|  |  |
| --- | --- |
| **SL.NO.** | **Topics** |
| 1 | What is software testing? |
| 2 | Why Testing is important? |
| 3 | What is Agile? |
| 4 | Advantages Disadvantages of agile |
| 5 | How is software testing in Agile different from normal testing? |
| 6 | Bug Life Cycle |
| 7 | Types of Software Testing |
| 8 | What is Automation Testing? |
| 9 | Advantages of software automation testing |
| 10 | Automation Testing Tools |
| 11 | Performance Testing |
| 12 | High Level Test Execution Flow |
| 13 | Test Report Dashboard |

**What is software testing?**

**Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is[Defect](https://www.guru99.com/defect-management-process.html)free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

**Why Testing is important?**

* Helps in saving money
* Security
* Enhancing the development process
* Improve quality of the product
* Satisfaction of the customer
* Easy while adding new features
* Determining the performance of the software

**What is Agile?**

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

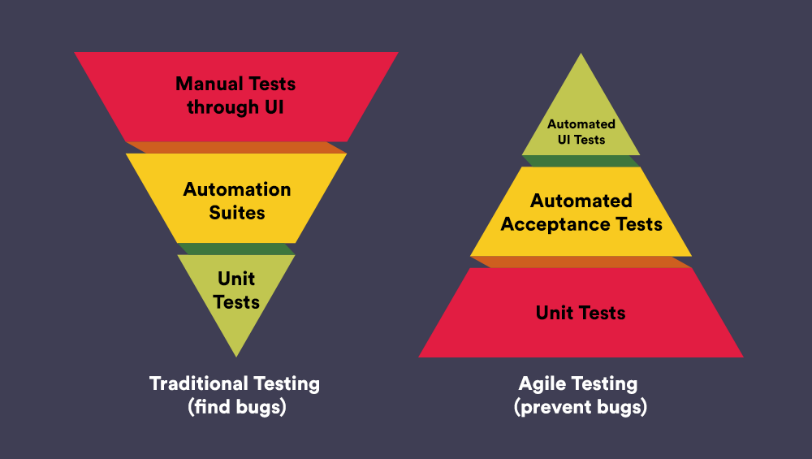
**Advantages Disadvantages of agile**

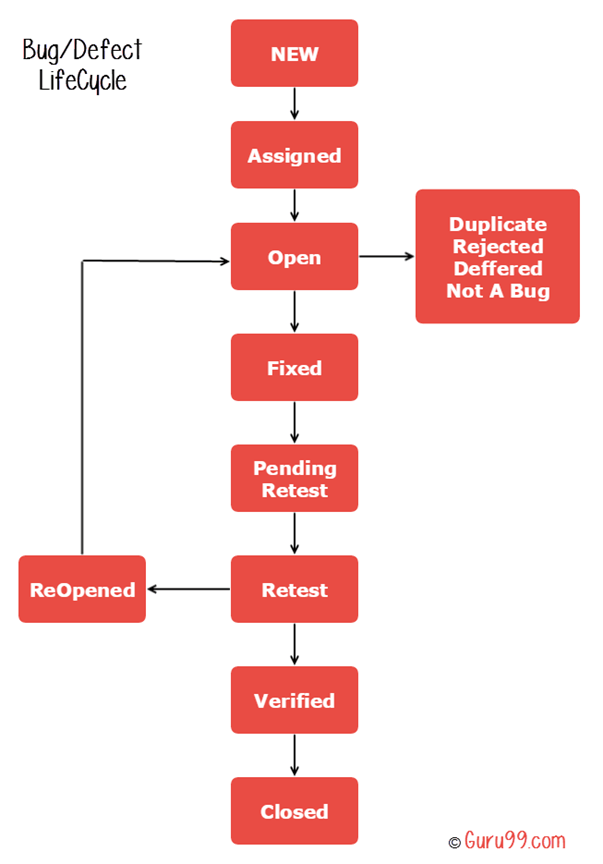
Advantages:

* In Agile methodology the delivery of software is unremitting.
* The customers are satisfied because after every Sprint working feature of the software is delivered to them.
* Customers can have a look of the working feature which fulfilled their expectations.
* If the customers have any feedback or any change in the feature then it can be accommodated in the current release of the product.
* In Agile methodology the daily interactions are required between the business people and the developers.

