DBServices

TEST AUTOMATION LEVELS

Evaluate and evolve your automated testing implementation

Author: Thiago Boschese | November | 2023

Introduction

Test automation has its roots in the early decades of software development, when programmers began to recognise the need to validate the quality and functionality of their programs. Initially, testing was carried out manually, with programmers reviewing the code and computer operators running black-box tests to verify the expected behaviour of the software. Over time, software complexity increased and the demand for more efficient and accurate testing processes grew, leading to the emergence of the first test automation tools and approaches.

The 1980s was a landmark period for test automation, with the development of various tools and frameworks that enabled the automation of black-box and white-box tests. Some of the first tools included "capture and playback", which recorded user actions and played them back automatically, making it easier to carry out repetitive functional tests. However, the limitations of these tools soon became apparent, as they were fragile and often failed to keep up with changes in the software. This led to the development of more advanced approaches, such as the use of scripting languages and code-based automation tools.

Today, test automation is an essential component of the software development process, with a wide range of tools and frameworks available to help teams effectively test the functionality and quality of their products. Continuous integration and continuous delivery (CI/CD) have gained prominence, allowing organisations to release new versions of software quickly and with confidence. In addition, agile practices and the Test-Driven Development (TDD) approach have helped incorporate test automation into the development lifecycle, resulting in significant improvements in the quality and efficiency of the software development process.

Here, we'll discuss how automated testing maturity can be defined in five distinct levels, providing a clear roadmap for development and QA teams to improve their testing processes. These levels not only help you identify where your team currently stands, but also offer guidance on how to move towards a more solid and reliable development environment.

Throughout this ebook, you'll learn about each of these levels, from the initial stage, where manual testing still predominates, to the end state, where a quality culture is ingrained throughout the organisation.
Test automation isn't just a trend; it's a necessity to ensure that your products meet customer expectations and stand out in the market. So dive in with us on this journey through the levels of test automation and start transforming the quality of your software today.

Benefits of Test Automation

Tangible Benefits

Improved Software Quality
Test automation enables fast and accurate test execution, identifying bugs and quality issues earlier in the development cycle, resulting in more reliable software.

Cost Reduction
Although the initial investment in test automation can be significant, in the long term it saves time and resources by reducing the need for time-consuming and repetitive manual testing.

Increased Efficiency
Automated tests can be run 24 hours a day, 7 days a week, speeding up software delivery and enabling faster iteration.

Early Regression Detection
Regressions are detected instantly, preventing errors from spreading to production environments.

Comprehensive Coverage
With automation, it is possible to achieve comprehensive test coverage, including unit, integration and e2e testing.

Intangible Benefits

Improved Customer Satisfaction

With fewer errors and more stable releases, customers are more satisfied with the software.

Increased Team Productivity

Automation frees up the testing team to focus on higher-value tasks, such as usability and exploratory testing.

Rapid Feedback

Automation provides instant feedback on code quality, allowing for immediate corrections.

Greater Confidence

The team and stakeholders gain confidence in the software, knowing that it has undergone comprehensive testing.

Support for Agile Practices

Test automation is essential for agile practices such as Continuous Integration and Continuous Delivery.

Approaches to Test Automation

Approaches to Test Automation

Unit Testing
Automate the testing of individual units of code to ensure that each component works as expected. This is fundamental to verifying the internal workings of the software.

Integration Tests
Automate tests that check the integration between different parts of the system, ensuring that they work well together.

User Interface (UI) tests
Automate the tests that simulate the user's interaction with the software, checking that the user interface is working correctly.

Regression tests
Automate tests that check that existing functionality still works as expected after changes to the code.

