What is Integration Testing? A Simple Guide on How to Perform Integration Testing

Last updated on Nov 25,2020   *28.3K Views*

Archana Choudary

Before a software is released to the market for commercial use, it undergoes rigorous testing. Software testing is classified into four different levels, namely Unit Testing, Integration Testing, System Testing and Acceptance Testing. These four levels of software testing are designed to detect errors, evaluate the correctness of behavior and performance of the software application. Through this "What is Integration Testing?" article, we will learn mainly about the second level of testing, Integration Testing. If you are new to software testing, be sure to also read the Beginners' Guide for Software Testing.

Let's take a look at topics covered in this article:

- Levels of Software Testing
- What is Integration Testing?
- Advantages of Integration Testing
- How is Integration Testing Done?
- Types of Integration Testing
- Challenges of Integration Testing

Levels of Software Testing

Software Testing *is a phase within the software development cycle in which business-critical software is verified for correctness, quality, and performance.*

There are four fundamental levels within software testing, each examining the software functionality from a unique vantage point within the development process. The four levels of software testing are as shown in the image below.
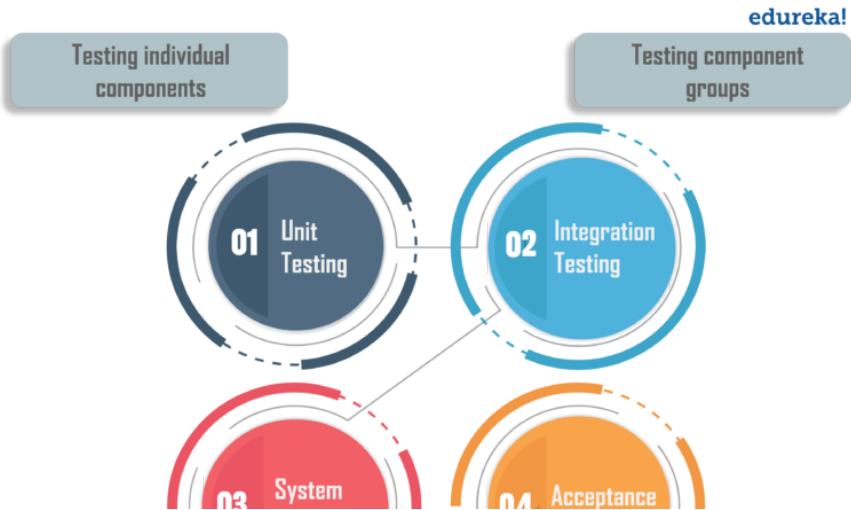
Integration Testing is a level of software testing where individual units are combined and tested to verify if they are working as they intend to when integrated. The main aim here is to test the interface between the modules.

Just unit testing is not enough for multiple reasons like:

- A module/unit is usually designed by an individual software developer whose techniques and programming logic differs from that of other programmers
- Often at the time of module development, user requirements change and these new requirements may not be unit tested. This instigates issues
- Issues like data formatting, error trapping, hardware interfaces, and third-party service interfaces are sometimes missed during unit testing

So, no matter how efficiently each module/unit is running, if they aren't properly integrated, it will affect the functionality of the software program. As a solution integration testing is implemented. This article 'What is Integration Testing?' further lists out the advantages of integration testing.

Advantages of Integration Testing

Performing integration testing offers a lot of benefits. Some of them are listed below:

- It makes sure that integrated modules work properly as intended
- The tester can start testing once the modules to be tested are available
- It detects errors related to the interface between modules
- Helps modules interact with API's and other third-party tools
- Typically covers a large volume of the system, so more efficient
- Increases the test coverage and improves the reliability of tests

Software engineers perform integration testing. Sometimes companies employ independent testers to do it for them. But, how is integration testing carried out in reality? Is it similar to other testing processes? Let's check it out the next part of this 'What is Integration Testing?' article.

How is Integration Testing Done?

The meaning of integration is quite straightforward – Combine the unit tested module one by one and test the functionality of the combined unit. Normally, integration testing is carried out after unit testing. Once all the individual units are created and tested, we start combining those tested modules and start performing the integrated testing. The main goal here is to test the interfaces between the units/modules. Here are a few simple steps that will get you started with integration testing:

Selenium Certification Training Course

Instructor-led Sessions

Assessments

Assignments

Lifetime Access

Explore Curriculum

- Prepare the test integration plan
- Decide on the type of integration testing approach
- Design test cases, test scenarios and test scripts accordingly
- Deploy the chosen modules together and get the integration tests running
- Track the defects and record the test results of tests
- Repeat the above steps until the complete system is tested

Remember that priority must be given to the integrated interface links between modules or units. As you can see, in the second

Case1: You have to test the Login Page which is developed and sent to the testing team. Login Page is dependent on Admin Page. But the Admin Page is not ready yet. To overcome this situation developers write a dummy program which acts as an Admin Page. This dummy program is Stub. Stubs are 'Called Programs'.

