Staffing Your Test Automation Team

© 2002-2009, Mosaic, Inc.
www.mosaicinc.com
205 N. Michigan Ave.
Suite 2211
Chicago, IL   60601
312-819-2220

INTRODUCTION

In today's fast paced business environment, test automation has increasingly become a mission-critical activity. Information systems are no longer back room systems. Today's systems impact the bottom line of a business, give it a competitive advantage and more importantly, are frequently responsible for increasing sales revenue. Consequently, the risks associated with developing these systems have escalated.

How can we adequately test a system not only during its development, but also over the life of the system? Frequently, the answer to this question is "**all we need is the right tool**".

Unfortunately, while automation may be the answer, a tool is not enough. A company also needs a proven testing process and a skilled test team structured to meet the challenges of automation. The purpose of this article is to examine how Mosaic has addressed the automation challenge and explain how the critical elements of tool, process and test team work together to provide the necessary foundation to achieve the payback, cost savings, and reduction in defects promised by the tool vendors.

PROGRAMMING AND SUCCESSFUL TEST AUTOMATION

To many IT professionals, test automation is synonymous with capture/playback tools. However, the traditional capture/playback method of automation so prevalent in vendor demonstrations is not adequate to address the needs of complex, multi-environment systems. Both manual and automated scripts are required to provide the robustness, maintainability, and reusability so crucial to long-term automation success.

Furthermore, multi-tiered object-oriented systems require monitoring of communication among components during testing. This is beyond the capability of the automation tool. A tester has to do hands-on programming to perform this task. Like all development efforts, the tester must base his or her scripts and programs on sound requirements – in this case the testing needs of the application. Due to the need for manually coded test tool scripts and supporting programs, successful test automation requires a test team to do development work in addition to their traditional testing work.

A STRUCTURED PROCESS

Successful test automation also mandates a testing process. Just as a developer needs a system development process, testers need a testing process to successfully use test tools. The testing process provides the steps, guidelines, and techniques that will ensure practical, successful automation. To achieve the testing and risk management goals of the project, a solid testing process is essential to focus the test automation effort where it can do the most good.

Mosaic developed its own testing methodology, MSTAR™, to consistently focus testing resources and tools where they are needed to meet the system's coverage objectives. MSTAR™ is used to determine the areas most appropriate for automation and to provide an effective framework to accomplish the automation. MSTAR™'s testing process also assists automation engineers in developing maintainable and reusable scripts. No matter what testing methodology is utilized, however, the most important point is that a company has a methodology. Otherwise, the full benefits of testing automation will never be successfully achieved.

MANAGING AUTOMATION EXPECTATIONS

In order to implement a successful automation effort, testers need to educate management on a number of different issues. Educating management on these issues could mean the difference between a successful automation effort and a failed automation attempt.

One such important issue addresses integrating test automation into the entire development process. Testers need to educate senior management as well as scripters on how a specific tool will fit into their software development environment and software development life cycle. It is equally important to remind everyone that manual testing does not end when automation is implemented.

Another issue concerns the purchasing of tools and their impact on test planning. Test planning, planning what to test and how to test, becomes much more complex and more important when a tool is purchased. For example, test automation is rarely justified for all testing. The tester needs to determine what tasks make sense to automate and what does not make sense to automate.

Moreover, an organization needs to understand that automation doesn't eliminate manual efforts on the part of testers. Testers need to maintain automation scripts and verify the execution of automated scripts. This is manual work that needs to be factored into the test team's testing activities.

In addition, in order to automate successfully, testers need training and time to master test tools. Many managers view test tools as simple capture/playback programs that a tester can learn and implement in his or her spare time. This is a misconception. Successful test automation frequently requires the use of complex tools – tools that are much more complex than just capture/playback tools. In order to use these tools effectively, testers need considerable training. Management needs to be committed to training testers on these tools and to establishing the infrastructure in which the tool operates.

Also, successful test automation requires the involvement of the software development team as well as testers. In the past, developers and testers had minimal contact. Frequently, the philosophy was "to throw code over the wall and see if it works". This type of interaction does not work when automating the testing process. Developers and testers will need to work much more closely together. Developers will have to provide support personnel and technical information on their development methods. They also will be asked to use the automation tools during their unit and/or integration testing. More troubling to the developer will be that test automation may raise concerns about the testability of their code. This especially will be the case if standards are not followed or if developers use odd, homegrown or even very new libraries/objects.

