Practical Approach to Promote System Test Automation in a Large-Scale Organization

Hideto Ogasawara[*1] Takashi Nakano[*2]

[*1]Chiba Institute of Technology [*2]Toshiba Corporation

In addition to the development of new software for installation in various products, the creation of derivation software utilizing existing software accumulated as development assets has been increasing. This approach, referred to as derivation development, is effective for the refurbishment of existing software. However, a large number of test cases need to be examined in order to assure the quality of such software for a new product, making it difficult to reduce testing worker hours and shorten testing periods. The Toshiba Group has been implementing a broad array of measures to rectify this situation through the development and introduction of system test automation techniques, as follows: (1) preparation of guidelines for the introduction and promotion of test automation, (2) development of test automation processes, (3) establishment of a test automation organization utilizing overseas resources, (4) maintenance of a unified management environment for test specifications and test scripts, and (5) development of a technique to generate test scripts with easy maintenance. We have conducted simulation experiments applying these techniques to a large-scale social infrastructure system and confirmed that the techniques make it possible to achieve a 30% reduction in the system testing costs of subsequent systems and a 40% reduction in initial introduction costs.

Keywords and Phrases: Test Automation Organization, Overseas Resources, Cost Reduction, Test Automation Process

1. Introduction

In recent years, the functions and values of products have been provided by systems and digital information processing, and the development scale of software has dramatically increased. In the field of software development, the importance of debugging and testing is increasing more to prevent problems from occurring after product shipment. The time and worker-hours of the interface test and the system test in software development have reached about 40% of the whole development.

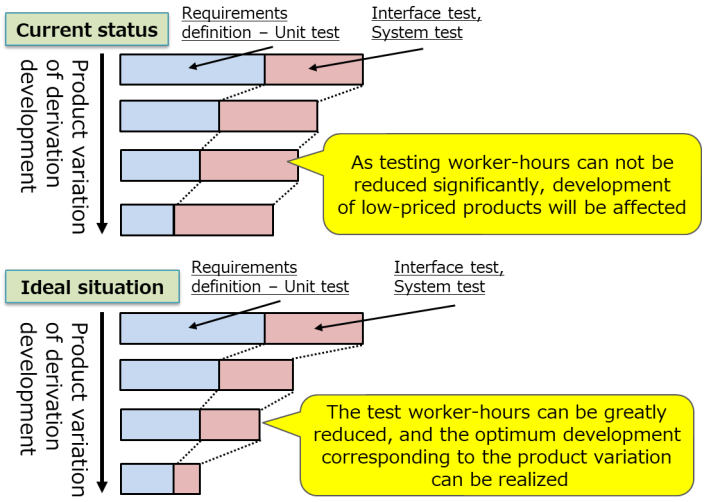

Figure 1: Comparison of ratios of testing cost in current and ideal derivation development

With the evolution and diversification of information technology (IT) and the acceleration of the speed of business and technological innovation, many test cases must be conducted in a short period of time.

In software development of products, there are many derived derivatives which remodel existing software rather than new development. Figure 1 shows the ratio of test worker-hours in derivation development. Derivation development tends to decrease the cost of requirements analysis, design, implementation and test worker-hours from requirements definition to unit test as compared with new

development. However, in the case of software development, changes to certain functions often affect different functions. In addition, it is difficult to accurately grasp the range of influence of change in advance, so in many cases, quality is ensured by carrying out many tests including functions not directly related to change in the testing process.

In software development of products, there are many derived derivatives which remodel existing software rather than new development. Figure 1 shows the ratio of test worker-hours in derivation development. Derivation development tends to decrease the cost of requirements analysis, design, implementation and test worker-hours from requirements definition to unit test as compared with new development. However, in the case of software development, changes to certain functions often affect different functions. In addition, it is difficult to accurately grasp the range of influence of change in advance, so in many cases, quality is ensured by carrying out many tests including functions not directly related to change in the testing process. For this reason, the time required for the test process could not be greatly reduced, and the ratio of the test process to the total development tends to increase. The smaller the scale of addition or change of specification, the more testing worker hours of the test process leads to cost increase of the whole of development.

The problems associated with an increase in testing worker hours are as follows:

(1) Increase of worker hours by repeated test

(2) Increase of backtracking worker hours due to omission of tests

Therefore, Toshiba group has developed a system test automation technology to verify whether the entire system meets the planned function in order to solve these problems. This paper describes the activities of system test automation conducted by Toshiba group and its application effect.

