A Process Model for Test Automation

Juhani Snellman

Helsinki 23rd November 2004

Seminar presentation

UNIVERSITY OF HELSINKI
Department of Computer Science

Tiedekunta/Osasto — Fakultet/Sektion — Faculty
Faculty of Natural Sciences

Laitos — Institution — Department
Department of Computer Science

Tekijä — Författare — Author
Juhani Snellman

Työn nimi — Arbetets titel — Title

A Process Model for Test Automation

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Seminar presentation	23rd November 2004	14 pages

Tiivistelmä — Referat — Abstract

Software quality has been an issue as long as there has been software development. Hardly anyone wishes to produce software which doesn't meet it's intended purpose. Software testing is the activity, which is aimed at asserting the software conformance to it's specification and that it actually does what it's supposed to do. Although software testing is not by any means new technique for improving software quality, it's still rarely used to it's full extent. In fact, it's often used inefficiently in terms of resources, cost and therefor the quality of the results given by testing is as bad as the quality of work put into testing.

Test automation has been wished often to be the panacea for software quality problems. However, it can be seen, that test automation must be controllable and implemented with same rigor as any quality paradigm. Test automation can improve software quality, but it doesn't' necessarily or directly do so. This paper addresses the problems test automation by presenting a process model for test automation projects. The test automation process model is used both for introducing test automation into a software organization and for improving existing automation in an organization.

ACM Computing Classification System (CCS):
A.1 [Introductory and Survey],
D.2.9 [software process models, software quality assurance]
K.2.5 [testing, test automation]

Avainsanat — Nyckelord — Keywords
test automation, automating testing, process model

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

Software quality has been an issue as long as software has been developed. Errors in the software have caused financial deficits, physical injuries and even losses of human life throughout the history of the computerized society. Approaches to achieve software correctness can be divided into two categories: active and passive. Active approach involves formal proof of software correctness, which is in a typical case infeasible to do. Software testing and debugging belong to the passive approach, since they can only show there are defects present in the software, but not that at some point none remains. [Ram75].

This paper concentrates to software testing, namely automated software testing. In chapter 2 general concepts about software testing are presented. Chapter 3 describes the essentials of test automation, what it is, why it should/should not be done, how it can be implemented in a software organization etc. In chapter 3 connections between the testing processes with the business processes of an organization are discussed, because it's essential when considering the practice of software engineering, but unfortunately often left out when discussing quality models. In chapter 4 a process model for test automation is presented and chapter 5 gives a practical example through an fictitious case study of a project implementing that process. Chapter 6 contains some final thoughts and a conclusion to the matters presented in this paper.

2 Software Testing

According to the British Standard 7925-1 Vocabulary of terms in software testing, 'testing' is defined as 'the process of exercising software to verify that it satisfies specified requirements and to detect errors' [BCS99]. Software is tested, because we want to know if it does what we want it to do and that it doesn't do anything we don't want it to do. Without testing, we have very little knowledge about the actual quality, capabilities and correctness of the software product in question.

2.1 Testing = Quality Analysis

One misconception often made is that testing will automatically improve the quality of the software product. Testing provides quality only indirectly, because directly it only provides information. Software development produces an artifact, which does something. Testing produces information about the quality of that artifact [Bul00]. In order to improve the quality of the product the information gathered by testing must be used by managers and software developers to actually change the software. If we look at testing from this perspective, we could say that testing is actually quality analysis i.e. analysis which produces information about the artifact's quality.

3 Test Automation

The BS 7925-1 standard defines test automation as 'The use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions'.

The goal of test automation is to make some activities of testing more efficient than what could be achieved manually. By automating tests, they can be ran more frequently than manually e.g. when doing regression testing. Also large scale performance tests are easier to automate using tools, than to simulate hundreds of users manually. Automated tests can be reused and are consistent and reliable in the sense that they will be executed in an exactly same way every time [FeG99].

However automating testing is hard. As with software development projects, most test automation projects will fail unless they are controlled. Actually test automa-

tion projects are even more prone to fail than development projects, because much less effort and care are given to testing projects compared to development projects.