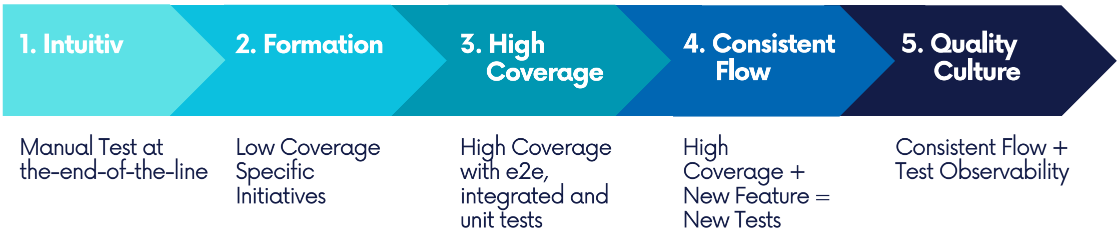
Acceptance Tests
Automate the tests that validate that the software meets the customer's requirements and that the main functionalities work as expected. Translated with www.DeepL.com/Translator (free version)

Test Automation Levels

Test Automation Levels

The creation of Test Automation Levels by DBServices was motivated by the observation of common patterns in the evolution of test automation in different contexts and clients. Based on the expertise of our consultants and specialists (some with more than 15 years' experience in the sector), we recognised the need for a framework that would allow organisations to assess and improve their test automation practices.

Test Automation Levels establishes five distinct levels of maturity, helping teams to identify where they are and how they can improve. The model's maturity levels are described below:



1. Intuitiv

Manual Test at the-end-of-the-line

2. Formation

Low Coverage Specific Initiatives

3. High Coverage

High Coverage with e2e, integrated and unit tests

4. Consistent Flow

High Coverage + New Feature = New Tests

5. Quality Culture

Consistent Flow + Test Observability

TAL1 - Intuitive

At the first level of test automation, we find the initial stage of many organisations. At this point, testing is predominantly manual, and automation has not yet played a significant role in the software development process.

The characteristics of TAL1 - Intuitive include:

Dominant Manual Testing

Manual testing is the norm, and the team does not have a defined automation strategy.

Little Automated Traction

Any existing automation is sporadic and not a fundamental part of the development cycle.

Limited Time and Resources

There is often a lack of time and resources dedicated to test automation.

Scalability Challenges

As software grows, reliance on manual testing becomes unsustainable.

Although Level 1 is a common starting point, it is important to recognise that it presents significant challenges in terms of software efficiency and quality. Many teams realise the need to evolve beyond this stage, looking to adopt more advanced and effective test automation practices.

Common challenges

- Reliance on manual testing can lead to inconsistencies and errors in software validation;

- Manual execution consumes precious time and resources;

- Software problems may not be identified until advanced stages of development or even after release.

Practices to evolve

- Invest in training the team in test automation and the associated best practices;

- Explore, experiment with and adopt test automation tools that meet your organisation's specific needs;

- Seek support from leadership and defend the importance of test automation for improving software quality.

TAL2 - Training

In TAL2 - Training, teams begin to take more decisive steps towards test automation. At this stage, awareness of the importance of automation is growing, and the team begins to devote more effort to its implementation.

The characteristics of TAL2 - Training include:

Focus on Automated Testing

The team begins to focus on creating and executing automated tests, recognising the benefits of automation.

Low Coverage

Although there are efforts towards automation, the coverage of automated tests is still limited, often addressing only the most critical parts of the application.

Exploration of Tools

The team begins to explore and adopt test automation tools, such as testing frameworks and user interface testing tools.

Integration with the Development Process

Automated tests are incorporated into the development cycle but are not yet developed together with the production code.

Continuous Learning

The team is willing to learn and improve their test automation skills.

TAL2 represents a transitional stage, where teams are moving in the right direction, but still have a long way to go to achieve efficient and comprehensive test automation. It's important to emphasise that this stage is fundamental to building a solid foundation for more advanced levels of automation.

Common challenges

- Low coverage of automated tests can result in undetected regressions;

- As automated testing increases, script maintenance can become a challenge;

- Automated tests may not be fully integrated into the development cycle, leading to miscommunication.

Practices to evolve

- Identify critical areas of the application and expand automated testing coverage gradually;

- Standardise the use of testing frameworks and tools to ensure consistency in automation;

- Work to develop automated tests simultaneously with production code to identify problems earlier;

- Continue to invest in training and developing the team in test automation.

TAL3 - High Coverage

At TAL3 - High Coverage, teams reach a significant milestone in their test automation journey. At this stage, the focus is on achieving comprehensive automated test coverage, covering the three main layers of testing: e2e, Integration and Unit.