Disadvantages:

* In Agile methodology the documentation is less.
* Sometimes in Agile methodology the requirement is not very clear hence it’s difficult to predict the expected result.
* In few of the projects at the starting of the software development life cycle it’s difficult to estimate the actual effort required.
* Because of the ever-evolving features, there is always a risk of the ever-lasting project.
* For complex projects, the resource requirement and effort are difficult to estimate.

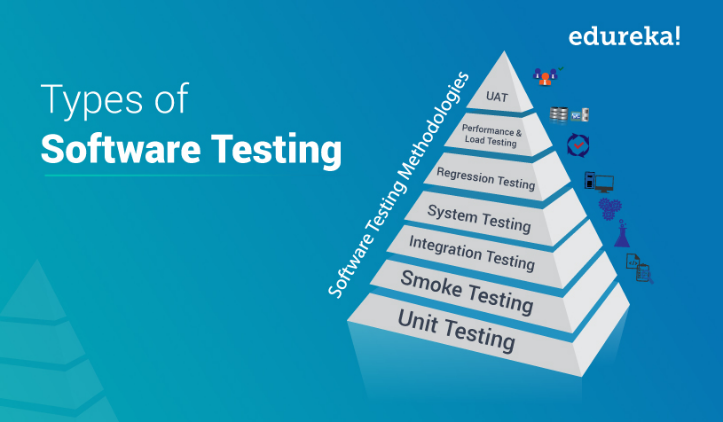
**How is software testing in Agile different from normal testing?**

**Bug Life Cycle**

Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime. It varies from organization to organization and also from project to project as it is governed by the software testing process and also depends upon the tools used.

The stages of Bug Life Cycle are shown in the figure.

**Types of Software Testing**



* Unit Testing
* Smoke Testing
* Integration Testing
* System Testing
* Regression Testing
* Performance & Load Testing
* User Acceptance Test

**What is Automation Testing?**

Automation testing is a Software testing technique to test and compare the actual outcome with the expected outcome. This can be achieved by writing test scripts or using any automation testing tool. Test automation is used to automate repetitive tasks and other testing tasks which are difficult to perform manually.

