

Lecture 5

Automation Testing

What is a Automated Testing?

Automated Testing means using special software for tasks that people usually do when checking and testing a software product.

Nowadays, many software projects use automation testing from start to end, especially in agile and DevOps methods.

This means the engineering_team runs tests automatically with the help of software tools_.

It will help to keep the testing team to make the process faster.

Continuous delivery (CD) and quickly sends the new code to users.

Automated testing is important for this because it converts the manual steps into automation.

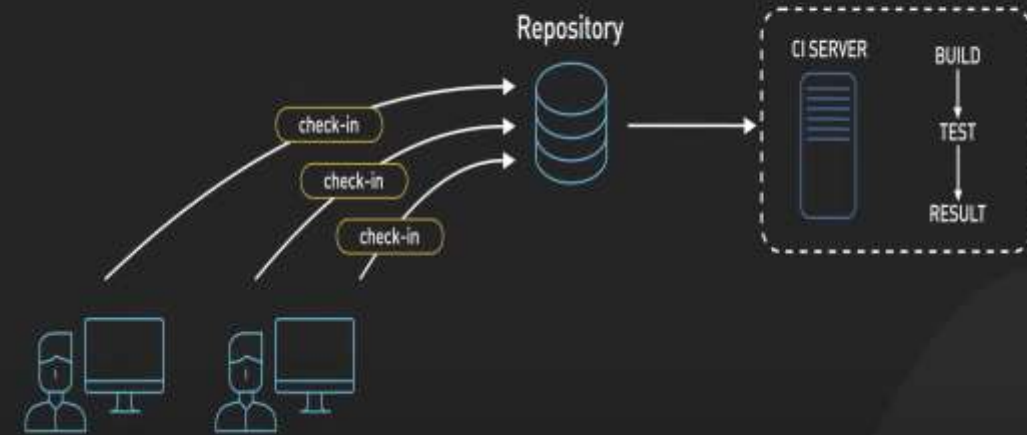
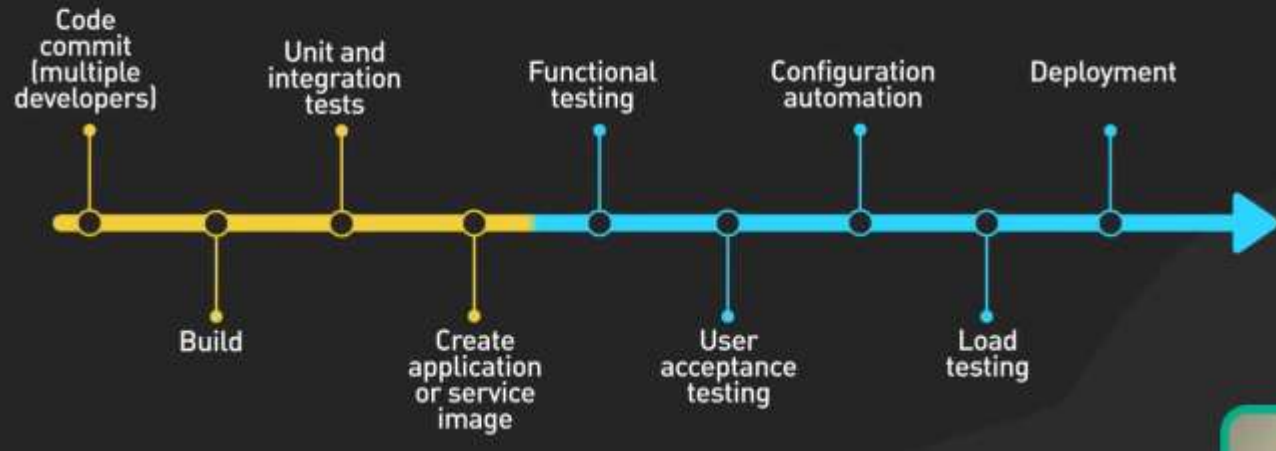
Continuous integration (CI) checks the new code changes to prevent issues.

CD gets after CI done everything well.

