

Quality Benefit Analysis of Software Automation Test Protocol

S Janardhana Rao¹, S. Ajay Kumar² Dr. R. Satya Prasad³, K. Eswara Rao⁴

¹(Team Leader, A P society for knowledge networks, Hyderabad, Andhra Pradesh, India)

²(S/w developer, Techmahindra, Chennai, India,)

³(Associate professor, Department of CSE, Acharya nagarjuna university, Andhra Pradesh, India)

⁴(Assistant professor, Department of CSE, AITAM, Tekkali, Andhra Pradesh, India)

ABSTRACT: *Software testing is very important to provide software quality and to improve the reliability. As the volume and complexity of software applications tend to increase, user's requirements for more quality software is also rising. The importance of software testing in the software development process is also increasing exponentially, while software testing effort also appears to be increasingly difficult. Hence software testing is still an open research area and effective software testing method is required to improve software quality. In this paper we analyze software automation testing, definition, characteristics and functions are discussed. We propose a method to improve the overall process of software automation testing.*

Keywords: *Software testing, Automation Protocol, software quality, software reliability*

I. Introduction

Software testing is an important stage in software life cycle. Quality testing directly determines the stable operation of software products. It is well known that software testing consumes more time with high labor intensity and easily-introduced artificial error therefore automation software testing method is very important. Automated testing technology in software testing is to be further improved, as work load of software testing is very large (accounting for about 40%-50% of overall development cycle), of which most of the work applies to automation, so the test improvement will bring very significant results to cost, quality and cycle of the whole software projects development work. Generally speaking, it is through the development of the automation tools and the execution of the testing scripts for automation testing to achieve the purpose of software quality evaluation.

In the software development process, first of all it requires that we verify by testing the software developed meet the demand analysis description in function, meet customer requirements of carrying load and the corresponding response time, computation requirements in performance; on the other hand, staff are also required to release software as soon as possible within the budget to reduce spending. Traditional software testing process usually carries out a small number of unit's tests first in the software development process, and then at the end stage of

the software development, focuses on a lot of testing, including functional and performance integration testing and system testing. As more and more complex of software projects, the software testing process brings the following issues for project developers

A. *The project process is difficult to control*

Project management becomes more difficult; a lot of software bugs are often not discovered until later stage of software testing, it is difficult to estimate the cost of expensive in solving the problem, so the project process cannot be controlled. In addition, in the software design process, relevant staff can not accurately understand the software quality state, potentially increasing the difficulty of the project management and development.

B. *Difficult to control project risk*

Project risks cannot be reduced until the development later stage. Usually after testing personnel carry out the system test, it is possible to determine whether the entire development process meet the system functionality, performance and reliability requirements of users.

C. *Project development costs exceed budget*

Throughout the project development cycle, the later errors are found, the higher repair costs, then the problem is not resolved in a timely manner will result in a rapid increase in the project cost.

The rest of the paper is organized as section 2: discuss about analysis of software automation test, section 3: presents improvements to software automation testing process, section 4: concludes the paper.

II. Analysis Of Software Automation Test

A. *The definition of software automation test*

Currently, the conventional definition of software automation test is "the use of an automated testing tool to verify the needs of a variety of software testing, which includes the management and implementation of testing activities." Usually, we carried out automated testing refers to an automated testing has used the formal process of manual testing process.

B. *Why should the introduction of automated software testing*

Typically, software testing is accompanied with many repetitive, non-intellectual and non-creative operations, computer can replace manual to accomplish this task, so that programmers are released from the complex and repetitive work and turn to a deeper level of project testing. Software automation testing is based on the existence of manual testing, mainly through the corresponding software testing tools, scripts, etc. to implement, has good maneuverability, repeatability and high efficiency characteristics. Automated testing is of high testing efficiency, low cost, it can automatically convert test with strong repeatability from manual to computer; very suitable for fast regression testing, change the situation that manual testing is difficult to make rapid assessment to the new iterative version; avoid common errors in manual testing, and test time-consuming boundary value can rapidly and accurately implement the concurrent operation of the multi-user that manual testing is difficult to achieve, automatic test is easy to complete. Therefore, the implementation of test automation is a development direction software testing industry. Mastering advanced technology in this field has a huge boost for a software company's core competitiveness and future development.

C. Common tools in software automation testing

There are many tools for software automation testing, some more commonly used tools are introduced in details given by the following the table below table 1.

