

Implementing Software Testing Automation Tools

A continuous process improvement approach

This document describes a continuous process improvement approach to implementing software test automation tools into an organization

***“So if you look at how progress is made and where competitive advantage is created, there's no doubt that the ability to use software tools effectively is critical to succeeding in today's global knowledge economy.”,
Bill Gates, BBC Interview, 2009***

www.TestManagement.com

Traq Software Ltd
St. Mary's Court, The Broadway,
Old Amersham, Buckinghamshire.
HP7 0UT United Kingdom
Telephone: +44 (0)1494 582037

Why Implement Test Automation Tools?

If you currently perform manual regression testing for each product release, have a large number of tests that need to be executed for each release, and you have frequent product releases then this is an ideal situation to make use of a test automation tool.

Manual regression testing is an inefficient use of a test analyst's time, especially if they are merely repeating documented tests. You can reduce your test cycle times and improve the accuracy of your testing by letting a software test automation tool execute these tests instead.

How to Start – The “Ad Hoc” Approach

So let's assume you purchase an automation tool for the test analysts to use but you find things do not go as you expected:

- Some analysts do not want to use the tool to automate their tests preferring to keep on testing manually.
- Some analysts give up using the tool, but some users are using the tool successfully.
- After several releases the application has changed and a significant number of automated test scripts need to be updated.
- The tool was expensive to purchase and so much time and resources are being used to maintain the automated test scripts it seems the tool is of marginal benefit.

This is a small sample of the problems that are likely to occur when you attempt to adopt a test automation tool in your organization in an ad hoc manner.

A Better Approach - “Plan - Do - Check - Act”

When you adopt any tool into your organization you intend to improve your current process but remember you are also changing the current process and this can be problematic.

Quality assurance and quality control people have developed a system that can make changes like these less problematic. It is an ideal way to adopt test automation into your organization.

This system consists of four steps, plan–do–check–act (PDCA).

Overview of PDCA

1. **Plan** a small-scale project, e.g. learn and use a test automation tool.
2. **Do**. Execute the plan, e.g. assign a resource to learn the tool and automate a suite of regression tests.
3. **Check**. Review the project, analyze the results and determine what you have learned.
4. **Act**. Take action based on what you have learned in the project:
 - i. If the project did not work, go through the steps again with a different plan, e.g. maybe use a different tool if the first one did not work out.
 - ii. If the project was successful, incorporate what you learned from the project into wider changes, e.g. automate all of your regression tests, or use the tool on other projects.
 - iii. Use what you have learned to plan new improvements, beginning the PDCA cycle again, e.g. make use of advanced automation techniques such as data driven testing.

PDCA Project Example

PLAN

Project: Develop a project plan to incorporate a test automation tool into your current testing process on a trial basis and automate a series of regression tests. Keep in mind that most tool vendors have a 30 to 60 day free trial period for their tools.

Process Definition: Define your current process. Incorporating a tool will fundamentally change your process, so benchmark your current process.

Data collection: To justify the investment in the tool you will need to determine the value it has added to your process, i.e. this is a cost-benefit analysis. Existing data to collect for the cost-benefit analysis will be:

- i. Number of manual regression tests.
- ii. Number of times regression tests were executed in previous releases.
- iii. Number of people required to execute the tests.
- iv. Time taken to execute the tests.

DO

Implement the plan. Collect data for the cost-benefit analysis and record observations during the project. The data to collect for the cost-benefit analysis will be:

- i. Number of regression tests automated.
- ii. Number of times regression tests were executed.
- iii. Number of people required to automate and execute tests.
- iv. Time to automate the tests.
- v. Time to learn how to use the tool.

CHECK

Review the project, analyze the results and identify what you've learned. You can now determine the value using the tool has added to your process, with:

$$\text{Value} = \text{Benefits} - \text{Costs},$$

Where:

Benefits, represent the time and resources saved in executing automated tests instead of manual test.

Costs, represent the time and resources invested in automating the tests.

In addition to the Value added, you now have a basis for training others to use the tool.

ACT

Let's assume the project was a success, i.e. the tool added value to your process, now you can act on what you have learned in this project, i.e.

- i. Update the current test process to incorporate automation.
- ii. Start collecting cost and benefit data during other projects that incorporate automation.
- iii. Develop a training regime for others to learn the tool.

At this point you are in a strong position to plan your next PDCA cycle for test automation. For example, you could automate the remaining regression tests for your next project, and follow that project by incorporating automation into other projects, or incorporating advanced automation techniques, etc.

Continuous Improvement

You now have a basis for continuous process improvement using test automation; however, as you continue to use automation to improve your test processes, and as test automation is adopted throughout your organization, the following overall pattern will emerge,

- There will be an increase in the people and resource investments required for things like training, user support and documentation.
- As such, there will need to be greater understanding, attention and commitment from management towards automation.
- Consequently, you will need to have further controls, measurements, and reporting in your processes to provide the feedback that will tell you if your improvements are adding value.

Summary

In summary we have covered a continuous process improvement approach to implementing test automation into your organization.

In doing so you can define what you want to accomplish, choose the changes you want to make, and determine the criteria you want to use to determine if the changes are adding value to your process.

What to do next

Traq Software specializes in providing software testing tools and test automation resources to the software testing industry. For help and support with your next software test automation project please contact us.