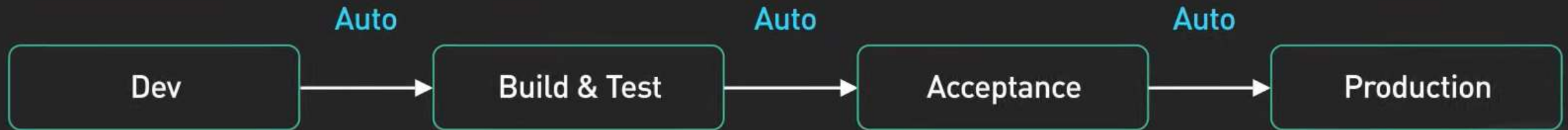
Automated testing, CI & CD will together prove that the new code is error-free and ready for deployment quickly for the project purpose.

○ Continuous integration

○ Continuous delivery



Continuous Deployment



What is a Automated Testing?

- It is used to automate the testing tasks that are difficult to perform manually.
- Automation tests can be run at any time of the day as they use scripted sequences to examine the software.
- Automation tests can also enter test data compare the expected result with the actual result and generate detailed test reports.
- The goal of automation tests is to reduce the number of test cases to be executed manually but not to eliminate manual testing.
- It is possible to record the test suit and replay it when required.

Why Transform From Manual to Automated Testing?

- In 1994, an aircraft completing its routine flight crashed just before landing. This was due to a bug or defect in their software. The testers didn't even care about the final testing, so this accident happened. In such high-priority cases, accurate automation testing becomes critical to replace a few manual tests. Below are some of the reasons for using automation testing:
- **Quality Assurance:** Manual testing is a boring task that and at the same time error-prone. Thus, using automation testing improves the quality of the software under test as more test coverage can be achieved.
- **Error or Bug-free Software:** Automation testing is more efficient for detecting bugs in comparison to manual testing.
- **No Human Intervention:** Manual testing requires huge manpower in comparison to automation testing which requires no human intervention and the test cases can be executed unattended.
- **Increased test coverage:** Automation testing ensures more test coverage in comparison to manual testing where it is not possible to achieve 100% test coverage.
- **Testing can be done frequently:** Automation testing means that the testing can be done frequently thus improving the overall quality of the software under test.

What Kinds of Software Tests Should Be Automated First?

1. End-to-End tests

End-to-end testing is a type of software testing used to test whether the flow of software from the initial stage to the final stage is behaving as expected. The purpose of end-to-end testing is to identify system dependencies and to make sure that the data integrity is maintained between various system components and systems. End-to-end testing: End-to-end testing, also known as end-to-end functional testing, is a type of testing that validates the flow of a system from start to finish.

2. Unit tests

Unit testing is automated and is run each time the code is changed to ensure that new code does not break existing functionality. Unit tests are designed to validate the smallest possible unit of code, such as a function or a method, and test it in isolation from the rest of the system.

3. Integration tests

Integration testing is the process of testing the interface between two software units or modules. It focuses on determining the correctness of the interface. The purpose of integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit-tested, integration testing is performed.

What Kinds of Software Tests Should Be Automated First?

4. Performance tests

Performance Testing is a type of software testing that ensures software applications perform properly under their expected workload. It is a testing technique carried out to determine system performance in terms of sensitivity, reactivity, and stability under a particular workload.

What Kinds of Software Tests Should Be Done Manually?

Automating tests will save the time and speed up the productivity, but not every test needs automation. sometime the investment in creating automated tests the benefits of manual testing.

1. Exploratory testing

Exploratory Testing is a type of software testing in which the tester is free to select any possible methodology to test the software. It is an unscripted approach to software testing. In exploratory testing, software developers use their learning, knowledge, skills, and abilities to test the software developed by themselves. Exploratory testing checks the functionality and operations of the software as well as identify the functional and technical faults in it.

2. Visual Regression Testing

Visual regression testing is a software development technique for identifying unintentional visual modifications made to the user interface (UI) of a web application. To find any differences, screenshots of the user interface (UI) of the application are compared before and after code changes. Developers can guarantee that changes to the coding don't bring about unexpected visual changes that could degrade the user experience by automating this process.

