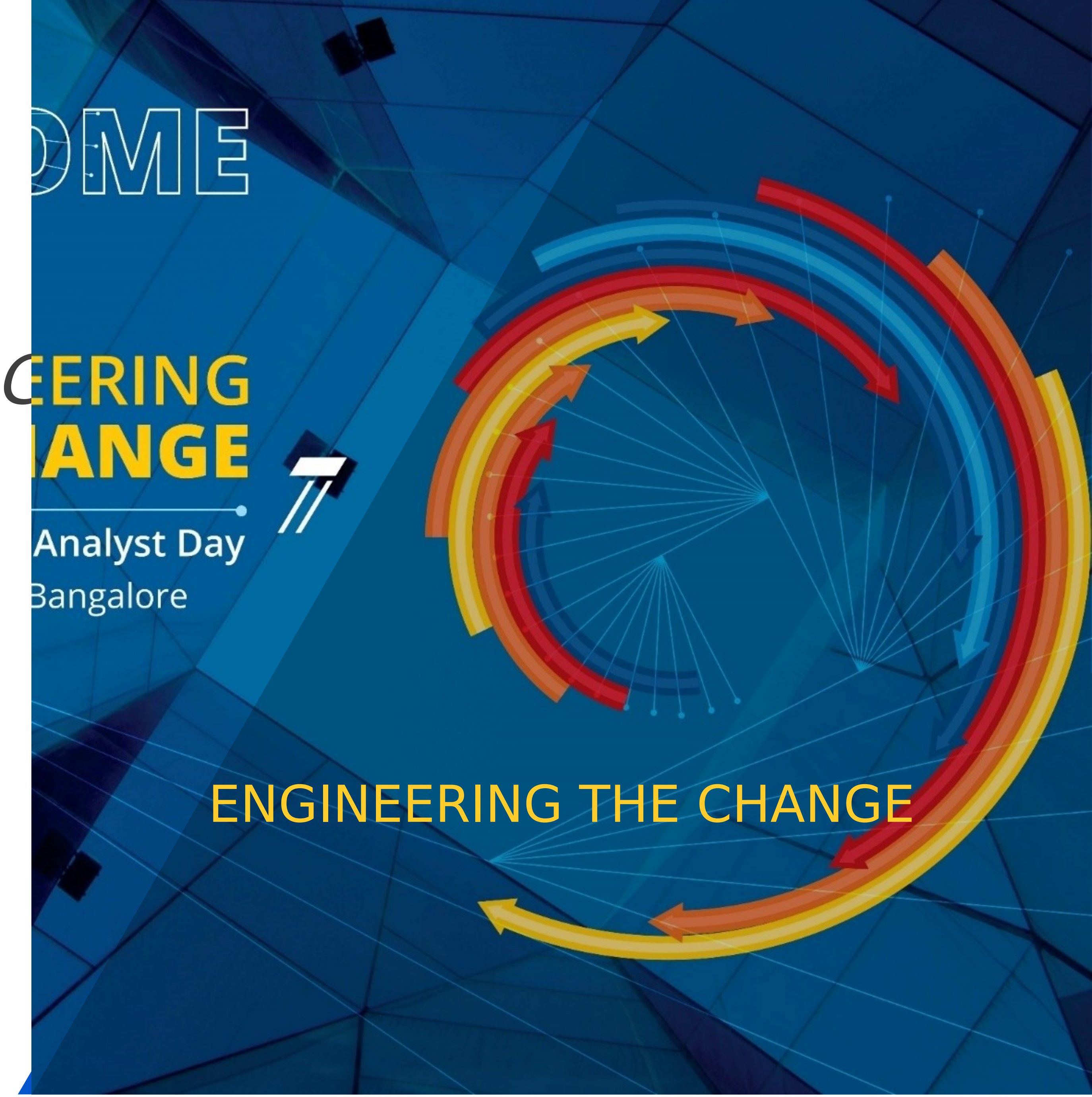
Manual Testing / TDLC

L&T Technology Services



ENGINEERING THE CHANGE

Date

Contents

AUTOMATION TESTING

1 *What is Test Automation*

2 ***Why Test Automation***

3 ***Benefits & Challenges***

4 ***Manual Testing Vs Test Automation***

5 ***When to Automate***

6 *What to Automate*

7 *A typical Test Automation Process*

8 *Test Automation Tools*

9 *Advance Automation Topics – AI, Bots, CI/CD/Devops*

What is Test Automation

Software Test automation makes use of specialized tools to control the execution of tests and compares the actual results against the expected result. Usually, regression tests, which are repetitive actions, are automated.
Testing Tools not only helps us to perform regression tests but also helps us to automate data set up generation, product installation, GUI interaction, defect logging, etc. Automation tools are used for both Functional and Non-Functional testing.

Test automation is expected to help run many test cases consistently and repeatedly on different versions of the SUT and/or environments. But test automation is more than a mechanism for running a test suite without human interaction. It involves a process of designing the test-ware, including:

- Software
- Documentation
- Test cases
- Test environments
- Test data

Why Test Automation

Objectives of test automation include:

- Improving test efficiency
- Providing wider function coverage
- Reducing the total test cost
- Performing tests that manual testers cannot
- Shortening the test execution period
- Increasing the test frequency/reducing the time required for test cycles

Benefits & Challenges

Advantages of test automation include:

- More tests can be run per build
- The possibility to create tests that cannot be done manually (real-time, remote, parallel tests)
- Tests can be more complex
- Tests run faster
- Tests are less subject to operator error
- More effective and efficient use of testing resources
- Quicker feedback regarding software quality
- Improved system reliability(e.g. repeatability, consistency)
- Improved consistency of tests

Disadvantages of test automation include:

- Additional costs are involved
- Initial investment to setup TAS
- Requires additional technologies
- Team needs to have development and automation skills
- On-going TAS maintenance requirement
- Can distract from testing objectives, e.g. focusing on automating tests cases at the expense of executing tests
- Tests can become more complex
- Additional errors may be introduced by automation

Limitations of test automation include:

- Not all manual tests can be automated
- The automation can only check machine-interpretable results
- The automation can only check actual results that can be verified by an automated test
- Not a replacement for exploratory testing

Manual Testing Vs Test Automation

Software testing is a huge domain, but it can be broadly categorized into two areas: manual testing and automated testing.

