

Test Process Model and Management in Software Engineering

Hongyu Li, Jinsheng Zhang and Donghu Gu

Yunnan Technology and Business University, Kunming, Yunnan 6517001, China

15887047588@163.com

Keywords: Software Engineering; Software Testing; Test Process; Test Management

Abstract. Computer software testing is an important part of software engineering and an important means for software developers and users to verify that software products are qualified. Under the guidance of software engineering concept, based on the rich experience of software testing project and its own good software testing technology, this paper fully considers the particularity of model software testing process management, and draws on the management theory of related disciplines to develop strict and quantitative Software testing process management tools enable effective management and control of the testing process. This paper proposes a more general software test process model from the perspective of software engineering as a whole. This model can help test managers to determine the current test status in a changing software development environment to some extent, find out the existing problems in time, and take corresponding measures according to the problems, so as to promote the test process to meet the development plan and quality. The purpose of the standard. This paper gives a case based on the author's actual experience and analyzes it. The analysis results show the rationality of the above model.

Introduction

The data shows that in a software development process, testing accounts for 40% - 60% of the entire work. Therefore, how to greatly improve the quality of testing at a relatively low cost plays a very important role in the final quality of the software. However, at present, there are still major manual methods to deal with various complicated test management tasks.^[1] The software requirements, configuration management, and quality control cannot meet the requirements. There is also a lack of management tools that can help automation and efficiency. This brings the following obstacles to the smooth progress of software testing:

- (1) The software test needs analysis is insufficient, and the change is uncontrollable;
- (2) The test data lacks statistical information, and there is no inheritance between projects, and it is impossible to achieve rapid retrieval of information;
- (3) The test case change and upgrade control are not perfect, and the traceability is not strong;
- (4) The software defects found are lack of effective management, and an effective software failure model cannot be established;
- (5) The regression test process is not standardized.

Therefore, it should be considered from the whole process of the software testing process, according to the current domestic mainstream testing process to establish a set of management tools that meet the testing requirements, to provide technical support for the process management of the software testing project.^[2] This allows the test process to share test results, greatly improving the test efficiency of the entire development life cycle.

Mercury TestDirector for Quality Center is another popular test process management tool in the industry that deploys high-quality applications quickly and efficiently. It is a single web-based application for managing all important aspects of the testing process. The approach is to provide a continuous, repeatable process for collecting requirements, planning and scheduling tests, analyzing results, and managing defects and failures.

In addition, the software testing process management tools of well-known foreign companies include Segue test management system SilkCentral Test Manager, T-Plan Professional, etc. These foreign products are commercial and are specifically for the testing process in general software development. developing.

In China, the concept and implementation of software engineering has received more and more attention in recent years. People have realized that software testing is an important part of software engineering.^[3] In order to effectively control the software testing process, it is necessary to have corresponding management methods, technical means and efficient tools to rationally plan the entire testing process. Supervise various activities and record, process and process various related information. If there is no professional test process management control tool, the test management is lack of science and efficiency, and the research and design development of such tools starts late in China, especially for large software, there is still no suitable test process. Management tools. In particular, the domestic software evaluation laboratory has its own professional testing process when conducting independent third-party software evaluation. The management method is quite different from the test management of foreign commercial software. There is currently no complete solution.

Software Test Process Model and Analysis

Software Test Process Model

The test is divided into four stages: planning, design, execution, and verification. This model can be used for unit testing, integration testing, and system testing. It can also be described for a single test case.

Analysis of each stage of the software test process model

(1) Test plan: Software testing requires an overall plan for pre-analysis. However, system testing, integration testing, and unit testing for a specific project also require separate planning. The test plan is primarily to pre-plan time and resources and produce a guided test plan document. If the software verification finds problems affecting the progress of the software test and the progress of the software release, the plan for the software test may be adjusted.

(2) Test design: The test design needs to be performed strictly according to the design document. From the V development model, we can see that the "Requirements Specification" is the basis for the system analysis use case design. The "Summary Design Specification" is the basis for the integration test design, and the "Detailed Design Specification"

It is the basis for unit test design.

(3) Test execution: Test according to the use cases in the test design. In the test, you may find that the use cases in the document are not sufficient. At this time, some new test cases need to be added to the other files. After the test is completed, the test cases are merged and supplemented.

(4) Test verification: The problems found in the software test need to be modified by the developer, while the software testers need to the developer.

(5) Modify to verify: If the verification does not pass, the developer will need to continue to modify.

Problems to be aware of during testing

System test: The system test mainly carries out the black box test. Usually system testing is done by independent testers. There are two issues to be aware of for system testing.

