CSTE Certified Software Tester

QUALITY ASSURANCE INSTITUTE WORLDWIDE

CHAPTER NETWORK

BEST PRACTICE
Solutions Workshop Notes for ‘Building Effective Measurement Solutions

Date: 4/30/08
Facilitator: Molly Byer
Co-Facilitators: Bob Crews and Shalla Hyderi Goyal

TOPIC 1 - TOOL SELECTION

How do I select the “right” tool? Is there a checklist or set of evaluation criteria?

1. Perform a Proof of Concept as a part of tool selection.
2. Create unique requirements for the proof of concept – Set up to 10 scenarios for the vendor to prove they can setup and check if they can do it satisfactorily. During that process, change the requirements and see how they are handled.
3. Know the future technology roadmap of the company.
4. Make sure that the tool is scalable. Get a list of Must Have’s vs. Nice To Have.
5. Look into the support agreements from vendor including tech support.
6. Validate that the data/template are compatible across vendors and adhere to industry standards for long term maintenance.
7. Find if different kinds of data can be used. Ask the vendor what platform they support.
8. Check if there is any available freeware to gain additional opportunities. List all tools and select the one that matches close to your needs.
9. Staffing and usability of tool – Weigh in and get feedback if the testers are comfortable using the tool.
10. Look to see if a common tool set can be setup ie. If it can address load test, UI test, performance test etc.
11. Know the language, or nearest equivalent, that the tool uses, such as C or VBScript.

TOPIC 2 - AGILE

What are the suggestions on the best approach to maintain automation scripts (artifacts) in an agile environment?

1. Most of the (Happy path + negative path) regression testing can be automated in agile. Look for low hanging fruits.
2. Keep it small and simple.
3. Use modular approach. Identify repetitive modules used across build.
4. Prioritize the test cases. 60% development is okay during early phases and don’t have to be 100% to attain quick efficiencies and avoid rework. The testing during the early phases can be addressed with appropriate level of automated in conjunction with the manual testing.
5. Present Return on Investment. Find if the automation is feasible for the phase as the project evolves.
6. Automation scripts have to find some defects to show benefits.

TOPIC 3 - MAINTENANCE

How do you plan for and manage the high cost of maintenance in test automation?

1. Use Regular Expressions/ functions for objects if these objects are expected to change. (QTP 9.5 can generate RegEx pattern matching strings automatically in Maintenance Run mode.)
2. Use Descriptive Programming to develop Object oriented test scripts, and not based on mouse/keyboard clicks/screen locations.
3. Breakdown scripts into functional modules.
4. In QTP, don't attempt a single script library full of Actions. Drive towards one reusable action/one script, and include a non-reusable unit test action.
5. Parameterize values to keep hard coded data out of scripts and increase scalability.
6. Identify automation test data with additional characters so that test data is not “stepped on” by other manual testers. If possible, create test scripts that, as a prerequisite, generate their own test data.
7. Tighter Change management, especially in an agile environment.
8. Ask your developers to uniquely name each object, and to define which property will be used for each object type.

TOPIC 4 - VALUE

How do you realize rapid and continued value in test automation? What specific measures/metrics have proven successful in showing value and how do you collect data and calculate?

1. Look for risk exposure from the business point of view to assess value.
2. Record times when you were able to identify a problem which would not have been found otherwise or the probability of finding that would have been low. Convert the value to cost if that defect had made it to production there by increasing the overall cost of quality.
3. Also recognize the times when testing could not have been completed by just manual testing based on time constraints.
4. Get metrics to present cost estimates.
5. Avoid tracking 'Number of Defects Found' and 'Number of Test Cases Passed'. Focus on 'Defect Severity' as well as 'Hours of Execution Time' multiplied by an average billing rate. Factor in overtime pay if tests run over 8 consecutive hours in a day, or over 40 hours in a seven day week.

TOPIC 5- ORGANIZATION ADOPTION

What steps should be taken to ensure organization adoption and education at individual/teams/management levels?

1. Assumption: Management buy-in complete for automation effort. Tool has been identified. Identified stakeholders from different areas of the organization and identified their expectations based on the importance.

2. Demonstrate a sample of automation along with their benefits to the identified stakeholders.
3. Educate and create awareness of the benefits of automation to the organization to gain support. Few activities could to be quantify the power of automation in terms of output size pre-automation and post-automation, early defect identification benefits etc.
4. Ability of automation to address more scenarios for regression purpose leading to less stress on workers. Leads to a culture where work life balance is enhanced.
5. Get co-worker's buy-in by automating the tedious time-consuming tests. In addition, automate any time consuming prerequisite steps.
6. Get developer's buy-in by initially repeating subtle defects manually on a system without the automation tool installed. Ask for their input in automation code reviews. Ask them what areas could be tested deeper. Ask if they have considered automated unit testing.
7. Maintain Management buy-in in terms of revenue: Potential cost to the company for each defect identified, cost equivalent of automated testing performed manually to identify defects, as well as increased testing depth and scope.
8. Identify a team member who is comfortable with public speaking to present your progress. Join a speaking group such as Toastmasters, if necessary.

<u>TOPIC 6 - CHANGING REQUIREMENTS</u>

What are the recommendations on when to start automation when requirements are still evolving? What should be the first steps?

1. Identify more stable requirements vs dynamic requirements. Go with stable requirements first and also get better insight to the nature of the dynamic change. Overall, understand that planning is an integral part of automation effort.
2. Determine features of automation tools which assist in rapidly revising scripts (ie QTP has descriptive programming), however, the issue is this requires appropriate skill set.
3. Identify areas that can be automated and can be reused though requirements are still changing. This can be done by creating framework in modular fashion.
4. Involve QA in requirement process to gain better insight. Also peer review test plans to gather feedback from other groups.

References

Conference – Quality Engineered Software And Testing Conferences - Chicago April, 30, 2008 (QUEST)
http://www.qaiquest.org/chicago/