Case2: You have to test Admin Page but the Login Page is not ready yet. To overcome this situation developers write a dummy program which acts like the Login Page. This dummy program is Driver. Drivers are 'Calling programs'.
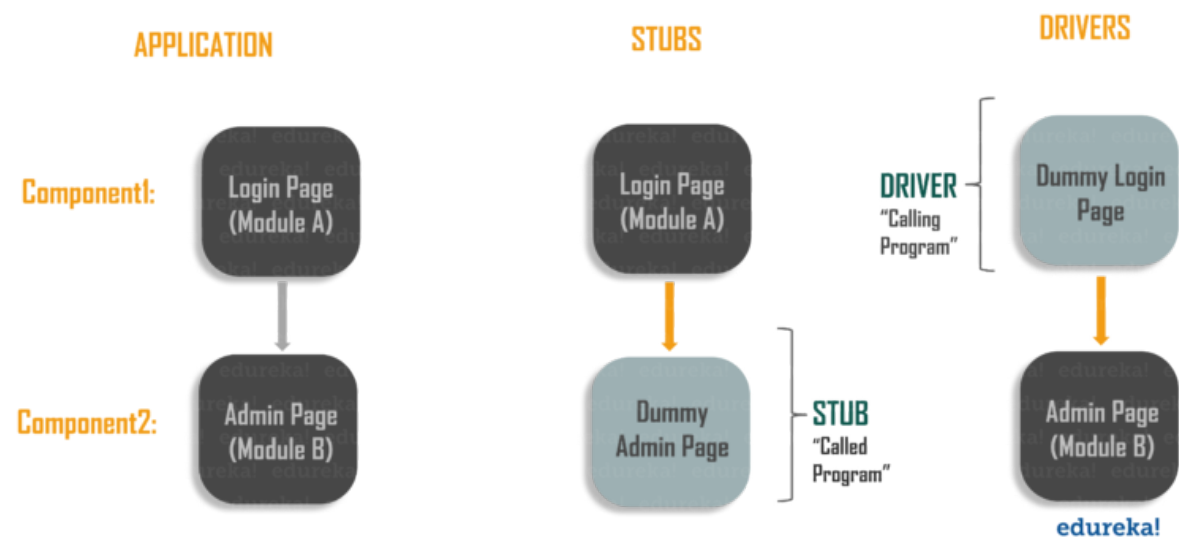
Now that you are aware of these necessary concepts, let's check out different types of integration tests. Integration testing types are categorized into different groups that are listed below:
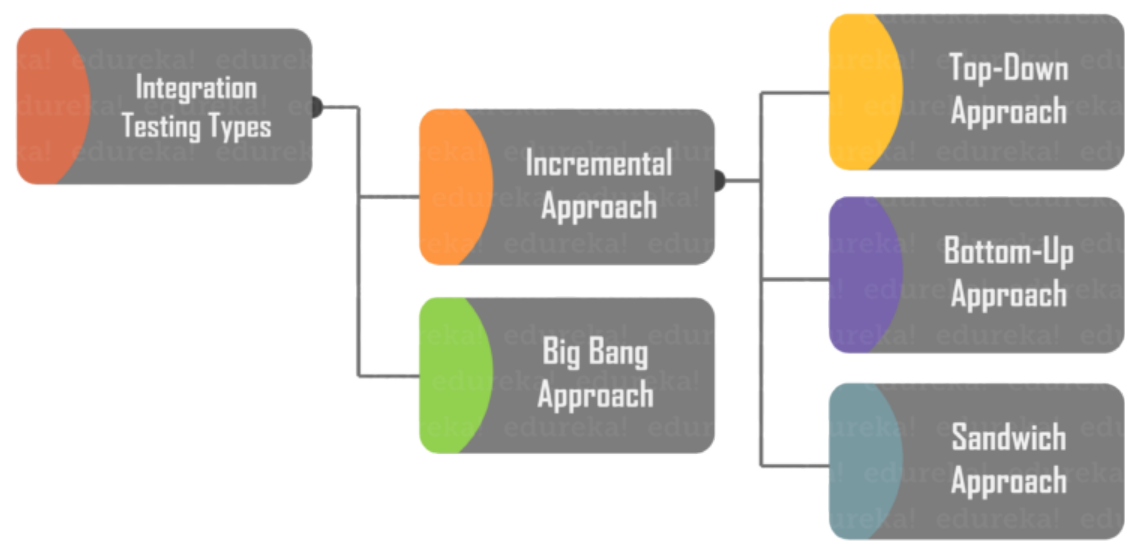
Big Bang Integration Testing