3.1 Automated is not Automatic

General attitude towards test automation is that the main activities concentrate in automating the test execution of test cases. Another misconception is that this will give you automatic testing. However there is more to test automation than just automating test execution, as there is more to a test than just running it. Usually before the execution of test cases there's some setting up of the test environment to do and after the test cases have been run, the results must be checked, logged and possible found defects reported [FeG99].

3.2 Automated Tests vs. Automated Testing

Automated tests and automated testing are two different concepts within the scope of test automation. Both are valid goals for automation, but it's important to recognize the differences. The difference has to do with the pre- and post-processing of test execution and is depicted in figure 1 on page 4.

On the left hand side we have automated tests, which leave all the test pre- and post-processing for manual work. Analysis of test failures is done most probably after the particular test case has been run. With automated testing on the right side all the pre- and post-processing is done automatically, and the test suite can therefore be run, for example, overnight. However, the first thing in the next morning that the tester will probably be doing is analyzing the failures of the entire set of failed cases, which can amount to a considerable amount of work [FeG99]. In case of automated testing, it is also possible, that something happened during the test run, e.g. the test machines crashed, which makes the results useless. These factors have to be taken under consideration when planning the automation.

3.3 Evaluating Risks

Risk evaluation is an essential part of the test automation process. To begin with we must identify the business area risks that the automation project will address. This will be analyzed in the global analysis phase in the beginning of the project when the overall use of test automation must be evaluated. If there's no risks to business

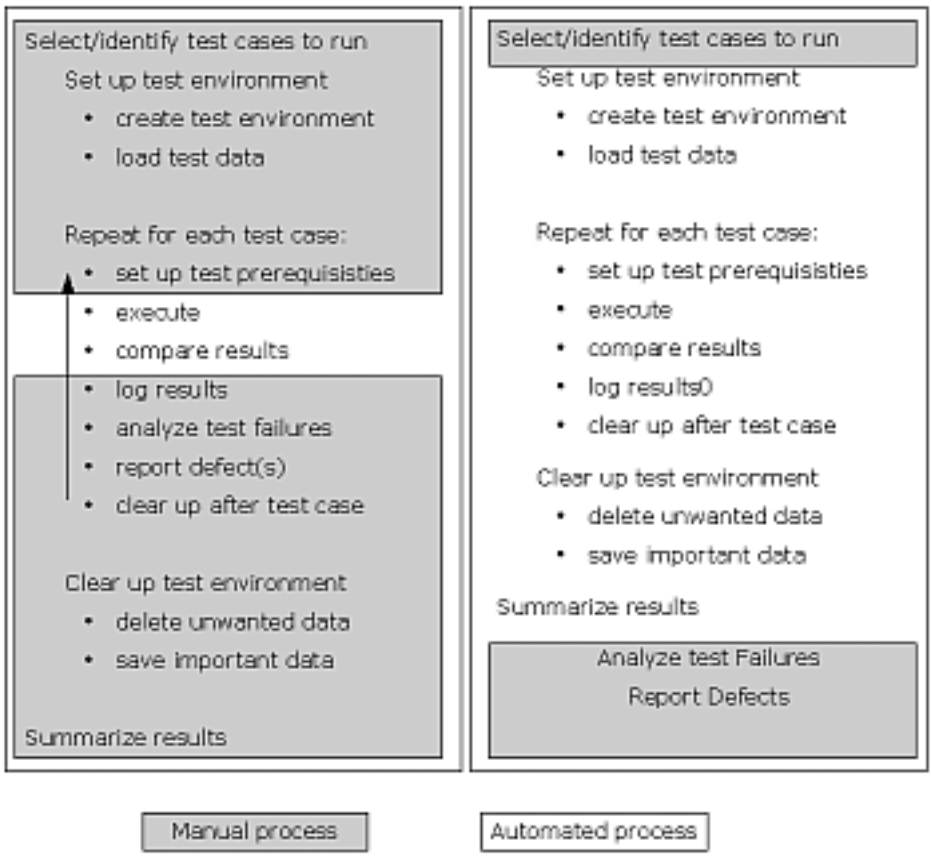

Figure 1: The difference between having automated tests and automated testing (from[FeG99]).

to begin with, why automate testing ? Then the risks which the test automation project itself imposes to the business of the organization (e.g. how likely are we to go over budget with test automation ? Can we keep our production level at a sufficiently high level during the automation ?) have to be analyzed. There's no point in automating testing if we are going to run ourselves out of business while doing it.

