Test Leadership and Management in the Age of AI
White Paper

Neil Price-Jones

NVP Software Solutions

www.nvp.ca

Introduction

Software Testing has a long history of automation and process improvement that has not always moved in one direction. Frequently the adage "One step forward, two steps back" could be applied to the history of test automation. Past methodologies have not always been successful and sometimes adoption of new methodologies has made the problems worse. Overall progress has often been achieved by taking the best parts of each methodology and incorporating them into the existing process.

The introduction of AI into software development and testing is disruptive and faster than any other automation experienced. The impact on Test Leads and Managers will be bigger and force many changes in how they operate. This White Paper addresses the challenges faced by Test Leads and Managers and provides some immediate steps that they need to take.

Target Audience

The target audience for this White Paper on Test Leadership and Management in the Age of AI is the following:

- Quality Assurance Managers and Leads
- Project and Program Managers who depend on results from Testing coming from multiple projects to assess trends and system health.
- CIOs who need overall system health parameters.
- Product and Marketing owners who depend on verifiable results to sell their products.

Software Test Automation

There is a long trend of automating aspects of software testing. The following is a very brief list of some of what has occurred in test automation over the years.

1. Gradual expansion of interfaces that could be recognized.
2. Addition of multiple methods of scripting (from specific languages to NLP, and Keyword Driven Testing).
3. Enhanced automation of test generation and execution.
4. Automatic responses to unexpected events occurring during execution.
5. Traceability matrices (manual and automated).
6. Semi and full automation of incident reporting, documentation, and workflow.
7. API emulators to allow more and expanded testing of complex systems.
8. Addition of weighted algorithms to assess and assign risk levels and criticality to issues.
9. Management tools created to manage test cases, test scripts, and defects (in particular).
10. Management tools progressively expanded to encompass more of the Lifecyle and different SDLCs.

A larger history of Software Development and Test Methodologies and their effectiveness is available from Capers Jones.

All of this has led to a reduction in the time to complete certain aspects of testing. The decline has been gradual and subject to reversals when new applications and increased complexity was incorporated. In addition, some systems, with very limited use, have rarely benefited from automation since it is not economically effective to automate or the interface is simply not recognizable.

Other items that impeded the 'progress' of automation include

1. New interfaces, new languages, and expansion of systems have made it harder to build a System Boundary Diagram in my experience. Thus, not only is it hard to determine what to test, but it is also hard to set up all the conditions. Manual testing, if it encountered such a problem, benefitted from the human-in-the-middle that would stop at the problem, perform root cause analysis, and continue with the test if appropriate. Automated tests did not always realise that there were issues and tried to keep executing or else reported a spurious issue.
2. Tools have been built and sometimes incorporated into other tools to address these issues. However, that led to a proliferation of tools for an organisation to buy, train (and sometimes retrain) the users and frequently the usage of the new tools was limited to very specific cases. These disparate technical solutions would usually be eventually consolidated into another tool with the cycle then repeating itself.

The impact on Test Leadership and Management has been cumulative although somewhat erratic. Clearly the amount of available information and the level of detail down to the most granular level of testing is now many times what it was before. If the test execution and incident reporting is integrated into a management tool, then a status is available to anyone with access and the correct security level. If Requirements, Test Cases, and Test Steps are also incorporated into the tool, then the ability to get a complete picture is greatly enhanced. There is a lot more information and a lot more oversight.

Software Testing using AI

Using AI in Software Testing brings a whole new level of automation and abstraction to the industry. Test scripts, test steps, expected results, execution, defect identification, classification and remediation and requirement generation are now available from within many software test tools or directly via prompts using AI. The addition of Agentic AI will remove many of the listed intermediate processes and allow the test creation, execution, and reporting to proceed in an automated manner with little or no human intervention.

We will see the Test Generation, Test Execution, and Test Reporting squeezed in time, as we have so many times already, via various AI test automation processes and tools. Automation has already provided rapid execution of testcases and instantaneous reporting of results down to the most detailed level. AI will provide this and much more.

Implementation of AI in testing is most likely to follow the same path as previous automation changes. AI will become part of the software testing toolkit for many people, but it will not replace all of what has been done in the past. In the short-term AI will be disruptive (especially since people will be deceived by the hype that it will do everything for testing).

Some organisations will make full use of AI and others will not use it at all due to bad experiences or security and trust issues. As usual the risk equation needs to be applied to all activities involved with AI and a determination made of impacts and probabilities to assess both the short- and long-term impacts to the overall software process. As Robert Sabourin stated at STARCANADA 2025, AI adds a third dimension to the usual Risk Graph with a third axis representing autonomous decisions made by AI.

AI in testing will make a difference to the central part of the testing cycle, but the beginning and end parts are being lost in the noise as is the impact on people who must sign off: the Test Lead or Test Manager.