(1) Operation mode problem. Although the current object-oriented development concept has been deeply rooted in the hearts of the people, in the design of the software, try to work towards the high cohesion of the module and the low coupling direction. However, we still can't avoid operating restrictions on some software features. Problems still occur under certain operational sequences. This limitation of the order of operations can be named the mode of operation. It may involve multiple functions of the software. Stress testing in various situations.^[5] Abnormal conditions, the test under the boundary condition is the focus of the system test, but in the system test, it is particularly necessary to focus on the combination with the stress test. For example, when a TCP connection-based program makes a TCP connection, it is an anomaly to issue a SYN connection but the connection is not completed. However, if there are tens of thousands of SYN connections failing in a short period of time, it is a denial of service attack.

(2) Integration testing: Integration testing can often be seen as a situation between black box testing and white box testing.^[6] Usually the test is done by an independent tester. The main thing

about integration testing is to observe the flow of data between modules. In the integration test, a large number of pile modules and drive modules may need to be completed to simulate the data flow. Therefore, it is very important to accurately design the data drive and the pile module. Strict monitoring is also required.

(3) Unit test: The unit test is a white box test. Usually done by the code completion. Unit testing typically requires 100% statement coverage. For functions that do not have too many judgment conditions, a 100% conditional coverage can be required.

Application Examples of the Test Management Process

According to the software test management process, we have applied in many projects. As the testers continue to be familiar with and master the test management process, more and more problems are found in the test, and more and more comprehensive.^[5]

The following three representative system development test data are selected for description. The three systems are the library management system, the student status management system and the advertising company management system. Software developers have the same level of familiarity with software tools, and these three softwares are also based.

Table 1 Test Management Project

Project code	Scale	Test management
library	5kloc	Not managed
Advertising	3kloc	Managed
Student status	4kloc	Managed

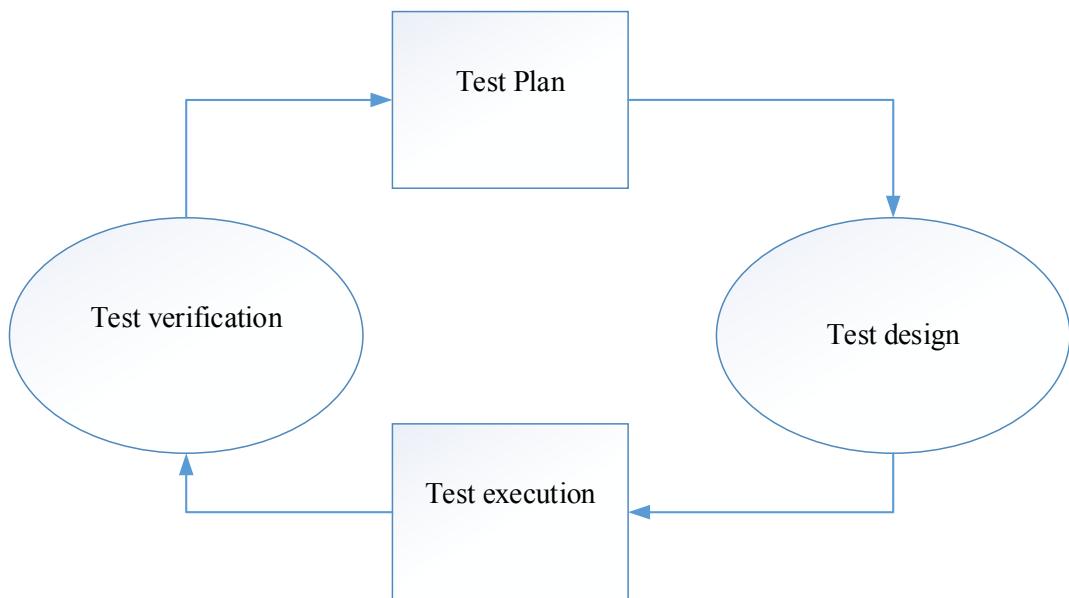


Figure. 1 Software test process model

The same nature and development difficulty, so the development data is comparable. Among them, the library project did not manage the testing process, the advertising project carried out simple management of the test, and the student management project adopted the mature management mode mentioned in this article. The collected data is divided into 2 types, which are found in the test and found after the test. ^[6]The data found in the test can directly indicate the quality of the test. However, the problem data found after the software is delivered can only be used as a reference. This is because the data found after the test is a direct standard for measuring the quality of the software, and the improvement of the quality of the software is related to various factors.

Table 2 Test methods used

Project	Test document	Test Plan	Problem tracking	Problem modification verification
Library	×	×	×	×
Advertising	√	×	×	√
Student	√	√	√	√

Table 3 Test Management Data

Project	Code size	Unit test	Integration Testing	System test	Post-delivery error
Library	5kloc	No	No	No	24.5 error / loc
Advertising	3kloc	5.0error/loc	5.3error/loc	6.2error/loc	12error/loc
Student	4kloc	14.0error/loc	4.3error/loc	4.0error/loc	3.2error/loc

As can be seen from Table 3, after the standardized test management, the problems found during the test phase are significantly improved, after the software is delivered. The error is also significantly reduced.