Difference Between Manual Testing vs Automated Testing

Parameters	Manual Testing	Automated Testing
Reliability	<u>Manual testing</u> is not accurate at all times due to human error, thus it is less reliable.	Since it is performed by third-party tools and/or scripts, therefore it is more reliable.
Investment	Heavy investment in human resources.	Investment in tools rather than human resources.
Time efficiency	Manual testing is time-consuming due to human intervention where test cases are generated manually.	<u>Automation testing</u> is time-saving as due to the use of the tools the execution is faster in comparison to manual testing.
Programming knowledge	There is no need to have programming knowledge to write the test cases.	It is important to have programming knowledge to write test cases.
Regression testing	There is a possibility that the test cases executed the first time will not be able to catch the regression bugs due to the frequently changing requirements.	When there are changes in the code, regression testing is done to catch the bugs due to changes in the code.

Automation Testing Types

Unit testing: is a phase in software testing to test the smallest piece of code known as a unit that can be logically isolated from the code. It is carried out during the development of the application.

Integration testing: is a phase in software testing in which individual software components are combined and tested as a group. It is carried out to check the compatibility of the component with the specified functional requirements.

Smoke testing: is a type of software testing that determines whether the built software is stable or not. It is the preliminary check of the software before its release in the market.

Performance testing: is a type of software testing that is carried out to determine how the system performs in terms of stability and responsiveness under a particular load.

Regression testing: is a type of software testing that confirms that previously developed software still works fine after the change and that the change has not adversely affected existing features.

Automation Testing Types

Security testing: Security testing is a type of software testing that uncovers the risks, and vulnerabilities in the security mechanism of the software application. It helps an organization to identify the loopholes in the security mechanism and take corrective measures to rectify the security gaps.

Acceptance testing: Acceptance testing is the last phase of software testing that is performed after the system testing. It helps to determine to what degree the application meets end users' approval.

API testing: API testing is a type of software testing that validates the Application Programming Interface(API) and checks the functionality, security, and reliability of the programming interface.

UI Testing: UI testing is a type of software testing that helps testers ensure that all the fields, buttons, and other items on the screen function as desired.

Automation Testing Process

Linear framework: This is the most basic form of framework and is also known as the record and playback framework. In this testers create and execute the test scripts for each test case. It is mostly suitable for small teams that don't have a lot of test automation experience.

Modular-Based Framework: This framework organizes each test case into small individual units known as modules each module is independent of the other, having different scenarios but all modules are handled by a single master script. This approach requires a lot of pre-planning and is best suited for testers who have experience with test automation.

Library Architecture Framework: This framework is the expansion of a modular-based framework with few differences. Here, the task is grouped within the test script into functions according to a common objective. These functions are stored in the library so that they can be accessed quickly when needed. This framework allows for greater flexibility and reusability but creating scripts takes a lot of time so testers with experience in automation testing can benefit from this framework.

Test Automation Frameworks

- 1. Test Tool Selection:** There will be some criteria for the Selection of the tool. The majority of the criteria include: Do we have skilled resources to allocate for automation tasks, Budget constraints, and Do the tool satisfies our needs?
- 2. Define Scope of Automation:** This includes a few basic points such as the Framework should support Automation Scripts, Less Maintenance must be there, High Return on Investment, Not many complex Test Cases
- 3. Planning, Design, and Development:** For this, we need to Install particular frameworks or libraries, and start designing and developing the test cases such as NUnit , JUnit , QUnit , or required Software Automation Tools.
- 4. Test Execution:** Final Execution of test cases will take place in this phase and it depends on Language to Language for .NET, we'll be using NUnit, for Java, we'll be using JUnit, for JavaScript, we'll be using QUnit or Jasmine, etc.
- 5. Maintenance:** Creation of Reports generated after Tests and that should be documented to refer to that in the future for the next iterations.

Criteria to Select Automation Tool

- **Ease of use:** Some tools have a steep learning curve, they may require users to learn a completely new scripting language to create test cases and some may require users to maintain a costly and large test infrastructure to run the test cases.
- **Support for multiple browsers:** Cross-browser testing is vital for acceptance testing. Users must check how easy it is to run the tests on different browsers that the application supports.
- **Flexibility:** No single tool framework can support all types of testing, so it is advisable to carefully observe what all tool offers and then decide.
- **Ease of analysis:** Not all tools provide the same sort of analysis. Some tools have a nice dashboard feature that shows all the statistics of the test like which test failed and which test passed. On the other hand, there can be some tools that will first request users to generate and download the test analysis report thus, not very user-friendly. It depends entirely on the tester, project requirement, and budget to decide which tool to use.

