Challenges Faced by Software Testers in Milenium Era: 2010-2015

Pua Pei Ru, *rubypua93@siswa.um.edu.my*

***Abstract*—This paper provides high-level summary of commonly-occuring challenges faced by software testers in the testing process according to eleven major categories.**

***Keywords*—Challenges, Software, Testing.**

I. Introduction

Software testing is doing to find the defects as soon as possible and get them fixed. It is very important in production and maintenance of high quality systems. Generally large organizations spend 60 percents of their development costs in software testing (Bharti Bhattad, Dr. Abhay Kothari, 2014). Basically, software testing includes four major tasks: generating test inputs, creating expected outputs, running test inputs, and verifying actual test outputs (TaoXie, 2014). It is a need to understand the issues faced in software testing so that the new techniques of software testing can be discovered to make the process more efficient.

II. Categories of Challenges

A. General Testing Issues

One of the challenges is the wrong test mindset of testers. The testers tend to show the system is work rather than how it fails. Second is some of the testing is not prioritized. Due to the time and budget, some of the test will be deleted so the important bugs may cannot be find. Moreover, the poor communication among the testers and other developers may fail the software testing. (Donald Firesmith, 2012)

B. Inadequate Test Coverage

Test coverage has gained the attention of researchers as one mean of assuring the adequacy of the testing process. Test coverage is used to measure of how much a program has been exercised with tests. The level of test coverage is an indicator of testing thoroughness, which in turn helps testers decide when to stop testing. The undetermined Effectiveness of Coverage Criteria will lead the problem in software testing. Despite the long time that has elapsed since many coverage criteria have been proposed, there is still no proper means of accurately measuring the effectiveness of different criteria. Although several researchers addressed the evaluation of effectiveness of coverage criteria and comparing them relatively, the results are not sufficient to guide practitioners in the industrial field (Dalal Z. Alrmuny, 2014)

C. Test Planning Issues

There is no separate Test and Evaluation Master Plan (TEMP) but only incomplete high-level overviews. If there is a test plan, test plans also may do not specify test processes but only list and briefly describe types of testing, not how to perform the testing. Sometimes, the incomplete test plans have no clear specific test objectives,testing techniques and test case selection criteria. The inadequate planned test resources also is one of the challenge of testers. The test plans provide inadequate test resources such as inadequate test time in schedule with inadequate schedule reserves, inadequate funding and insufficient test environment (Donald Firesmith, 2012)

D. Modern Life Cycle Issues

The Regression Testing is Inadequate (amount needed is greatly increased). Automation of regression testing is inadequate. Regression tests are not developed iteratively and incrementally. Refactoring may inadvertently obsolete tests and may delete test hooks. Testing assets (test documents, environments, and test cases) are not properly documented (Agile). Testing assets are not adequately maintained (fixed and updated). (Donald Firesmith, 2012)

E. Requirement and Testing Issues

The different type of requirements problems make the testing process become more difficult. For example, unstable requirements cause inadequate regression testing. This is worsened by inadequate automation of regression tests. Ambiguous requirements are easily misunderstood and unverifiable. Moreover, many requirements are missing and incomplete. Furthermore, testing is performed too late which just before delivery will not provide adequate time for defect removal and regression testing. The testing assumes the correctness of test oracles (requirements, architecture, and design). Hence building tests based on inadequate oracles provides false data. (Donald Firesmith, 2012)

F. Unit Testing Issues

The inadequate low-level requirement make the testing process become more difficult. The derived requirement merely restates its parent requirement. Allocated requirements are not derived to be at the proper level of abstraction. Then, inadequate detailed design is also one of the issues. There is insufficient detail to drive testing. An unstable design leads to obsolete testing. Next, there might be lack of pre-test peer review. There is no pre-test peer review of the test input,

preconditions (pre-test state), and test oracle (expected test outputs and postconditions). Moreover, there is inadequate regression testing before integration. Modified code (bug fixes and changed design) is not retested prior to integration testing. (Donald Firesmith, 2012)

G. Integration Testing Issues

There is insufficient integration test environments (test beds). Lack of sufficient test beds due to cost or hardware needed elsewhere causes competition for limited resources among test teams. Poor test bed fidelity and quality also will affect the result of testing process. Insufficient software, hardware, and system fidelity to actual system (for example, due to inadequate software simulations/drivers, prototype or wrong version). Next, unavailable components in the integration testing such as not all component are ready at the proper time. (Donald Firesmith, 2012)

H. Specialty Engineering Testing Issues

First of all, inadequate capacity testing. There is little or no testing to determine performance as capacity limits are approached, reached, and exceeded. Second is inadequate reliability testing. There is little or no long duration testing under operational profiles. Next, there is inadequate safety testing, means little or no testing of safeguards for example fail-safe behavior based on safety analysis (mishap cases). There is also little or no penetration testing, security features, or testing of security controls and fail-secure behavior based on security analysis (e.g., attack trees, misuse cases). Furthermore, there is inadequate usability testing for instance there is little or no explicit usability testing of human interfaces for user friendliness, learnability, etc. (Donald Firesmith, 2012)

I. Sos Testing Issues

One of the problem is Inadequate SoS planning. Test planning often does not exist at the SoS level. There are no clear test completion/acceptance criteria. Next, poor or missing SoS requirements is also a challenge faced by testers. Requirements only exist at the system level, leaving no clear and indicate requirements to be verified at the SoS level. Then, SoS may not properly scheduled. SoS testing is not in the system integrated master schedules and depends on uncoordinated system schedules. Besides, poor code quality which the code contains many defects that should have been found during lower-level (unit, integration, and system) testing, which unnecessarily slows SoS testing. (Donald Firesmith, 2012)

J. Regression Testing Issues

Insufficient test automation is one of the issues faced by testers. It is because sufficient regression testing will only be performed if it is relatively quick and painless. The regression tests which is not updated for example regression test suite (test inputs, test drivers, test stubs, oracles) is not updated for new or changed capabilities. Next, only low-level regression tests organized in the system. Only unit tests and some integration tests are rerun. System and/or the SoS tests are not rerun. (Donald Firesmith, 2012)

K. Maintenance Testing Issues

Test documentation not maintained for instance the test plans, test procedures, and other documentation are not maintained as changes occur. Sometimes, the test documentation and software are not under configuration control. The test work products become out of synch with the SW as it evolves over time. The most common problem is the disagreement over finding. It is unclear where the funding should come from (development or sustainment funding). (Donald Firesmith, 2012)

III. Conclusion

Software testing is crucial for any software which is developed by any organization. This is one of the major activities of any software development life cycle as software consists of number of program to perform some different or specific tasks. Software Testing ensures that software which will be deployed to any client it will be error free . Hence, the challenges will will be faced along the process need to be penetrate deeply so that the constructive solutions can be suggested in order to make the testing process more smoothly and the testing results are correct.

References

[1] Donald Firesmith, *Common Testing Problems: Pitfalls to Prevent and Mitigate*, 2012.

[2] Dalal Z. Alrmuny, *Open Problems in Software Test Coverage*, Lecture Notes on Software Engineering, Vol. 2, No. 1, 2014.

[3] Bharti Bhattad and Dr. Abhay Kothari, *Study of Defects, TestCases and Testing Challenges in Website Projects Using Manual And Automated Techniques*, 2014.

[4] TaoXie, *Cooperative Software Testing and Analysis: Advances and Challenges*, 2014.