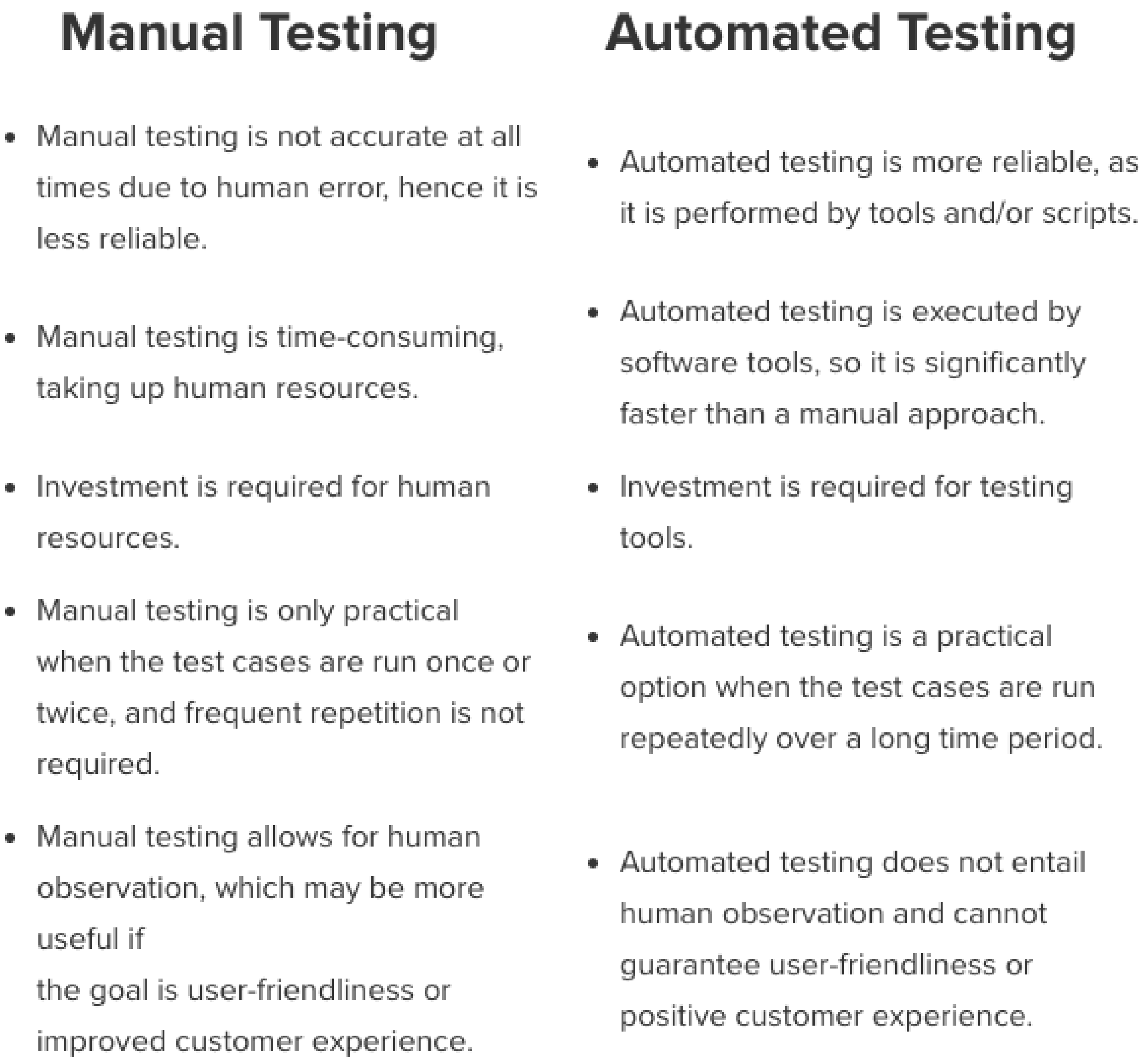
Both manual and automated testing offer benefits and disadvantages. It’s worth knowing the difference, and when to use one or the other for best results.

In manual testing (as the name suggests), test cases are executed manually (by a human, that is) without any support from tools or scripts. But with automated testing, test cases are executed with the assistance of tools, scripts, and software.

Testing is an integral part of any successful software project. The type of testing (manual or automated) depends on various factors, including project requirements, budget, timeline, expertise, and suitability. Three vital factors of any project are of course time, cost, and quality - the goal of any successful project is to reduce the cost and time required to complete it successfully while maintaining quality output. When it comes to testing, one type may accomplish this goal better than the other.

Manual vs. Automated Testing: the Pros and Cons
Manual testing and automated testing cover two vast areas. Within each category, specific testing methods are available, such as black box testing, white box testing, integration testing, system testing, performance testing, and load testing. Some of these methods are better suited to manual testing, and some are best performed through automation. Here’s a brief comparison of each type, along with some pros and cons:

Manual Testing Vs Test Automation (contd.)



Manual Testing

- Manual testing is not accurate at all times due to human error, hence it is less reliable.
- Manual testing is time-consuming, taking up human resources.
- Investment is required for human resources.
- Manual testing is only practical when the test cases are run once or twice, and frequent repetition is not required.
- Manual testing allows for human observation, which may be more useful if
  the goal is user-friendliness or improved customer experience.

Automated Testing

- Automated testing is more reliable, as it is performed by tools and/or scripts.
- Automated testing is executed by software tools, so it is significantly faster than a manual approach.
- Investment is required for testing tools.
- Automated testing is a practical option when the test cases are run repeatedly over a long time period.
- Automated testing does not entail human observation and cannot guarantee user-friendliness or positive customer experience.