2. Problems and measures for system test automation

The process of system test automation is shown in Figure 2. In Step 1, system test execution is automated by creating test scripts based on existing test specifications in order to solve the problem (1) mentioned in section 1. After that, it applies to multiple product series, and drastically reduces testing worker hours of system test. In step 2, we develop a test specification and test script automatic generation technology in order to solve the problem (2) mentioned in section 1. By applying these technologies, it is possible to develop products satisfying the requirements of QCD (Quality, Cos, Delivery).

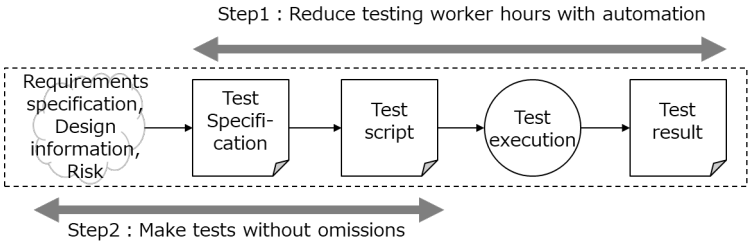

Figure 2: Two steps to promote system test automation

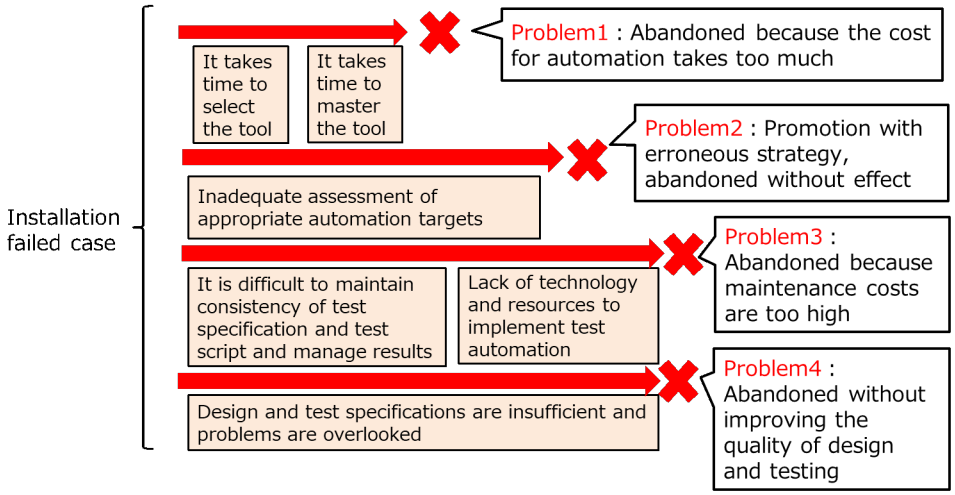

Figure 3: Problems in system test automation

On the other hand, Figure 3 shows the challenges faced by many organizations when introducing and fixing system test automation. Toshiba Group also introduced a test automation tool in multiple development departments and tried it. However, the problems listed in Figure 3 became obvious, and there were scattered circumstances in which system test automation was not continued.

Step 1 has Problem 1, 2, and 3. To summarize, "It takes too much cost to introduce and maintain system test automation, so it will not be effective". Step 2 has Problem 4, and it became clear that "Because the current test specification is insufficient, it is not possible to efficiently and effectively create test scripts". Taking these as an impediment factor in each step, we developed the measures in Table 1.

Table 1: Measures to overcome hindering

Inhibitory factor	Measures	Measure details
Step 1: Installation and maintenance costs take too much, so no effect (Problem 1, 2, 3)	(1) Overcoming introduction barriers	Tool selection, identification of automated targets, planning of automation and support of execution management
	(2) Organization construction	Clarification of roles and processes of developers, support personnel, test developers
	(3) Improvement of the environment	Establish an environment in which test specifications, test scripts, and test execution results can be associated and managed
	(4) Development of easily maintainable test script	Application of test script development technology
Step 2: Because there is a missing test specification, test script can not be created efficiently and effectively (Problem 4)	(5) Diagnosis and improvement for quality improvement of test design	Test process diagnosis / improvement technology
	(6) Efficiently create test script	Development of test script automatic generation technology

3. Efforts to disseminate and deploy system test automation technology

We tried automating system test for software development of large-scale social infrastructure system. First, we implemented the measures (1) to (4). Initiatives of measures (1) to (4) are shown below.