Table 1: Tools in software automation testing

| | |
|---------------------------------|---|
| Unit testing | Java testing tool: Junit |
| Functional testing | Mercury Interactive Company WinRunner, QuickTest Pro, IBM Rational Company's Rational Robot, Rational Xde Tester, Compuware Company's QARun, RadView Company's WebFT, Empirix Company's eTester, etc. |
| Stress testing | Mercury Interactive Company's Winload, Compuware Company's QALoad |
| Load testing | Mercury Interactive Company's LoadRunner |
| Web testing tool | Empirix Company's eTest Suite |
| Web systematic testing tool | Microsoft Company's Web Application Stress Tool |
| Regression testing | Rational Company's Team Test |
| Performance testing | Mercury Interactive Company's Loadrunner, Compuware Company's QALoad, Microsoft Company's web Application Stress Tool, etc. |
| Testing process management tool | IBM Rational Company's Rational Testmanager, Mercury Interactive Company's TestDirector |
| Defect tracking tool | Compuware Company's senior changes and defect management tool: TrackRecord |

| | |
|-------------------|---|
| White box testing | IBM Rational Company's Rational Purify pluS, Compuware Company's DevPartner Studio Professional Edition |
| Black box testing | QACenter, SQATeamtest, Rational Visual Test |

D. Design test scripts

Script is a set of code that is executed by test tools, it can be generated by recording the operation of the test, and then do modifications, thus reducing the workload of scripting. Of course, it can directly use scripting language to write scripts. Scripting technologies can be divided into linear script, structured script, sharing scripts, data-driven scripts and keyword driven script. The principles of editing script is try to cover most test items, with strong compatibility, making a new project testing can be easily inherited and modify the original script, to reduce the subsequent workload.

III. Improvements to Software Automated Testing Process

A. Determine the feasibility of products automated testing

Automated testing can improve test efficiency, but for short period, tight scheduled projects, automated testing should not be used. Preparatory work for the implementation of automated testing is very large, to apply enterprise-level automated testing framework into a project needs to assess its suitability, it must not be blindly applied to a test project, in particular, not suitable for project with short cycle, because it is very likely to need a lot of test preparation and implementation of the framework, which will lead to insolent. For example, for a unit test automation, and develop hundreds of lines of test a code line with only 10 rows of function, regardless the maintenance costs of developed automated testing, at least you can have an intuitive understanding: shift from manual testing to automated testing has costs, and this investment is far higher than cost of manual testing, it must firstly to determine whether automated testing of a software project is appropriate.

B. Make risk assessment for software project testing

Financial risk, although some projects have automated testing conditions, organizational restructuring cost estimates after the introduction of automated testing is necessary. Entry mode risks must be considered with automated testing and manual testing combination, design appropriate ratio to ensure smooth testing. At the start, 80% tests can be set to be manual testing, when these goals are achieved, then increase the automated testing rates. Time estimates, after finishing the assessment of above several indicators, it is necessary to estimate the time period to

implement test automation in order to prevent unnecessary waste of time, to reduce the unwarranted consumption in personnel, capital and resources. Although after the test automation goes in the right track, it will play a multiplier effect, the early investment is huge, it is required to fully consider all the factors, clarify implementation plan and strictly according to plan, in order to minimize risk.

Workflow change risk. The test team or even the whole development organization implement test automation, more or less due to adapt to the work process of test tools, bringing the corresponding changes in team's test process, development process. And, if the change is unreasonable, it will cause much complain mood of team members, so it should minimize the changes, and to overcome the difficulties that may exist in changes.

C. Develop careful test plan

Test plan includes the establishment of test environment, determining the test strategy, testing design process control and fault-tolerant processing. According to characteristics of developed software project, develop careful plan, design various steps of test. Due to automated test has relatively high requirements on software design quality, it cannot do anything to emergencies and issues cannot be solved by the software, so solutions should be determined in advance for test abnormality, the problem can be resolved quickly, reduce unnecessary overhead. Automation test strategy refers to make sure which tests execute automated testing and when to adopt the automation testing. One of the misunderstandings for people is that they should realize the probability of the automated testing to the most degree, that is, the earlier the better; usually people believe that the higher the automated testing is, the higher the utilization rate of the automated testing instrument is and the greater we can obtain the investment rewards. In fact, the profits we obtain from test automation tool should be reflected in the test on quality, not quantity, and what tests should be chosen to implement the automation and how to develop executive test scripts are more important than how many tests has been automated. How do people select the best way for the execution of automated test? Usually it can be divided into three steps, that is, the extraction of testing suitable for automation, the evaluation of the time consumption for each automation testing and the verification of the order for the automation testing priorities according to the test goal. Firstly, make the tablet to extract automation testing project with the principle of selecting the testing item which can get the biggest investment rewards, namely, to shorten the time period most, to reduce the risk most and to improve the test accuracy most. Then, evaluate the time consumption for the automation test. Presently there is not a simple mathematical model to test the proportion between the time consumptions for the automation testing and manual testing; however, according to the estimation of the test expert, the time of an