When to automate

Manual testing is best suited to the following areas/scenarios:
Exploratory Testing: This type of testing requires the tester's knowledge, experience, analytical/logical skills, creativity, and intuition. The test is characterized here by poorly written specification documentation, and/or a short time for execution. We need the human skills to execute the testing process in this scenario.
Usability Testing: This is an area in which you need to measure how user-friendly, efficient, or convenient the software or product is for the end users. Here, human observation is the most important factor, so a manual approach is preferable.
Ad-hoc Testing: In this scenario, there is no specific approach. It is a totally unplanned method of testing where the understanding and insight of the tester is the only important factor.

Automated testing is the preferred option in the following areas/scenarios:
Regression Testing: Here, automated testing is suitable because of frequent code changes and the ability to run the regressions in a timely manner.
Load Testing: Automated testing is also the best way to complete the testing efficiently when it comes to load testing.
Repeated Execution: Testing which requires the repeated execution of a task is best automated.
Performance Testing: Similarly, testing which requires the simulation of thousands of concurrent users requires automation.

What to automate

Automation's benefits can be applied to just about any department within an enterprise; from HR to Accounts, Dev to Ops - even the mailroom. However, certain processes are more suited to automation than others. There are telltale signs to watch out for which indicate a process is primed for automation.

Medium and High Volume

Workflows vary dramatically in size. They can consist of simple processes which are composed of few steps, to processes requiring dozens, if not hundreds of items.

When we think about workflows with minimal steps or items, we should ask 'does it make business sense to automate this process?' Conversely, processes with medium and high volume items are clearly business-pivotal processes, primed for automation.

Manual Completion Requires Three or More Users

Generally, if a repeatable task involves three or more people, the likelihood is that it would be more efficient if it was automated. There is less chance of a communication breakdown, making it more secure and more accurate. Furthermore, by automating such routines, you'll free up the man hours of at least three individuals.

The Process Relies Upon Time-Sensitive Activities

Automation creates a log of when items pass through the workflow; recording when issues occur, when they are resolved and when they are actioned. By analyzing these logs, it's easy to spot where bottlenecks occur in your process. New workflow rules can be attributed to overcome these, thus ensuring deadlines are consistently met.

A Process Impacts Upon Various Systems

If a workflow item requires the aid of, or touches upon, various other systems and tools in the tech-stack, then guess what? Yep, it's ready for automation. Rules can applied which grant and restrict permissions to agents, enabling the timely completion of the process as well as ensuring security protocols are adhered to.

Transparent Processes Require Automation

Remaining compliant and providing full audit trails is crucial. If something goes wrong with a manual process, it's nearly impossible to provide a full accurate trail of exactly what happened, meaning the failure can't be properly identified. With automation, you have no such worries. A dedicated automation solution provides

A typical Test Automation Process

Determine the scope
Any process starts with the definition. Therefore, before implementing test automation, you should **determine the automation scope**. When starting tests development, a QA engineer should first define the order according to the tests’ priority rate.

Prepare to automate
The following points need to be considered while preparation for automation:

- Automated tests should cover the most stable part of the functionality and the one that is tested for about 3-4 times per week.
- As a rule, smoke tests (or other regression tests) are chosen for that very purpose.

Select tools for automation
As soon as the scope is defined, a QA engineer should **select test automation tools**. The tested interfaces define the package of applied tools. Different types of interfaces presuppose different tools’ range.
There is no any one-size-fits-all solution. The choice of a test automation tool will depend on the technology the software is built on. For example, QTP does not support Informatica. That means the tool cannot be used for the software. We prefer the most reliable and proven solutions: Selenium WebDriver, Coded UI, Ranorex, TestStack.White, Appium, Xamarin, and many more.

Develop the framework
The framework is the basis for further automated tests’ development. It provides an opportunity to optimize test development efforts by re-using the code. You can utilize any of the ready-to-use frameworks presented on the market, like the Robot framework for Selenium.
The usefulness of frameworks is hard to underestimate. Thanks to frameworks, it’s possible to maintain consistency of testing and improve re-usability. You can also count on minimizing code usage and improving

A typical Test Automation Process (contd.)

Configure the environment

All the tests run in **the environment**, which is to be **well-configured**. Upon this step, you should create and support the environment to successfully run automated tests and store the results.