3.1 Measures (1): Overcoming introduction barriers

In order to overcome the introduction barriers concerning problem 1 and problem 2 in Figure 3, we selected tools, determined the scope of automation, and supported planning and execution management of automation. Based on these experiences, we also created a guide to introduce and promote system test automation.

The system test automation introduction and promotion guide shows the steps of the strategy for introducing system test automation in organizations and projects. In this guide, we provide reference information such as what to do for each step, what should not be done.

Figure 4 shows the steps to develop system test automation strategy.

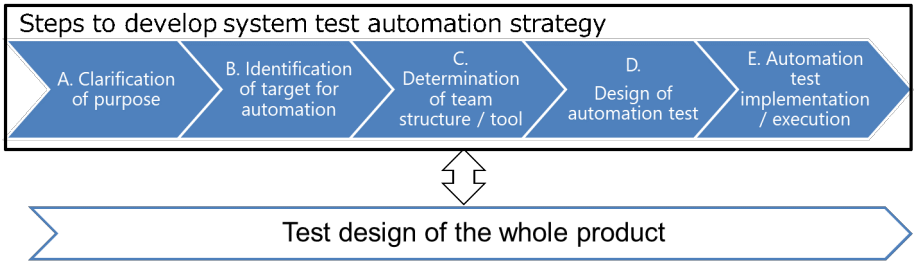

Figure 4: Steps to develop test automation strategy

It is a point to clarify the purpose of automation (for example, improving the efficiency of system test) without aiming at system test automation itself, and to identify the object to be automated without trying to automate everything. Therefore, merit and disadvantage of system test automation, suitable or not suitable of system test automation are classified from three viewpoints, project features, software features, and test features.

Also, in order to prevent the activity from stopping when there was no person in charge of system test automation, we described that the system test automation activity should be incorporated into the development process of the organization. Furthermore, the necessity to clarify the roles of the design engineer, developer, test engineer, and test automation engineer, etc. was also stated.

3.2 Measures (2): Improvement of organization

Based on the description of the system test automation introduction and promotion guide, we developed an organization that automates the system test. Figure 5 shows the roles of software developers, test developers, and support engineer. At the base of software development in Vietnam, we set up a test center consisting of about twenty people. In order to increase the application of system test automation within Toshiba Group, we integrated system test automation technology to this test center. Members of the test center are assigned to the project as test automation developers and implement system test automation. Utilizing the resources of Vietnam also led to cost savings in test script development and maintenance. In addition, by clarifying the role sharing, we aimed to improve the quality of the system testing by testing experts.

Toshiba's Software Engineering Center provided support for designing the whole activities and holding regular meetings.

We informed the members of the test center the purpose and goal of the system test automation in the development project and the future image of promoting the system test automation in the company. And we shared that system test automation is an important activity for Toshiba.

We held two regular meetings of software developers and support engineers, test developers and support engineers every week. The main confirmation / examination items at the regular meeting are as follows.