3.4 Connecting Test Automation to Business Processes

So how to connect the test automation process with other business processes of the organization ? Test automation connects to the development and support processes by the information it provides for those processes. In addition to those two parts of the organization the test automation process must interact also with the executives

i.e.decision-making processes [Bul00].

First of all we have to identify all the stakeholders that will be influenced by the process. The management (or whoever is going to pay for the project) is most probably going to be very interested on where you are going to spend their money. Project managers in the software development department want to know how the test automation will affect the work of the developers. Other stakeholders influenced by test automation include e.g. the production, maintenance and testing organizations. By testing you produce information, and for that information you need a customer, otherwise it's useless information [Bul00].

3.5 Costs and Return of Investment

Automating testing is hard and costs money as it is with any quality related activity. Although it's often forgotten by practitioners of quality assurance, e.g. testers, everything is dependent on costs and benefits when it comes to software engineering as a business. Thriving for perfection can drive you out of business fast if you're not taking care of your business at the same time. Testing is not a moral issue, it's not done just because it's the right thing to do. [Bul00].

The hard part is actually evaluating the costs of test automation and the return of investment. The values of testing need to be quantified i.e. you need to specify in understandable terms that test automation will be wroth it. For example, you can estimate, that after the testing has been automated the coverage of functions tested compared to the present will be 80the increased coverage will most probably result in saving of so and so much money because the defects are found before releasing. In order to be worth the money and effort, the automation project has to be carried out in proper.

3.6 Justifying Test Automation

Why to automate testing then? Test automation can be justified in a software organization if it is reasonable to believe, that test automation will provide financial benefits to the organization. Of course it takes more than blind faith and a thorough analysis of the costs and benefits of introducing test automation has to be made before we can start planning for test automation.

Also the progress of the automation has to be monitored so we can show to the decision makers that the automation is proceeding and that we are getting the benefits that we sought after in the beginning.[Bul00][FeG99].

3.7 What to automate

So test automation can be beneficial to a software organization. But tests should not be automated just because they can be. Test automation can be helpful but only under certain circumstances. Only automate what needs to be automated. You won't get many friends using a half years development budget's worth of money into an exquisite automation framework which is no use to anyone. It's usually quoted, that automating a single test case takes about 3-5 times longer than to execute that test case manually [FeG99]. So the candidates for automation have to be selected carefully. Stable manual test cases e.g. regression test cases, are usually ripe for automation, but there's no point in automating test cases if the software under test is known to go through significant changes, which would result in changing all the automated test cases.

4 Test Automation Process - The Theory

It would be reasonable enough to question, whether there should be a process model for test automation. Couldn't we automate testing without a process ? Of course we could, as well as we could develop our software without any processes. Then of course we would probably lose all the repeatability, visibility and control to the project.

The problem with software quality is not that we don't have good enough programmers. The problems emerge when we have many programmers working on the same project, which is when we need some form of management. Process is a tool for managing the complexities inherent in software project. Test automation is not any less complex or 'hard to do' than developing software is. Therefore it's more than reasonable to have a process for managing test automation projects.

In this paper I approach test automation from a project point of view and therefore I

treat the process model as an abstraction of that project. The process outlined here shares characteristics with many other software processes. In fact most development processes more or less include the following high level phases:

- Analysis
- Evaluation
- Action
- Post-evaluation

In the test automation project, those phases are included within a single iteration cycle. Additionally the process has phases for Global Analysis, Conclusion and Reporting. High level flowchart of the process can be seen in the figure 2.

