product. Why should I (user) invest in testing it?

Checking purchased material has always remained user's responsibility. When you are buying a software verifying delivery of software is also a caveat emptor. Software engineering assembly line is not as matured discipline as others like automobile, and hence requires greater efforts to exercise caveat. User Acceptance Testing (UAT) is a globally accepted norms and technique to satisfy users that the product delivered is complete, accurate and acceptable to users. This does not, in any case, relieve software vendor or system integrators from the responsibility of delivering complete, accurate and acceptable software. The gap found in UAT in terms of completeness, accuracy and acceptability of software is a separate subject under SLA management. But verifying the purchased material with due care is prudent buyer behavior. That applies to everything, including software.

**Myth:** We have already gone live; we don't need software testing.

Do you stop changing requirements after going live? Do you stop implementing statutory changes in software after going live? Do you stop evolving software to tap new business opportunities? Do you stop optimizing software and systems after going live? If answer is no (which must be no!!!), then one need to ensure that:

- Any change in software is thoroughly tested for the same completeness, accuracy and acceptance
- All affected programs due to change in software are tested and found working
- Any change has not resulted in defects in working portion of software

If one need to ensure above points, one need to test software before putting any change in software into production environment. Each change is a testable subject.

**Myth:** Let software testing run parallel to our 'go live' plan. We will correct defects after making our software live

This is a grievous error. It kills the very purpose of software testing. If defective software is put into production before resolving defects reported in software testing, the outcome can be one or all of following:

Stretched duration / deadlines of pilot or parallel live of software

Defect reported in live is always top priority for solution, jeopardizing other plans and priorities

Defects in production resulting into user resistance for new software

In short moving software into live environment without completing testing and resolving defects reported in testing can result into vicious circle of priorities and consequent delay in almost everything scheduled thereafter.

Now let me spare next few paragraphs on some critical aspects of making UAT effective and efficient.

#### Creating a specialized team

A pre occupied person with primary responsibilities of anything other than testing will not be able to conduct in-depth testing. Organization should create a specialized testing squad responsible primarily for conducting UAT. The size, composition and mixture of in-house or outsourced can be factored as per industry, availability of skill set and timelines.

#### Reusable library

Reusability of testing assets is one of the Key Success Factor (KSF). Test library documentation and management has highly matured activity in IT industry by now. However, it is often seen that users are not organized in managing testing assets library. One can look for software (licensed or open source) to manage the library of testing assets. User Acceptance testing is always last hurdle that software needs to pass before going live. Hence, all delays in previous stages accumulate on UAT schedule. Generally, UAT is seen suffering from 'No time left' syndrome. Reusable testing assets are critical in achieving multiplied efficiency expectations from UAT team.

#### Automation

Automation is widely adopted by software vendor community. However, the same has not been extensively used by user communities. This is one more approach to handle multiplied efficiency expectation is to gradually automate test library. However, I wish to caution here that Automation will pay off to only those organizations which are seriously interested in Continuous Quality Management. One time testing through automated tools or adhoc approach for testing will result into multifold cost and loss of investment. The primary purpose of automated testing is to verify that changes has not disturbed software portion which otherwise was complete, accurate and acceptable.

#### Integration between UAT and release management

Integration between user acceptance testing and release management is necessary to reap the fruits of investment in continuous quality management. Disintegrated approach between release management and UAT may result into vicious circle of:

Repeated defects in production environment  
Insertion of hidden defects addition into good portion of software  
Reopening of previously resolved defects  
And many more....

Let me clearly state that if release management and UAT are not integrated then, Continuous Quality Management may remain just a dream.

SLA monitoring on the basis of outcome of UAT  
UAT should also generate Continuous MIS to monitor SLA with software vendors. One should monitor defect ratios, reopen rate of defect, defect trending, defect density per piece of software etc. for future improvement and over all benefit of both software vendor as well as user organization.

Organization needs to view UAT Quality management activity as cost saving exercise instead of as a Cost Centre. It saves organization from frequent disturbances into production environment, financial losses because of hidden functional issues in software. Historically it is experienced and proved that correcting a defect before software moves to production environment is much cheaper than correcting the same after software moves into production environment. User Acceptance Testing is last fort to prevent defects from moving into production environment. Hence, conducting UAT effectively, efficiently and continuously pays off the investment in multi fold. The team responsible for UAT should also keep track of value created by them.

Have you ever counted **HOW MUCH LOSS** do you incur due to software problems when,

- Your software **HALTS** during business hours.
- Your software generates **INACCURATE** information
- Resolved software **ISSUES REOCCUR**
- All **INVESTMENTS** in quality control of software are **WASTED**
- Your **SOFTWARE VENDOR** delivers more **PROBLEMS** than solutions
- There are unexplained **DELAYS** in **GO LIVE**

**??????**

---

Author is Director at AUDITime Group, Chartered Accountant and Certified Information System Auditor with 12 years of experience in testing / auditing financial software applications. AUDITime is software quality management specialist with 12 years of experience in capital market, banking, insurance and logistic industry. Any further interest or inquiry can be routed to [madhav.bhadra@aqmtechnologies.com](mailto:madhav.bhadra@aqmtechnologies.com) or [sales@aqmtechnologies.com](mailto:sales@aqmtechnologies.com)