(1) Confirmation of test script development status

(2) Identification of system test automatic target

(3) Preempt risk for test script development and execution

(4) Confirmation of automated system test execution result

(5) Confirming the symptom that seems to be a problem

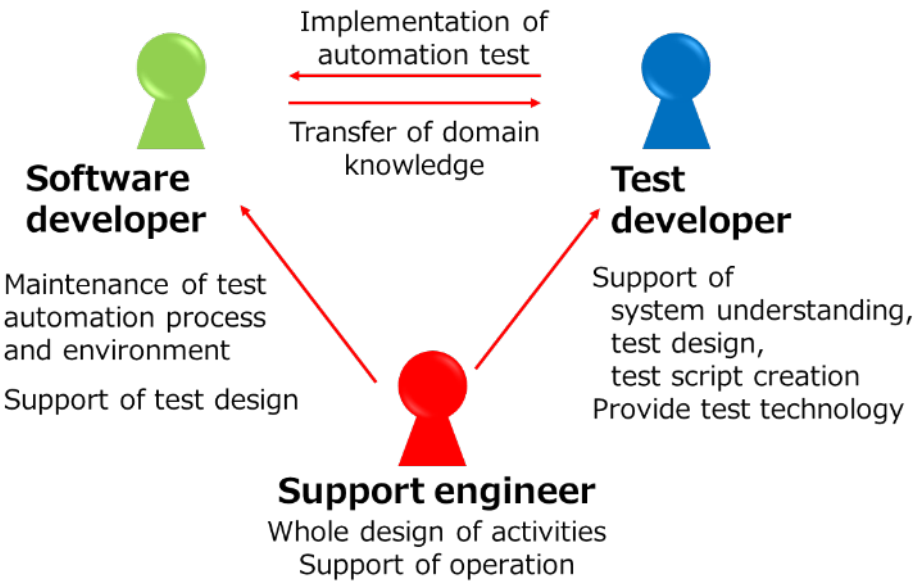

Figure 5: Roles of workers in charge of each section

3.3 Measures (3): Improvement of the environment

Regarding problem 3 in Figure 3, the environment for system test automation was improved. In order to practice system test automation, it is necessary to manage stable system tests. If correspondence between the test specification and the test script is not established, it becomes difficult to manage the system test result, and it takes time to modify the existing test script at the time of specification change or derivation development. In the worst case, reworking may occur. Toshiba Group widely uses the test management tool TETRAPLUS which accumulates and manages test specifications and test results. By using this tool, it is possible to confirm and analyze the state of implementation of the test in various views. In addition, a mechanism for reusing test assets (test specifications / results) is provided.

In order to maintain the consistency of the test script and the test specification, a system test automation environment centered on the test management tool TETRAPLUS shown in Figure 6 was developed in cooperation with the test center.

Test scripts created with test automation tools on the market or free software and test specifications registered in TETRAPLUS were drawn and the results of the tests automatically executed using the developed linkage tool were saved in TETRAPLUS. As a result, test scripts that require modification from the test specification can be traced at the time of specification change, etc., and unified management of the test script is realized. In addition, by registering the screen capture acquired at the time of the test with the test automation tool on TETRAPLUS, evidence management of the system test result can be performed. Furthermore, with regard to the tests that it is difficult to automatically decide whether or not to perform the system test, we devised so that the tester could see the screen captured on TETRAPLUS and decide whether to accept the system test.