In this testing approach, once all the modules are developed and tested individually, they are integrated once and tested together at once. The only advantage of this type of testing is that it is very much suitable for smaller systems.
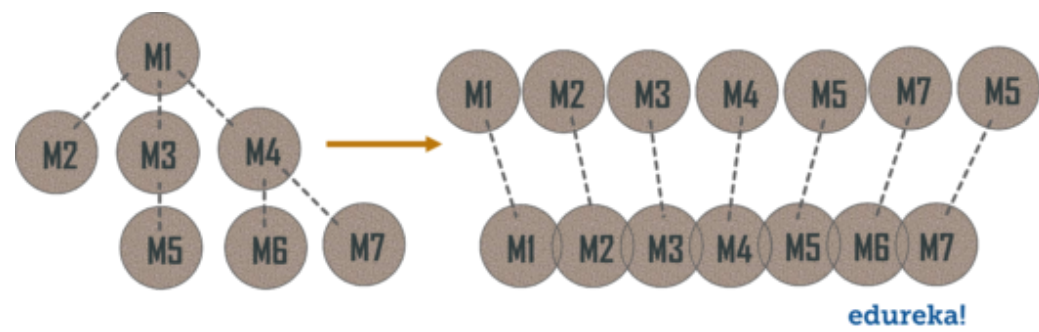
Disadvantages

- Fault localization is tough
- A lot of delay before testing
- Critical issues are not resolved on priority
- Difficult to find the root cause of problems

Incremental Integration Testing

Incremental Testing is a form able operation to arrange mod do together that any logically related into one model.

Top-Down Integration Testing

The top-down approach starts by testing the top-most modules and gradually moving down to the lowest set of modules one-by-one. Testing takes place from top to down following the control flow of the software system. As there is a possibility that the lower level modules might not have been developed while top modules are tested, we use stubs instead of those not ready modules. For simple applications, stubs would simply return the control to their superior modules. For complex applications, they would simulate the full range of responses.
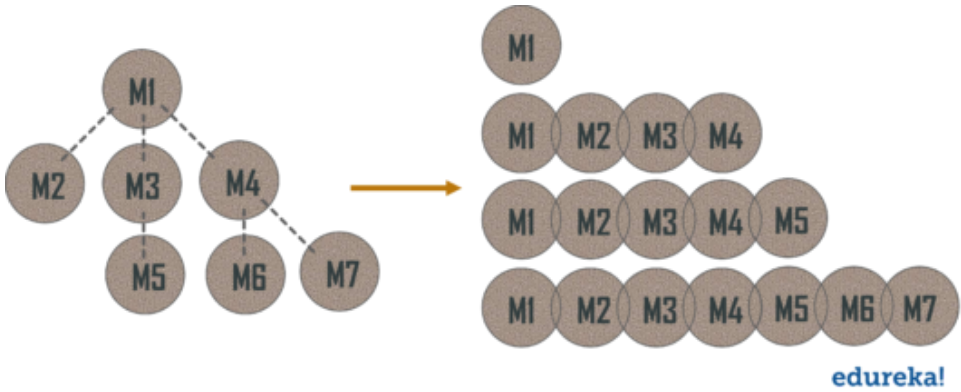

Advantages:

- Fault localization is easier
- The test product is extremely consistent
- The stubs can be written in lesser time compared to drivers
- Critical modules are tested on priority
- Major design flaws are detected as early as possible

Disadvantages

- Requires several stubs
- Poor support for early release
- Basic functionality is tested at the end of the cycle