The characteristics of TAL3 - High Coverage include:

Comprehensive Test Coverage

The team achieves high coverage of automated testing, ensuring that tests are performed on all layers of the application, from end-to-end (e2e) to unit tests;

Defined Quality Standards

Quality standards are established for the automated tests, ensuring that they are reliable and relevant;

Critical Test Automation

In addition to essential tests, the team also automates tests that would be impractical to perform manually due to their complexity or repetition;

Continuous Integration

Test automation is integrated into the Continuous Integration (CI) process, ensuring that tests are run automatically whenever there is a change in the code;

Coverage Monitoring

The team closely monitors test coverage and is committed to keeping it high.

TAL3 represents a major step forward in terms of test automation maturity. Comprehensive coverage of automated tests helps ensure that errors are identified earlier in the development cycle, saving time and resources in the long run.

Common Challenges

- With a large number of automated tests, maintenance can become complex. The need to update scripts is constant;
- Running all the automated tests can take time, impacting efficiency;
- Identifying which tests should be automated can be a challenge.

Practices to evolve

- Identify and prioritise the most critical tests for automation;
- Implement orchestration tools to manage and execute tests efficiently;
- Maintain constant monitoring of test coverage and quality;
- Exploit automated acceptance testing (ATDD) and continuous acceptance testing (CAT) to validate functionalities automatically.

TAL4 - Consistent Flow

TAL4 - Consistent Flow represents an advanced stage in the test automation journey, where teams achieve high-quality automated test coverage while integrating test automation directly into the agile development cycle.

The features of TAL4 - Consistent Flow include:

Abundant, High-Quality Coverage

The team maintains comprehensive coverage of automated tests, ensuring that all critical aspects of the application are tested effectively;

Development in Sprint Time

Automated tests are continuously developed and improved during the sprint cycle, keeping pace with the development of production code;

Automated Acceptance Testing

Automated acceptance testing (ATDD) and continuous acceptance testing (CAT) are commonplace, ensuring that functionalities are validated automatically.

Full integration with CI/CD

Test automation is fully integrated into the Continuous Integration/Continuous Delivery (CI/CD) pipeline, allowing tests to be run automatically at each stage of the process;

Fast Feedback

The team receives immediate feedback on the quality of the code, identifying regression problems quickly.

TAL4 represents a point at which test automation is not just an additional task, but an intrinsic part of software development. Automated tests are developed and maintained in parallel with the production code, ensuring that the software is tested comprehensively and consistently.

Common Challenges

- With continuous automation, the test code base can grow, becoming complex and difficult to manage;

- The team may face resistance to change when trying to integrate test automation deeper into the development cycle;

- Constant test maintenance is required to keep up with changes in the application.

Practices to evolve

- Work closely with developers to develop tests automatically while production code is being developed;

- Implement efficient orchestration strategies to manage large-scale test execution;

- Ensure that test feedback is integrated directly into the development tools used by the team;

- Maintain strict quality standards and carry out regular refactoring on tests to ensure that they remain effective.

TAL5 - Quality Culture

TAL5 - Quality Culture represents the most advanced stage in the test automation and software quality journey. At this level, test automation is not just a practice, but an integral part of the organisational culture, driving quality excellence in all aspects of software development.

The characteristics of TAL5 - Quality Culture include:

Advanced Consistent Flow

The team maintains a consistent development flow, with automated tests developed and improved continuously during the sprint cycle;

Test Observability

The team invests in observability tools and practices that provide detailed insights into the performance and quality of automated tests;

Collective Responsibility

Software quality is the responsibility of everyone in the organisation, from developers and testers to managers and team leaders;

Real-Time Feedback

Feedback on software quality is instantaneous and accessible to all interested parties;

Continuous Improvement

The team is constantly looking for ways to improve quality, automating improvement processes and continuous learning;

Strategic Alignment

Software quality is aligned with the organisation's strategic objectives, contributing to customer satisfaction and market success.

Level 5 represents the pinnacle of test automation and software quality. Here, quality is not seen as an obstacle to be overcome, but as a driver of organisational success. A culture of quality permeates all areas of the company, resulting in high-quality products and satisfied customers.