automation testing for software is thrice to ten times as much as a manual testing, for complex testing, or even longer. Therefore, if a test suite need 100 hours to test, for automation test it will takes 300 to 1000 hours or more time, and certainly we have to admit that any estimation is only a surmise. Although we should make the judgment on the basis of enterprise testers' actual testing skills, test software actual characteristics and the actual testing tool-use complexity, one thing is uncountable that the time consumption by the first implementation automated test is longer than the time by the familiarity with tools and the testing, so in evaluation of the time consumption by automation testing we must take it into consideration. My basic principle is to select the test with the longer time consumption to execute automation test firstly.

D. The combination of two programs: manual testing and automated testing

Test automation can bring very significant benefits, but we cannot completely rely on automated testing, it is only a part of testing, is a complement to manual testing. Automated testing is no substitute for manual testing; both have their own characteristics, with different test object and test range. According to report, automated testing can only detect 15% of defects, and manual testing can detect 85%. Before automated testing, we should first establish a concept of software automation testing. Unit testing, integration testing, system load testing, performance testing, stability testing, reliability testing are suitable for automated testing. Instable software testing, software with short development cycle, one-time software is not suitable for automated testing. We can choose according to actual situation. In most cases, manual testing and automated testing should be combined, to complete the testing tasks in the most effective way. During functional testing, the precise meaning of automated testing tool is regression testing tool, at that time the tool cannot find more new problems, but can guarantee accuracy and objectivity of parts that have been tested

E. Summarizes the process of software automated testing

To record the entire testing process, summarize experiences and lessons learned in the process, generate test reports, find ideas and methods for resolving similar problems, so in the future testing similar software projects can reuse the previous results achieved, and if able to do so, on the basis of the test, design related test template, if the future has software projects with similar structure, we can follow this model to test, thereby greatly improve the efficiency of the test.

IV. Conclusion

In short, domestic development of automation test in software project is not very mature, has not really formed a whole set of reasonable and effective norms. In practice, we have to combine features of traditional manual testing

and automated testing, play their strengths, so that automated testing strategies and tools work as a weapon in the hands of testers, release them from boring and repetitive work, focusing time and energy on complex work requiring intelligent judgment and other new test cases. In addition, we must avoid equate automated testing and testers together, not asking too much for automated testing. We shall set a proper automated testing perception, have a clear understanding that manual testing is a powerful complement to automated testing and not replace the status of the tester. Any single technology or operation advances cannot independently ensure a large scale of improvements to software development efficiency, stability and maintainability in a short time. Automated testing is also an accumulation of experience, gradual process, software testers cannot be expected to automate all the tests in a short time. Successful automated testing requires developing appropriate automated test program, and reasonable automated testing is the first step in the success implementation of the automated testing strategy. Only by fully taking into account the risk of implementation of automated testing, resources and objectives, it is possible to develop automated testing strategies for your own, and

ultimately improve test efficiency and reduce cost of test. Each enterprises with testing department or engaged in testing business, should learn overseas and domestic advanced testing experience, refer to popular industry standard, find their own team's testing methods and models to create greater social value.

References

- [1] Yang Xina, Yuan Congling. "The Java-based Web Application Performance Testing", 2006.
- [2] Chen Hua. "The Accuracy of Testing Process", 2005.
- [3] Chen Yanzhuo. Metric Testing Tools - Three Indicators for Assessment of Automated Performance [4] Testing Tools. software world, 2005.
- [5] Software Automation Testing Technology. IBM Rational Technical White Paper.
- [6] Liu Sheng. Design and Practice of Software Automation Testing Framework. Posts & Telecom Press.
- [7] Li Yongzhong. Analysis of Effective use of Automated Testing in Software testing, 2006.
- [8] Richard Bender. Requirement-based Test is the Essence of SoftwareTesting, 2010.