Impact of AI Automation on Test Leads

The Impact of AI on the Test Lead or Manager will be substantially greater than incremental automation and with a squeezed timeline in both speed of adoption and the test cycle. There will be a reduction in time required for scripting and test execution. Testers who are familiar with writing scripts (either for manual or automated execution) will need to retrain in writing prompts and orchestrating testing. A reduction in the number of testers is inevitable.

The impact on Test Leads and Managers will be that many projects will feel that they need not involve testing until the last possible minute since AI will do all the detailed work that used to be completed by the testers and test leads prior to the execution.   The front-end work will be taken away as will the long execution times and we will have very quick tests generated by AI, run by AI and reported on by AI.

Two Major Concerns

The addition of AI raises two other big concerns.

1. Manual testers would often think about what else could go wrong when writing or executing testcases. If they observed something out of the ordinary on the periphery of a test script or thought about other things that needed to be tested, they could incorporate this new knowledge into the testing and check for any issues. This might be something close to the current test, or it could be something completely new that was in another piece of the software. The act of executing a test provided more information than just the results of the test and often stimulated further investigation. Better testing was the result.
2. The internal workings of the scripts and decisions being made as to what were acceptable changes or even what the automated script (of any type) was testing were often obscured

from the test manager by complex test tools with many settings that could impact how a script was executed and what was reported as an incident. This led to a distrust in automation and many people resisted further automation as a result. In some cases, manual testers were trusted more than any automation despite the chance that the manual tester could make mistakes in execution or fail to notice issues.

Impact of AI

Analysis

AI in testing provides the promise of removing a lot of the tedious work of writing test cases and scripts, executing those testcases, and generating reports. It will even generate Requirements from the testcases if requested.

The issue is whether the results can be trusted and represent a valid test of the system. If critical parts of the test are missed or the incidents are not given the correct priority, then it would be likely that the system could be signed off as ready for promotion based on incorrect and incomplete data.

As a Test Lead or Test Manager who must sign off on incomplete or incorrect information, this is a concern. Information has never been complete in the past, but the concern now is, with so much of the process taken out of direct control, there may be layers of misconceptions and problems buried beneath the steps and interface that are not necessarily visible.

Some will argue that this has been the case for years as testing has been automated and testers got further from the actual testing and into writing scripts. That is somewhat true and this is an ongoing trend. But this is a larger and faster change than other changes.

Problem Statement

The challenge is to ensure that what has been tested is accurately represented in the reports and statistics generated from the testing. A second challenge is to ensure that the required scope of testing has been completed and that the risk to the final customer has been minimised. In the event of an industry that might require a testing audit, the artifacts must be maintained, be safe from modification and be available to the auditor in an accessible format.

Summary

The implementation of AI in testing has been treated as a technical solution so far with limited understanding of the impact on reporting. The Test Lead or Manager will need to change the way testing is monitored and reported with changes in what is checked and when it is checked in the testing process. Items that were originally trusted to be completed and signed off by the relevant people now will be generated and will need to be checked after that generation. This will be at a time when the project is already under time pressure. The generation of test cases from requirement and subsequent execution and reporting without a time gap will need monitoring to ensure that a signoff with assurance of completion and sufficient scope can be provided.

Changes for the Test Lead or Manager

The Test Lead who is dealing with a system in which AI is being used will need to:

1. Ensure their inclusion at the beginning of the project to allow for time and resources to inspect and approve all test artifacts.
2. Plan for the training required for testers in the new method of working.
3. Plan for the automation of the process of Quality Control.
4. Review the documentation to ensure it accurately represents the needs of the users.
5. Build in Risk identification, documentation, analysis and resolution for the new methodology.
6. Shift to Test Governance activities versus test case and script reviews.
7. Plan for the reduced time test cycle while ensuring the correct testing has been completed.
8. Deal with project changes that occur during the active testing cycle.
9. Complete the required scope of testing with reduced resources.
10. Plan for UX testing to be completed by human resources.
11. Ensure that resources and time is budgeted for post execution analysis of coverage.
12. Ensure ALL test artifacts are subject to stringent version control by creating a process or acquiring a tool with sufficient security.
13. Mandate and enforce the assignment of a priority to ALL test artifacts and that priority must carry through all the linked artifacts.
14. Build the process whereby new or changed test artifacts are subject to the same level of control and review as the initial set.
15. Plan for resources and time to orchestrate the review of artifacts throughout the testing process.
16. Build a process to ensure that the AI generated test artifacts correctly and clearly represent a test of the system and reduce the risk of use to an acceptable level.
17. Schedule reviews of the process to check for omissions.

Conclusions

AI can represent a great addition to the tester's tool kit and can help the Test Lead/Manager provide even more data and analysis and free up time for consideration of what might have been missed. This is time for which we have always looked. Caution must be exercised in terms of ensuring that the tests represent and comply with the requirements for testing proof and sign off. We cannot wait until the compressed test cycles are over to find that too much has been missed. Reviews and sign offs must be strategically placed for the best benefit.