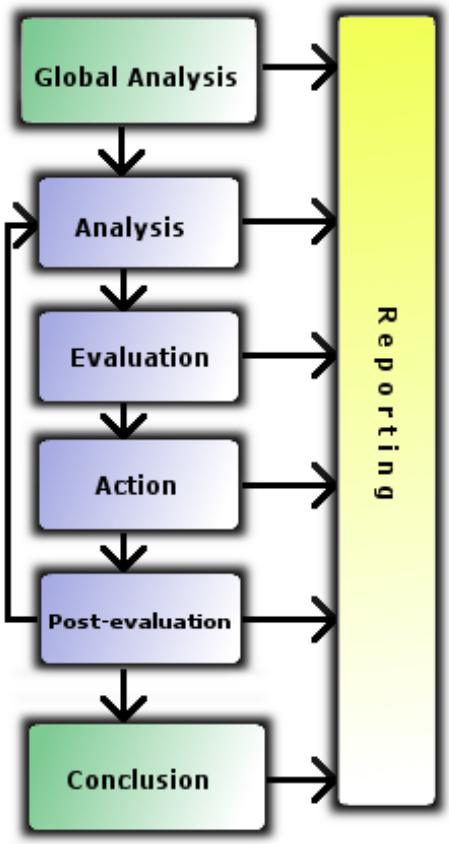

Figure 2: High level view of one iteration in the Test Automation Process

4.1 Process Model

This section outlines the actual model for the test automation process. The process model is a theoretical abstraction of a prototype automation project. However this process approaches test automation from a specific point of view and test automation could also be implemented in many other ways as well. In the following the phases of the process are explained in more detail.

4.2 Global Analysis

In the global analysis phase of the process the organizational factors are addressed and analyzed. As a result of this phase we should have a clear picture on which business risks test automation is planned to alleviate. For example, we analyze that the quality of the software must be improved because bad quality can result in the failure of existing business functionality. In this case test automation can be a valid alternative for seeking that improved quality.

4.3 Analysis

As the analysis is the information gathering phase, it is essential to use enough time and resources to this phase, because the results of the subsequent steps rely heavily on the information that is gathered here. If you have no idea where you are in the first place, it's hard to define how to get where you have to go.

When an iteration cycle begins, the current situation will be analyzed. The effort required by this phase will be the greatest at the beginning of the project i.e. during the first iteration cycles. When the project proceeds the analysis phase mainly functions as an assuring phase that the project is proceeding as planned and that no new surprises have been encountered.

4.4 Evaluation

Evaluation of what can, should and will be done is the most laborious part of the process. Success of the evaluation is also very much dependent on the professional skills of the person doing the evaluation. First of all, we select a set of actions which we want to execute in order to achieve the goals we have set for test automation. This first set of actions contains our 'dream set' i.e. it's the theoretically sound set

of actions if it weren't for the reality of business environments. Next we evaluate the different requirements of each of the actions in the preliminary set. For each action defined in the process model we have explicit requirements and constraints. For example if the action would be 'Develop Perl-script tool for evaluating the contents of the software under test on a file level', it could have a skill requirement so that the action can be only performed if we have Perl-skilled resources for executing the tasks.

According to the action requirements and constraints we reduce the set of actions and then evaluate them against business requirements, risks and prioritize the action set and select the final set of actions for the current iteration after which the actions are performed. Possible actions and their requirements and constraints are presented in the process documentation.

4.5 Action

Performing the actions in the iteration cycle is the actual 'hands-in' work involved in the automation project. This includes developing scripts, frameworks and testware for testing. Note that these actions don't include executing the tests as that should be the responsibility of a separate testing process. The actions of the test automation process for example enable the automatic execution of test cases by developing tools necessary to do so, but the actual testing itself is not in the scope of this process.