Criteria to Select Automation Tool

- **Cost of tool:** Some tools are free and some are commercial tools but many other factors need to be considered before deciding whether to use free or paid tools. If a tool takes a lot of time to develop test cases and it is a business-critical process that is at stake then it is better to use a paid tool that can generate test cases easily and at a faster rate.
- **Availability of support:** Free tools mostly provide community support on the other hand commercial tools provide customer support, and training material like tutorials, videos, etc. Thus, it is very important to keep in mind the complexity of the tests before selecting the appropriate tool.

Best Practices for Test Automation

Plan self-contained test cases: It is important to ensure that the test is clearly defined and well-written. The test cases should be self-contained and easy to understand.

Plan the order to execute tests: Planning the test in the manner that the one test creates the state for the second test can be beneficial as it can help to run test cases in order one after another.

Use tools with automatic scheduling: If possible use tools that can schedule testing automatically according to a schedule.

Set up an alarm for test failure: If possible select a tool that can raise an alarm when a test failure occurs. Then a decision needs to be made whether to continue with the test or abort it.

Reassess test plans as the app develops and changes: It is important to continuously reassess the test plan as there is no point in wasting resources in testing the legacy features in the application under test.

Popular Automation Tools

Selenium: is an automated testing tool that is used for [Regression testing](#) and provides a playback and recording facility. It can be used with frameworks like [JUnit](#) and [Test-NG](#). It provides a single interface and lets users write test cases in languages like Ruby, Java, Python, etc.

QTP: Quick Test Professional (QTP) is an automated functional testing tool to test both web and desktop applications. It is based on the VB scripting language and it provides functional and regression test automation for software applications.

Sikuli: It is a GUI-based test automation tool that is used for interacting with elements of web pages. It is used to search and automate graphical user interfaces using screenshots.

Appium: is an open-source test automation framework that allows QAs to conduct automated app testing on different platforms like iOS, Android, and Windows SDK.

Jmeter: [Apache JMeter](#) is an open-source Java application that is used to load test the functional behavior of the application and measure the performance.

Advantages of Automation Testing

- **Simplifies Test Case Execution:** Automation testing can be left virtually unattended and thus it allows monitoring of the results at the end of the process. Thus, simplifying the overall test execution and increasing the efficiency of the application.
- **Improves Reliability of Tests:** Automation testing ensures that there is equal focus on all the areas of the testing, thus ensuring the best quality end product.
- **Increases amount of test coverage:** Using automation testing, more test cases can be created and executed for the application under test. Thus, resulting in higher test coverage and the detection of more bugs. This allows for the testing of more complex applications and more features can be tested.
- **Minimizing Human Interaction:** In automation testing, everything is automated from test case creation to execution thus there are no chances for human error due to neglect. This reduces the necessity for fixing glitches in the post-release phase.
- **Saves Time and Money:** The initial investment for automation testing is on the higher side but it is cost-efficient and time-efficient in the long run. This is due to the reduction in the amount of time required for test case creation and execution which contributes to the high quality of work.
- **Earlier detection of defects:** Automation testing documents the defects, thus making it easier for the development team to fix the defect and give a faster output. The earlier the defect is identified, the more easier and cost-efficient it is to fix the defects.

Disadvantages of Automation Testing

High initial cost: Automation testing in the initial phases requires a lot of time and money investment. It requires a lot of effort for selecting the tool and designing customized software.

100% test automation is not possible: Generally, the effort is to automate all the test cases but in practical real situations not all test cases can be automated some test cases require human intervention for careful observation. There is always a human factor, i.e., it can't test everything like humans(design, usability, etc.).

Not possible to automate all testing types: It is not possible to automate tests that verify the user-friendliness of the system. Similarly, if we talk about the graphics or sound files, even their testing cannot be automated as automated tests typically use textual descriptions to verify the output.

Programming knowledge is required: Every automation testing tool uses any one of the programming languages to write test scripts. Thus, it is mandatory to have programming knowledge for automation testing.

False positives and negatives: Automation tests may sometimes fail and reflect that there is some issue in the system but there is no issue present and in some cases, it may generate false negatives if tests are designed to verify that some functionality exists and not to verify that it works as expected.

Software Quality Assurance (SQA)