**Advantages of software automation testing**

● Quickly Determine the Stability of Your Build

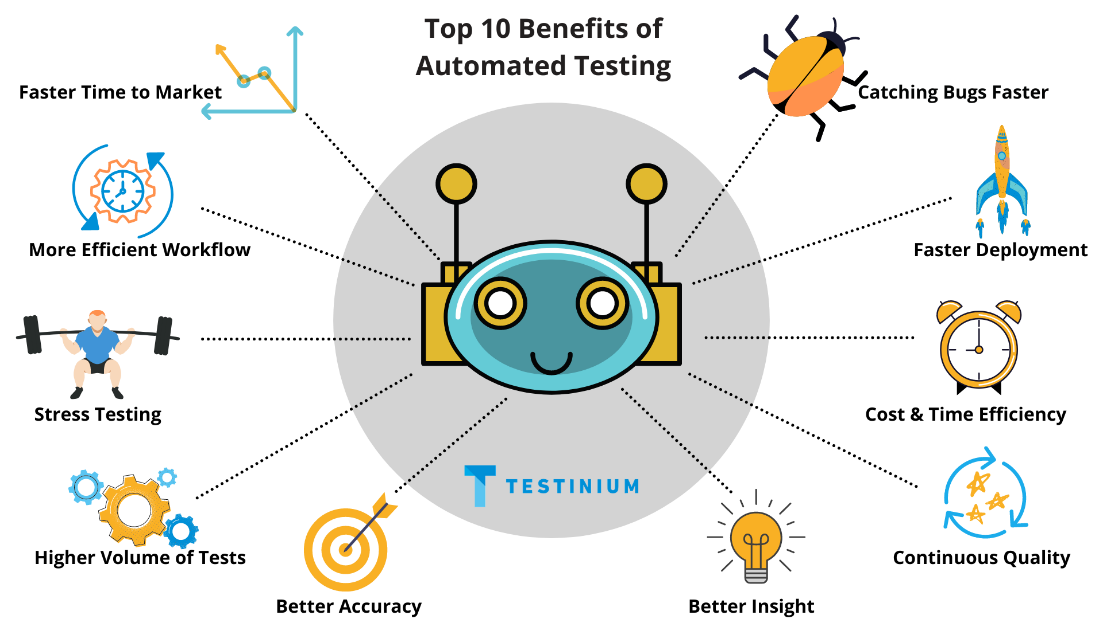
● Immediate Feedback

● Fast Development and Delivery

● Increased Productivity

● Early detection of the defect

● Higher Test Coverage



**Automation Testing Tools**

**Automation Testing tools for Functional Automation:**

* Selenium
* Appium
* Cypress
* Auto IT
* Geb
* HPE Unified Functional Testing (UFT)
* EggPlant