4.6 Post-evaluation

The post-evaluation phase deals with the both self evaluation of the process itself during iterations and the results produced by the actions in the iteration. The post-evaluation of the iteration provides information for the next analysis phase and can also affect higher level decisions if we find out something new that didn't come up in the global analysis. As the post evaluation phase is also the last in the iteration it also deals with progress reporting to different stakeholders.

4.7 Conclusion

When the project ends - and by all means it should at some point - the conclusion phase contains all the activities necessary to bring the project to a controllable and graceful end. Instead of just abandoning the project and considering the case closed,

the conclusion activities make sure that all the artifacts produced by the process are documented and stored for future development and maintenance.

4.8 Reporting

Reporting is a very essential task often overlooked by professional testers. If information doesn't flow within the organization, you're pretty much adding no value with automating the testing. Stakeholders paying for the automation will most probably want to know how things are proceeding at any given time. Developers need to know what automation is going to give to them as well. If nobody knows what you are doing, they are not going to trust your work either [Bul00].

5 Test Automation Project

Introducing test automation to - or improving in - an organization is here approached from a project point of view in a form of a fictitious case study. Activities within a single iteration cycle of the project can be seen in the figure 3 on the page 11.

5.1 Analysis provides the base of knowledge

Global analysis tells us that we have a team of 5 testers with various experience in the field. Three of them have also other responsibilities within the organization. The organization wishes to get more value for money when doing testing. They have also been having problems with the quality of Product A.

When we start the first iteration of the project we find that an set of manual smoke test cases exists for Product A. There are two software products developed: Product A is a stable product from which new versions and customizations are developed regularly. Product B is a new product which is very much under development and hence prone to changes.

5.2 How to evaluate what should be automated?

From our analysis we know that among the next potential actions would be setting up configuration management for the automation scripts we are going to develop.