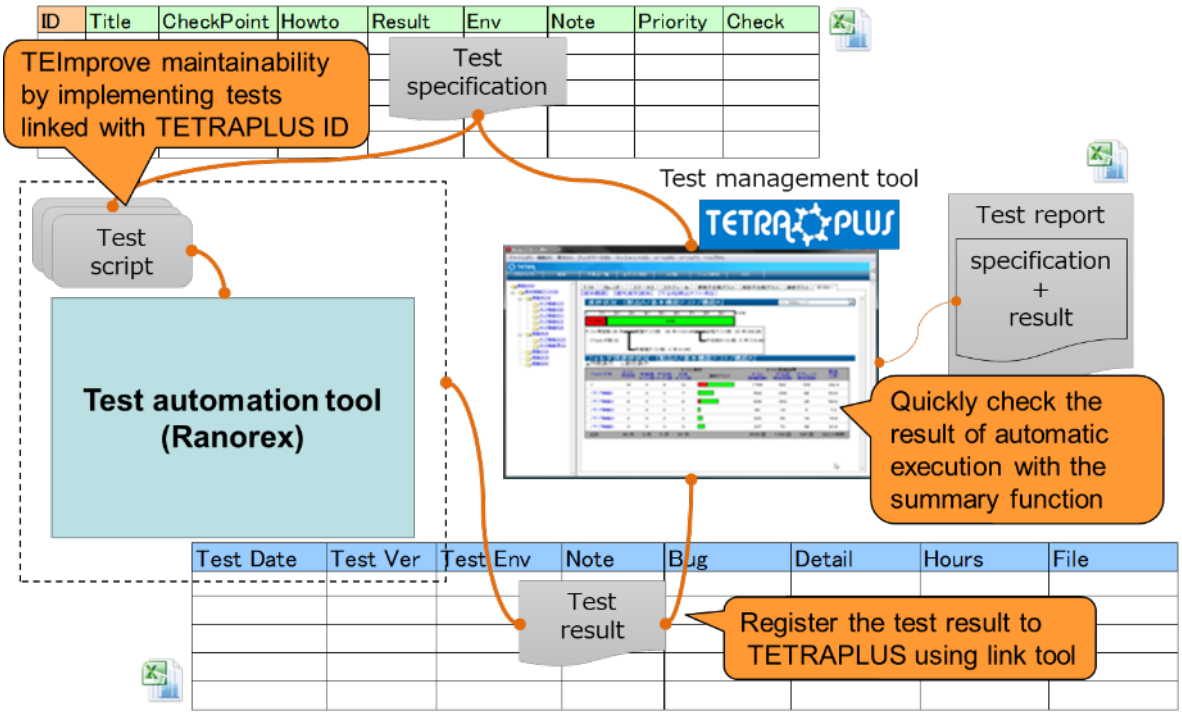

Figure 6: Established system test automation environment

3.4 Measures (4): Develop test scripts that are easy to maintain

For efficiently executing the system test, Preprocessing, postprocessing, common processing, etc. are cut out, and the structure of the data driven script which is easy to cope with the change of the system test environment and system test data was adopted.

4. Example of application effect

We show examples of the effect of system test automation on development of large-scale social infrastructure system.

We identified the scope of system test automation considering suitable or not suitable, reuse of automation, and judged that it is effective if 50% of the existing test specification could be automated, and we developed a test script. In this application example, 30% of system test automation is completed, and manual test cost can be reduced by almost 30%. This time it costed the initial introduction of system test automation, but it is thought that the cost of the system test corresponding to this 30% can be reduced after the next model.

Figure 7 shows the cost transition prediction of the system test. Utilizing the test center, the initial introduction cost was reduced by 40% compared with the case where the system test automation is advanced only by its own department (the initial introduction cost in the case where system test automation is realized only in its own department is a predicted value). As a result, it is possible to carry out the test including the system test automation part at almost the same cost compared with the previous model, and it can be seen that the initial introduction cost of automation can be almost recovered by this development. In cost forecasting, by further promoting system test automation, the cost of system testing after the next model can be further reduced, and great cost reduction can be expected from now on.

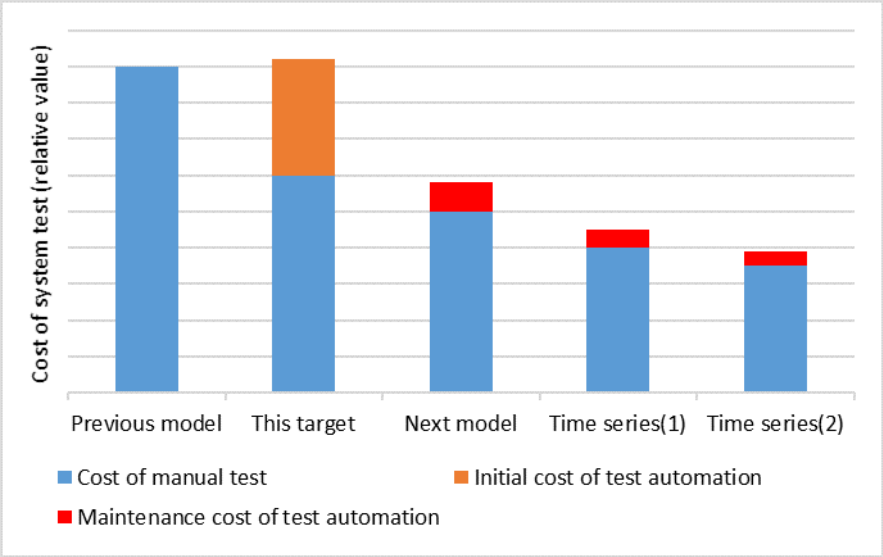

Figure 7: Changes in estimated system testing costs

5. Conclusion

We were able to confirm the cost reduction effect by implementing efforts to disseminate and deploy system test automation against the problem of increase in testing worker hours. Automated system test utilizing Vietnam's test center is spreading within Toshiba Group.

Currently, we are promoting measures (5) and (6) in Table 1, and we will promote system test automation in terms of quality and promote the spread and deployment of system test automation in the future.

References

Ogasawara, H., Sasaki, M., Nakano, T. and Aman, H. (2014). Application of the test select method using mathematical programming model. *EurpoSPI2014*. 101-108.

Graham, D. (2014). Test Automation Problems - Survey results.