Common Challenges

- Managing the complexity of automated testing can be challenging in a high automation environment;

- Cultivating a quality culture can be challenging, as it involves deep changes in organisational mindset and practices;

- Maintaining a high level of test automation and quality culture requires continuous investment of resources.

Practices for Maintaining TAL5

- Expand test automation to cover all of the organisation's products and services;

- Utilise advanced observability tools to gain detailed insights into the quality of software in production;

- Establish continuous improvement processes, including regular reviews of practices and results;

- Promote software quality education and training throughout the organisation.

Final Considerations

As we come to the end of this eBook on "Test Automation Levels", we hope you have gained valuable insights into the test automation journey, from the early stages to the pinnacle of quality culture. Test automation plays a crucial role in improving software quality, speeding up the development process and customer satisfaction.

Our journey took us through the five levels of test automation maturity:

TAL1 - Intuitive
IThe starting point where automation is not yet well defined.

TAL2 - Training
The transition stage with a focus on automated testing.

TAL3 - High Coverage
Achieving comprehensive coverage of automated tests.

TAL4 - Consistent Flow
Complete integration of test automation into the agile development cycle.

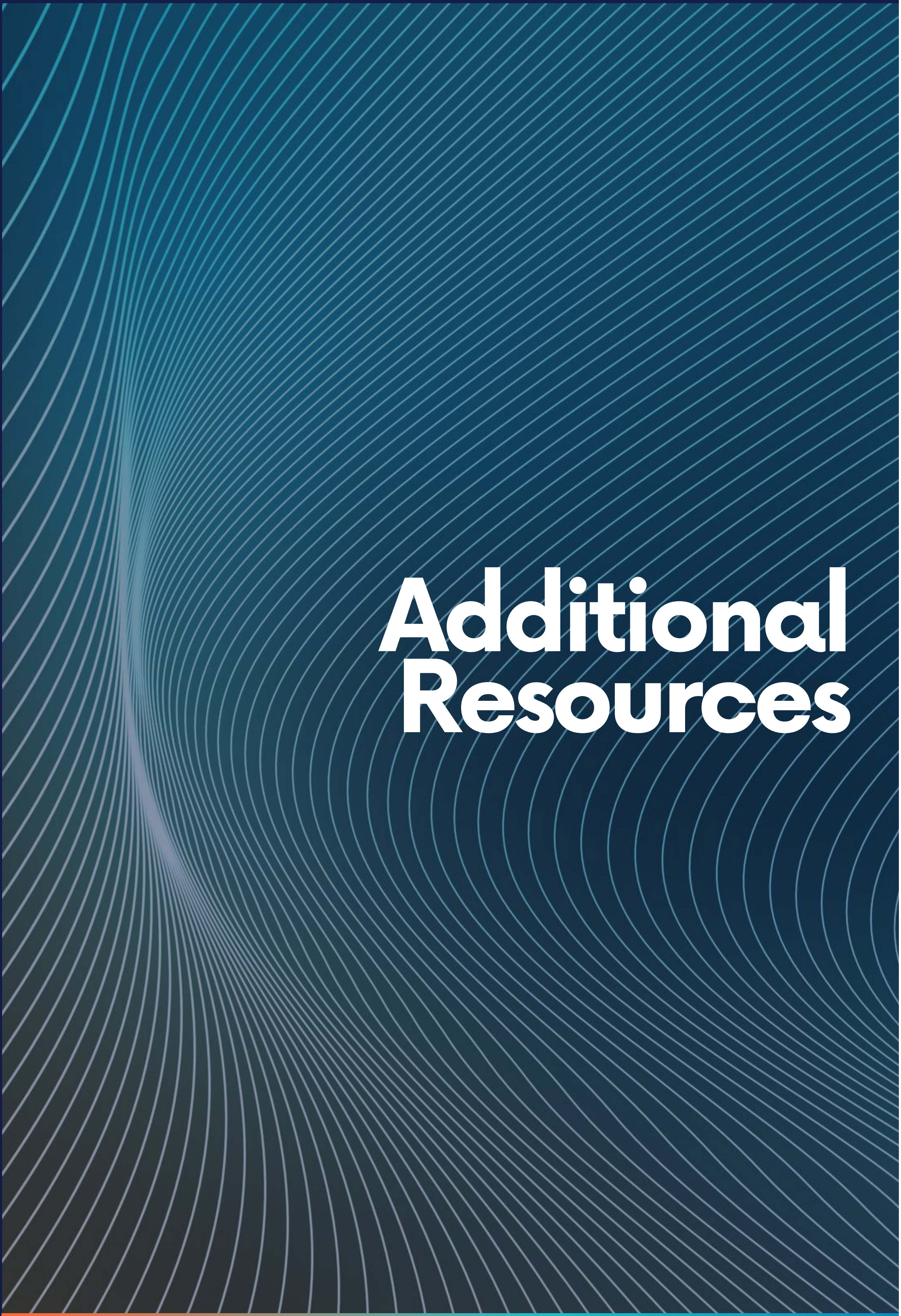
TAL5 - Quality Culture
The pinnacle of quality excellence, where the organisational culture is centred on quality.