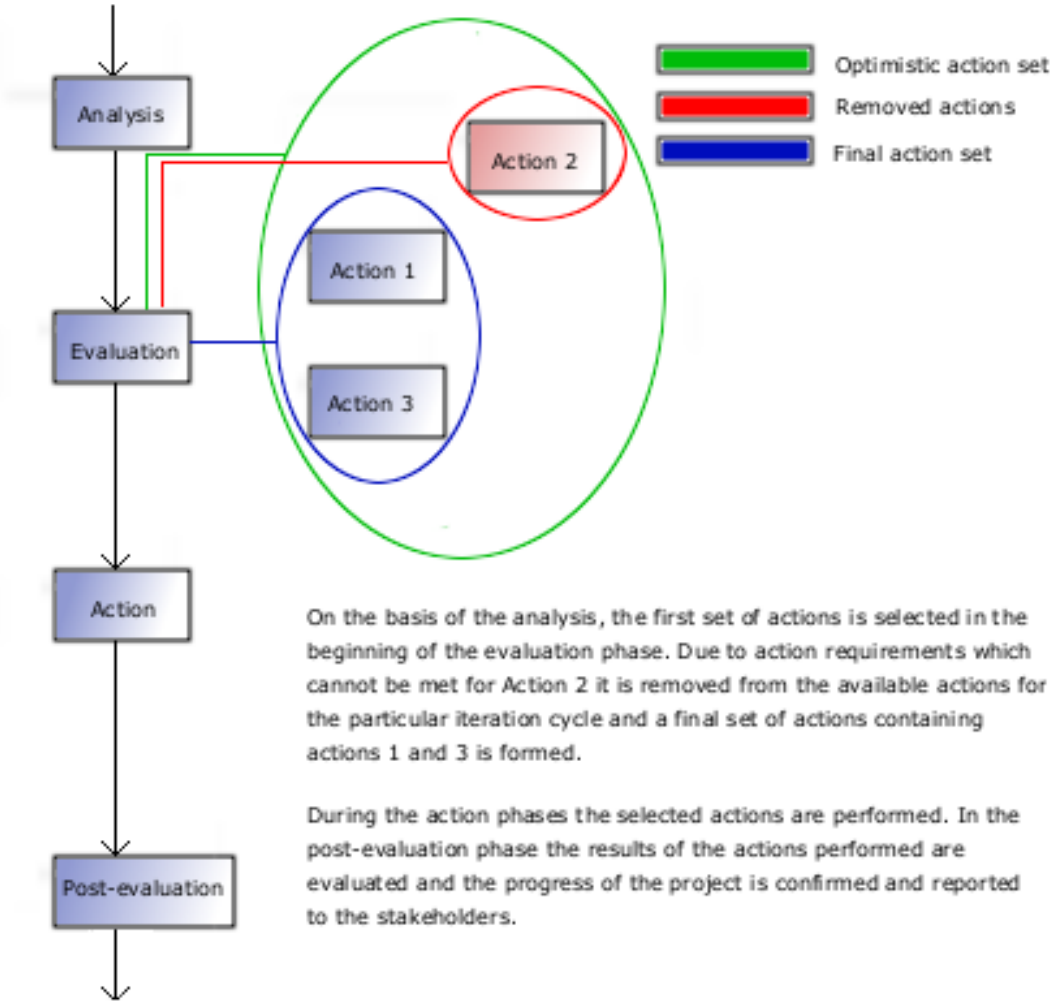

Figure 3: The phases of the process during a single project iteration cycle

Let's say this is Action 1. Action 2 is developing a test suite of scripts for testing the GUI of the software under test. Automating existing smoke tests is Action 3. We don't select any automating actions concerning Product B, because it is not stable yet and the maintenance of the automated tests would at this point take too much resources. Therefore it's better to leave the testing of Product B manual. From the action requirements we find that automating the GUI testing requires skills that we don't have available in our test team. We make a note of this to the report, but exclude this action from the iteration. Options in the next iteration would be to provide training for our test team for GUI test automation.

5.3 Performing Actions and Post-Analysis

The actions selected in the previous phase are resourced, planned and executed by the members of the test team. For smoke test case automation Perl-scripting is successfully used. Post-evaluation reveals that configuration management could be shared with the development team and this can be further examined in the analysis phase of the next iteration.

5.4 Project Conclusion

During several iteration the automation project is considered to have reached the set goals. During the conclusion phase all the documentation and scripts produced during the project are given a final review and stored for later retrieval. Final reports are presented to the management of the organization, which show that now with test automation we were able to execute 23cases than previously. Defects were found more often after automation and the overall defect density in the final release has dropped to 17 defects per module from 54 before automation was introduced.

5.5 Reporting during the project

Progress reports during the project kept the management and other parties informed on the progress of the automation project. This was found extremely helpful at a point of time, when the analysis discovered that to further proceed with automation training for the developers would have to be provided. The management then decided to increase the budget of the automation project because they were able to see why the money was needed and what it would get for them.

6 Conclusion

In this paper various factors affecting test automation have been discussed while discussing a process model for test automation. Too often the practice has been not taken into consideration when presenting process models and I have tried to tie the process model to the business processes that exist in the commercial software engineering organizations. Test automation is a useful tool if used right, but implementing test automation in an organization requires at least the same level of professionalism as does software development.

The process model described in this paper is at the present only a theoretical model, which will be further developed. The process follows a general iterative work-flow of analysis, evaluation, action and post-evaluation. At the start of the process a global analysis of the business processes is made. As a result of the global analysis organizational factors affecting the test automation project have been identified. At the final stage of the process a conclusion phase will bring the project to a controlled end. Parallel to all phases a reporting phase will operate to give stakeholders information on the current status and the progression of the project.

References

BCS99 BCS SIGIST, *BS 7925-1 Vocabulary of terms in software testing*, v6.2 edition, 1999.

Bul00 Bullock, J., Calculating the value of testing. *Software Testing & Quality Engineering, May/June, 2000.*

FeG99 Fewster, M. and Graham, D. *Software Test Automation*, chapter 1,2,3. ACM Press Books, 1999.

Ram75 Ramamoorthy, C. and Ho, S., Testing large software with automated evaluation systems. *Proceedings of the international conference on Reliable software*, 1975, pages 382–394.