**Automation Testing tools for Non-Functional Automation:**

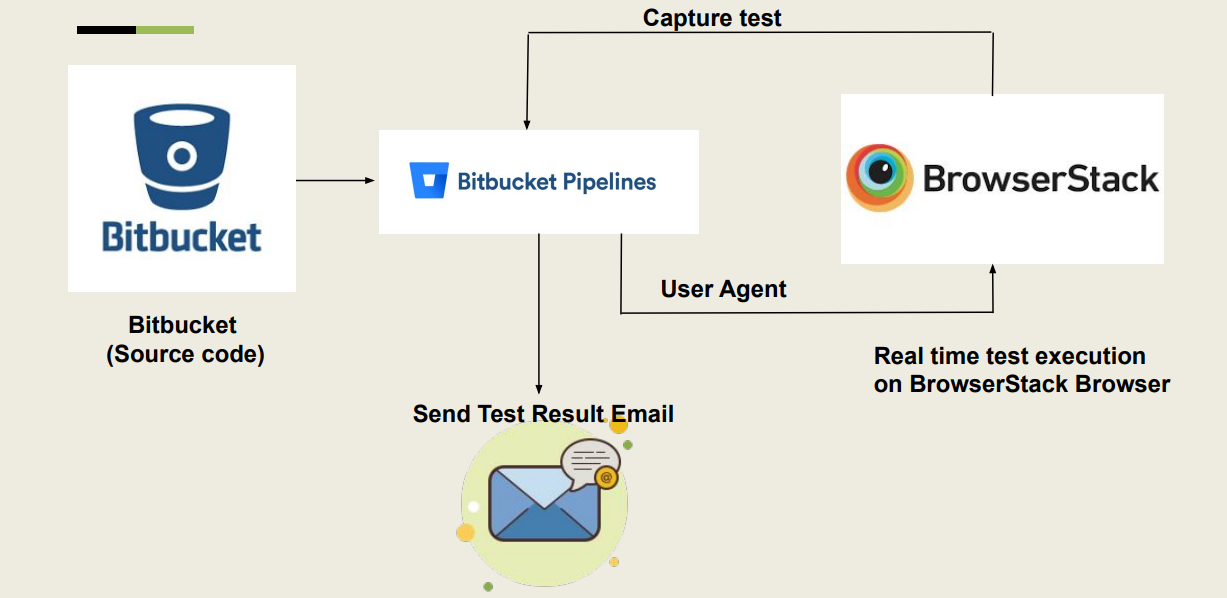
* JMeter
* LoadRunner
* Wrk

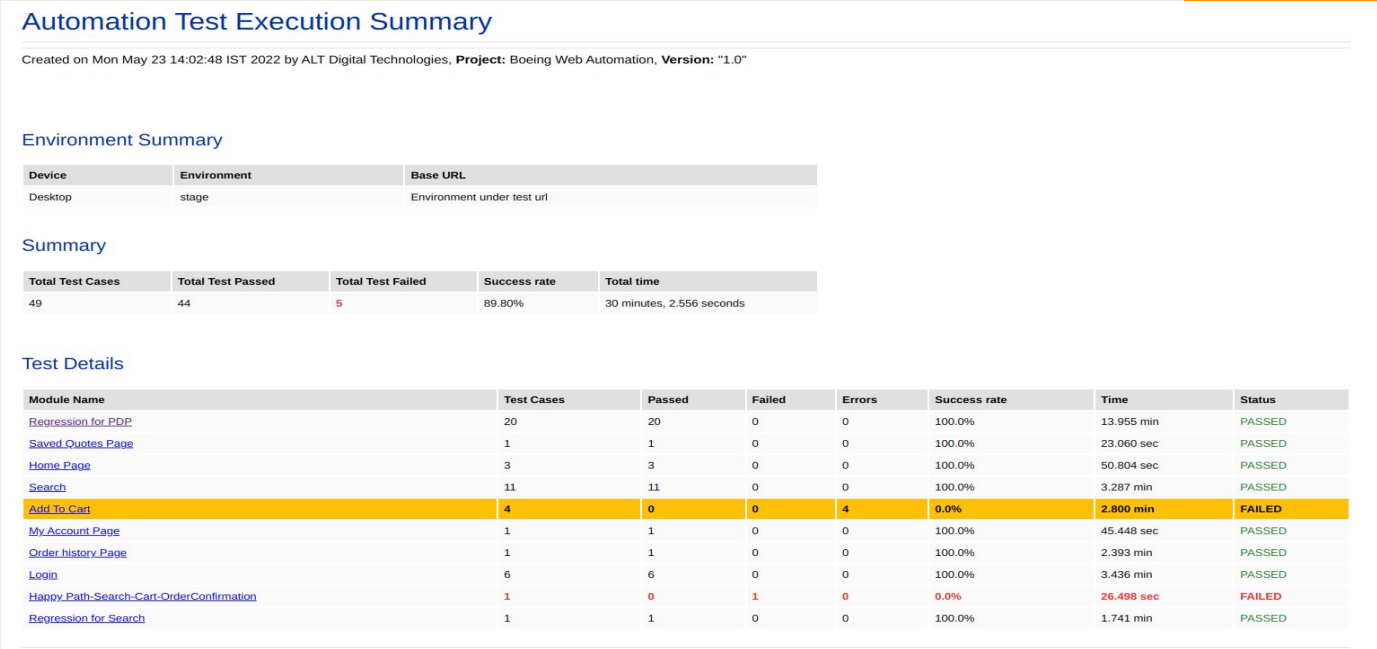
**Performance Testing**

Performance testing is a non-functional [software testing](https://www.microfocus.com/products/performance-engineering/overview) technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload. It’s a key step in ensuring software quality, but unfortunately, is often seen as an afterthought, in isolation, and to begin once functional testing is completed, and in most cases, after the code is ready to release.

The goals of [performance testing](https://www.microfocus.com/solutions/performance-testing) include evaluating application output, processing speed, data transfer velocity, network bandwidth usage, maximum concurrent users, memory utilization, workload efficiency, and command response times.

**High Level Test Execution Flow**



**Test Report Dashboard**

**Conclusion**

Software testing is an important part of the software development process. It is not a single activity that takes place after code implementation, but is part of each stage of the lifecycle. A successful test strategy will begin with consideration during requirements specification. Testing details will be fleshed through high and low level system designs, and testing will be carried out by developers and separate test groups after code implementation.

As with the other activities in the software lifecycle, testing has its own unique challenges. As software systems become more and more complex, the importance of effective, is well planned testing efforts will only increase.