Each level has its own challenges, benefits and specific practices. Regardless of the level your team or organisation is at, remember that test automation is a continuous journey.

Throughout this eBook, we discuss the tangible and intangible benefits of test automation, explore the considerations when choosing tools and frameworks, and provide insights into how to overcome common challenges at each stage.

To continue improving your test automation practice, encourage continuous learning, collaboration between teams and strategic alignment with the organisation's objectives. As you progress along the test automation journey, you'll see improvements in software quality, greater team efficiency and increased customer satisfaction.

We wish you and your team every success in your quest for excellence in software quality and agile development.



Additional Resources

In this section, we provide a list of additional resources you can explore to deepen your knowledge of test automation and improve your skills:

Books

"Selenium WebDriver 3 Practical Guide" by Satya Avasarala.

"Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble e David Farley.

"Python Testing with pytest" by Brian Okken.

"Agile Testing: A Practical Guide for Testers and Agile Teams" by Lisa Crispin e Janet Gregory.

"Test-Driven Development: By Example" by Kent Beck.

“Robot Framework: Automação versátil e consistente para testes” by Thomas Daniel Vieira.

Blogs and Websites

Ministry of Testing: A resource-rich online community about testing and test automation.

SeleniumHQ: The official Selenium WebDriver website with detailed documentation and tutorials.

Test Automation University: A free online learning platform with courses on test automation.

Martin Fowler's Blog: Valuable articles and insights into development practices, including automated testing.

Medium DBServices: Articles on Test Automation, Management, DevOps

Online Communities

Stack Overflow: A question and answer forum where you can find solutions to specific test automation problems.

LinkedIn Groups: There are groups dedicated to software testing and test automation where you can join discussions and gain valuable insights.

Reddit - r/QualityAssurance: An online community where quality professionals share experiences and knowledge.

Online Courses

Test Automation University: Offers a variety of free courses on test automation.

Coursera: Courses on test automation, software development and software quality.

edX: Courses from renowned universities and institutions on testing and software quality.

QA Academy: Video lessons on test automation frameworks available on YouTube from DBServices.

Open Masterclass: Monthly meetings with the DBServices consulting team with classes on Test Automation, DevOps and Software Engineering.

About DBServices and Author

About DBServices Portugal

At DBServices, we help to design, develop and support digital platforms for large and mid-sized companies in ITC, Banking, Retail, Government, Health, and other industries in Europe and Americas. Among our clients we can mention Worten, Sonae, Edenred, Natixis, Farfetch, Efacec, Segurança Social de Portugal and McDonald's.

We aim to be strategic partners for our clients when they need to improve their software development lifecycle or engineering productivity through Automation, DevOps and Agile.

We know that a resilient, robust and scalable platform, combined with a lean and agile software development pipeline, can promote the agility, innovation, quality and low costs required by our clients in a competitive and disruptive digital economy.

About the author

Thiago Boschese is a Tech and Business Development Consultant at DBServices. He specialises in Business Management and Digital Products and has led high-impact initiatives in large companies such as Embraer, Sanofi, Santander, Carrefour, Klabin, Centauro, Tok&Stok, Sonae MC and others.

November | 2023

DBServices

hello@dbservices.pt | +351 211 582 275