Finally, an organization's project management team must have a clear understanding about the types of roles and responsibilities required for a success automation effort. The creation of the test environment starts when an organization purchases hardware and installs a tool. Then the test team and development team need to work together to build and maintain a test automation environment that may include dedicated servers, workstations, databases and the like. Management needs to include development of the test environment in the overall project plan. They should also budget for the resources needed to successfully automate testing.

ROLES AND RESPONSIBILITIES IN SUCCESSFUL AUTOMATION

Mosaic consultants have found that in addition to a testing process, it is essential to have dedicated automation testing resource(s) when using tools to support the test team as they automate the testing process. The number of testing resources will vary depending on the size of the project or organization. However, the following roles, duties and skills will help ensure the success of the testing and automation effort.

Testing Role	Duties	Skills Required
Manual Test Engineer	Design and develop manual scenarios from system requirements.	Good at testing and documentation.
Automation Test Engineer	Design and develop automation scripts. These may be developed from manual scenarios.	Good knowledge of tools and testing.
Lead Automator	Design and develop automation architecture, assist in design and development of test automation across projects. Responsible for in-house training, knowledge transfer and collaboration.	Excellent knowledge of test automation tool and client's practices and procedures, development skills.
Automation Environment Expert	Design and configure test environment. Update automation software.	Excellent knowledge of test environments, tools and practices and procedures.

Manual Test Engineer

Even with test automation, the manual test engineer remains a key role. This individual analyzes project risk, identifies project requirements, develops the documentation to design the overall testing scenarios, and specifies the test data requirements to achieve the needed test coverage. The manual test engineer also works with other automation resources to determine which areas are suitable for automation and which areas are not. For those areas not suitable for automation, the manual test engineer will write, maintain, and manually execute the tests.

The manual engineer should be trained in the test tools in order to run the automated scripts generated by other team members. This allows a single person or group to run all the tests and verify the results from both the manual and automated tests. This also enables a manual test engineer to gain experience in automation tools and further his/her career growth within organization.

Automation Test Engineer

The automation test engineer is responsible for developing the automated scripts. These scripts may be captured via the tool or hand coded in the tool's unique language. It is also the automation test engineer's responsibility to ensure that the scripts are complete and executable. If the organization requires test script reusability, the automation test engineer will be responsible for their maintenance.

Furthermore, the automation test engineer will assist in the execution and verification of the automated script results. This may involve assisting the manual test engineer in locating and interpreting logs and files created by the automation tool.

Lead Automator

The lead automator should have the technical skill of a developer and also strong tool knowledge. The lead automator works with the test engineers to design and develop a reusable architecture for the test automation. This architecture often develops over time. It provides the automation test engineers with reusable functions that improve the robustness, reusability, and maintainability of their test scripts. Having an architecture also allows the test automation engineers to successfully automate with standard tool capabilities and easy to use functions. This in turn reduces a test team's training requirements and speeds up their productivity. Ideally the lead automator should look at the reusability of the architecture not only for a single project but also for projects across the organization.

The lead automator's other key tasks include providing assistance in using the tool, training staff, and resolving technical problems with the automation tool. This last task frequently requires interaction with the tool vendor's support staff. Having a single point of contact with the tool vendor will often facilitate issue resolution.

The lead automator should also be the tool's advocate within an organization. He or she should establish tool standards that any project team would follow when using the tool. These standards include setting up the roles as defined in this article, and scripting, environmental and coding standards that best suit the project and organization. If an outside consultant is used, there should be a plan in place to transfer his or her knowledge to your staff.

Automation Environment Expert

Finally an organization should include the role of automation environment expert to the development team. This role could be assumed by the lead automator for a small project or organization, or by an individual from the operations staff. In order to successfully execute his or her automation tasks, the expert will need system access at the administrative level. These tasks include the initial installation and configuration of the tool and the design and configuration of the tool/test environment for a given project. It also means upgrading the test tool and environment as needed.

The expert also may be responsible for installing the client side of the tool as well as managing any licensing issues associated with the tool. It is possible that the expert may need to un-install after the completion of one project so that the tool can be installed on a different set of devices. He or she also might need to increase the number or type of licenses to conduct stress or performance tests. These changes often occur and an expert needs to be available to provide these services.

CONCLUSION

Successful tool automation depends not only on tools but also on a standard testing process and the right test team roles, duties and skills. Tools, process and test team are the three essential legs of the test automation stool. Moreover, the automation test team needs to have a blend of testing, programming, and tool knowledge. If an organization wants to reap the automation benefits promised by tool venders, it needs to use the tool as a complement to manual testing. It also means adopting a strong test methodology and training the test team on the ins and outs of the selected tool in an organization's unique development environment.