Software Test Process Management

Software test process management overview

Software test management is to monitor the entire software testing process through certain management methods and tools to improve the quality of software products. Software test management is the rationale for 3P: Processes test process, Products test work product, People tester. The testing process includes technical processes, management processes, and support processes.

The management of the testing process is mainly to measure and analyze the effectiveness and efficiency of the software testing process, and to continuously improve the metric-based software testing process. Test work products include test plans, test instructions, test cases, and test reports

Etc. The management of test work products is mainly to measure and analyze test software products, collect data needed for quality analysis and product release decision, check and review software test work products.^[5] The management of testers includes the establishment of an effective software testing organization to enable testers to achieve the work objectives specified in the test plan, and to collect and analyze software performance data.^[9]

The purpose of software testing process management is to control and manage the entire testing process of software products, improve the software development capabilities of software development organizations, especially the management level of software product testing, instill and strengthen the management philosophy of enterprises, and ensure software development organizations. Develop the quality of products and further enhance their market competitiveness. Currently popular test management tools

There are many kinds of software test management tools currently used. Among them, the mainstream software test management tools on the market are: TestCenter, TestDirector, TestManager. The following is a detailed comparison of the performance of each software test management tool in the form of a table:

Table 4 Test Management Tool Comparison Table

Tool Name	Overview	Benefit	Disadvantages
TestCenter	Full-featured domestic test management tool	1. Support test management for business flow. 2. Provides a standardized test process to facilitate user quality management of the test.	1. Cannot integrate with mainstream test tools. 2. The function needs to be further improved.
TestDirector	Former MI company's test management tools	1. And the Rational Test Series' name test management tool is powerful. 2. Web interface. 3. There are test case execution tracking functions. 4. Have flexible defects system. 5. Tight integration with its own defect management tools.	1. There is a limit on the number of simultaneous online users in each project library. 2. There may be some instability, but there is no problem with the basic functions.
TestManager	Recommended in Rational test solutions Test case management tool.	1. Powerful. 2. Management of folder form, can be used for test cases Unlimited rating. 3. Can be combined with Rational's testing tool robot, functional. 4. There are functions for test case execution, but must be A corresponding manual or automated script.	1. Localization support is not good. The Chinese character display is too small. 2. Test cases are not very stable when they are many. Sometimes it causes loss of test cases. 3. The client must be installed before it can be used. It is not convenient to communicate with developers. 4. The test case is displayed in a single form.

Conclusions

This paper studies the related technologies of software test management, analyzes the current status of the more popular test management tools at home and abroad, and can customize the process configuration by custom software test flow software. You can let users define new processes themselves, copy existing processes, sort processes, and define fields in the process. During the testing process, the configuration management software of each work product performs identification identification and version management of the work products generated during the testing process, manages the change of the work products, and periodically performs configuration auditing. Whether it is the control of the test process or the evaluation of the test results, it needs to be measured. Through the analysis and processing of the test data, it can be done "in mind."

References

- [1] Lubke D, Van Lessen T. Modeling Test Cases in BPMN for Behavior-Driven Development[J]. IEEE Software, 2016, 33(5):15-21.
- [2] Lohrmann M, Reichert M. Effective application of process improvement patterns to business processes[J]. Software & Systems Modeling, 2016, 15(2):353-375.
- [3] Hofmair M W, Melik-Merkumians M and B?Ck M, et al. Patching process optimization in an agent-controlled timber mill[J]. Journal of Intelligent Manufacturing, 2017, 28(1):69-84.
- [4] Mohan K K, Srividya A, Verma A K. Prototype dependability model in software: an application using BOCR models[J]. International Journal of System Assurance Engineering and

- Management, 2016, 7(2):167-182.
- [5] Khan A A, Keung J W and Amin F E, et al. SPIIMM: Toward a Model for Software Process Improvement Implementation and Management in Global Software Development[J]. IEEE Access, 2017, PP(99):1-1.
 - [6] Fricker S A, Schneider K and Fotrousi F, et al. Workshop videos for requirements communication[J]. Requirements Engineering, 2015, 21(4):521-552.
 - [7] Galli J S, Vairetti C and Carlos Rodríguez, et al. Analysis and improvement of business process models using spreadsheets[J]. Information Systems, 2016, 57(C):1-19.
 - [8] Farahani E D and Habibi J. Configuration Management Model in Evolutionary Software Product Line[J]. International Journal of Software Engineering and Knowledge Engineering, 2016, 26(03):433-455.
 - [9] Alotaibi, Youseef. Business process modelling challenges and solutions: a literature review[J]. Journal of Intelligent Manufacturing, 2016, 27(4):701-723.