Test automation will require **test data**, which means you are to **prepare** the set of files and test accounts beforehand. Otherwise, you tackle risks that may damage the process and provide you with irrelevant test results.

Start to automate

Eventually, when all the preliminary preparations are done, testers can get down to **automated test development**. A regular process of providing new automated tests includes the following points:

- Selection of the manual test case according to the stated priorities
- Code writing for the automated test
- Adding the automated test to the debug test execution
- Adding the automated test to the test execution for newly created builds.

As well as analysis, the **tests need support**, which presupposes the process of updating automated tests along with the updates of the interface and/or business logic.

When the tests are launched, they should be **monitored**. You cannot let them go along without tracking the process. While monitoring the automated test, remember to take into consideration the following aspects:

- Automated test coverage, cost per test
- Useful vs. irrelevant results after test execution
- Cost per test
- The scope of support in comparison with the number of executed tests
- Economic effect (ROI, return on investment).

Test Automation Tools

Here, is a simple process to determine the best tool for your project needs

- Identify the tests that need to be automated
- Research and analyze the automation tools that meet your automation needs and budget
- Based on the requirements, shortlist two most suitable tools
- Do a pilot for two best tools and select the better one
- Discuss the chosen automation tools with other stakeholders, explain the choice, and get their approval
- Proceed to test automation

There is no best tool in the market. You need to find the best tool for your test automation project goals.

Advance Automation Topics – AI, Bots, CI/CD/Devops

AI

The terms *Artificial Intelligence* and *Automation* are often used interchangeably. They're are associated with software or physical robots and other machines that allow us to operate more efficiently and effectively — whether it's a mechanical construct piecing together a car or sending a follow-up email the day after your customer hasn't completed his order.

However, the are pretty big differences between complexity level of both systems. Automation is basically making a hardware or software that is capable of doing things automatically — without human intervention. Artificial Intelligence, however, is a science and engineering of making intelligent machines (according to John McCarthy, person who coined this term). AI is all about trying to make machines or software mimic, and eventually supersede human behaviour and intelligence.

Bots or Robotic Process Automation (RPA)

The basic difference between automation and robotics can be seen in their definitions:

Automation — Automation means using computer software, machines or other technology to carry out a task which would otherwise be done by a human worker. There are many types of automation, ranging from the fully mechanical to the fully virtual, and from the very simple to the mind-blowingly complex.

Robotics — Robotics is a branch of engineering which incorporates multiple disciplines to design, build, program and use robotic machines.

There are obviously crossovers between the two. Robots are used to automate some physical tasks, such as in manufacturing. However, many types of automation have nothing to do with physical robots. Also, many branches of robotics have nothing to do with automation.

Advance Automation Topics – AI, Bots, CI/CD/Devops (contd.)

Continuous integration (CI) is a software engineering practice where members of a team integrate their work with increasing frequency. In keeping with CI practice, teams strive to integrate at least daily and even hourly, approaching integration that occurs "continuous-ly."
Historically, integration has been a costly engineering activity. So, to avoid thrash, CI emphasizes automation tools that drive build and test, ultimately focusing on achieving a software-defined life cycle. When CI is successful, build and integration effort drops, and teams can detect integration errors as quickly as practical.

Continuous delivery (CD) is to packaging and deployment what CI is to build and test. Teams practicing CD can build, configure, and package software and orchestrate its deployment in such a way that it can be released to production in a software-defined manner (low cost, high automation) at any time.
High-functioning CI/CD practices directly facilitate agile development because software change reaches production more frequently. As a result, customers have more opportunities to experience and provide feedback on change.

DevOps focuses on limitations of culture and roles as agile development does process. The intention of DevOps is to avoid the negative impact that overspecialization and stovepiping roles in an organization have on preventing rapid or even effective response to production issues. DevOps organizations break down the barriers between Operations and Engineering by cross-training each team in the other's skills. This approach improves everyone's ability to appreciate and participate in each other's tasks and leads to more high-quality collaboration and more frequent communication.

THANK YOU