Bottom-Up Integration Testing

The bottom-up approach starts with testing the lowest units of the application and gradually moving up one-by-one. Here testing takes place from the bottom of the control flow to upwards. Again it's possible that the higher level modules might not have been developed by the time lower modules are tested. In such cases, we simulate the functionality of missing modules by using drivers. These drivers perform a range of tasks such as invoking module under test, pass test data or receive output data.
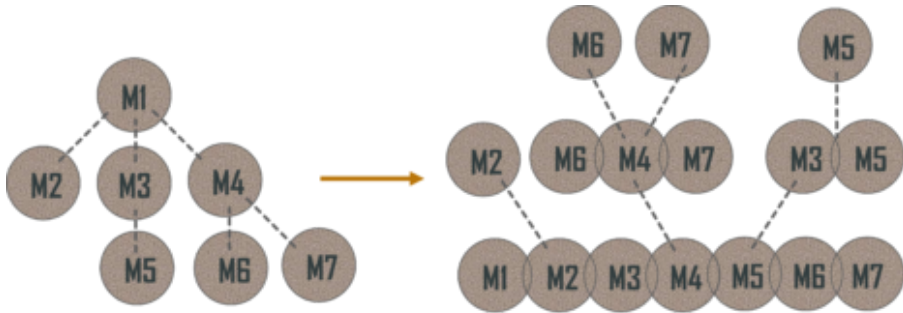
- Need for drivers complicates test data management
- Poor support for early release
- Key interfaces defects are detected late

Sandwich Integration Testing

To overcome the limitations and to exploit the advantages of top-down and bottom-up approaches, a hybrid approach of integration testing is used. This approach is known as sandwich integration testing or mixed integration testing. Here, the system is viewed as three layers. Main target layer in the middle, another layer above the target layer, and the last layer below the target layer. The top-down approach is used on the layer from the top to the middle layer. The bottom-up approach is used on the layer from the bottom to middle. Big bang approach is used for modules in the middle.
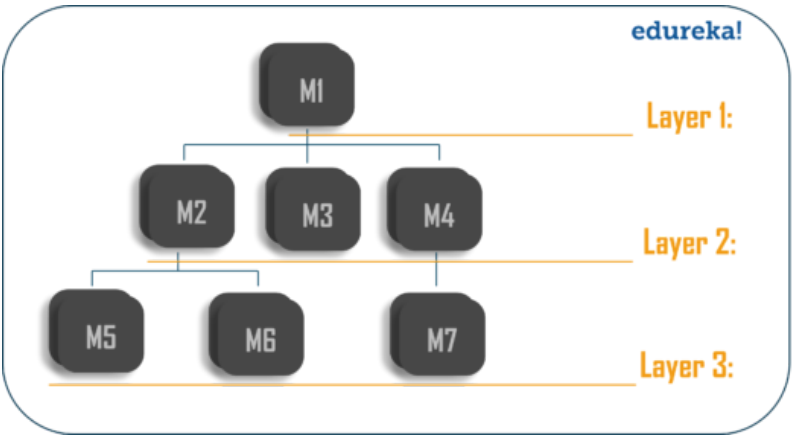
Advantages

- Top-Down and Bottom-Up testing techniques can be performed in parallel or one after the other
- Very useful for large enterprises and huge projects that further have several subprojects

Disadvantages

Selenium Certification Training Course

Weekday / Weekend Batches

See Batch Details

- The cost requirement is very high
- Cannot be used for smaller systems with huge interdependence between the modules
- Different skill sets are required for testers at different levels

These are the different approaches that you can use to perform integration testing based on your testing requirements. You should check which testing strategy can be adopted and prepare the test data and test plan accordingly. As useful as integration testing can be, it is not without its negative points. You need to understand the challenges that you might face when implementing it.

Challenges of Integration Testing

- Managing integration testing is difficult sometimes because of various factors like database, platforms, environment, etc
- Integrating a new system to a legacy system or integrating two legacy system needs a lot of testing efforts and changes
- Less compatibility between the two systems developed by two different companies is a challenge for programmers
- There are way too many different paths and permutations to apply for testing the integrated systems

Despite these challenges, integration testing is very useful in the software testing process. It is an important part of the testing cycle that makes it easy finding defects when two or more units are integrated. With this, we have reached the end of the 'What is Integration Testing?' article. Hope the things that you have learned here today will help you as you head out on your software testing journey.

Selenium WebDriver Tutorial – Get Started With Selenium WebDriver

[Watch Now]

Selenium Tutorial – A Complete Tutorial on Selenium Automation Testing

[Watch Now]

Automated Testing Using Selenium Webdriver
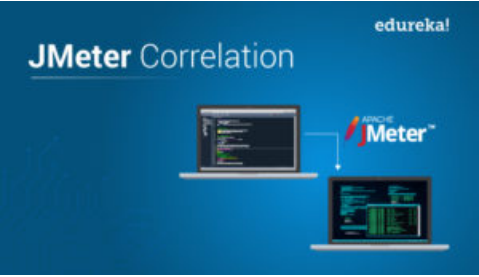
[Watch Now]

Test Automation Wi Selenium Webdriver
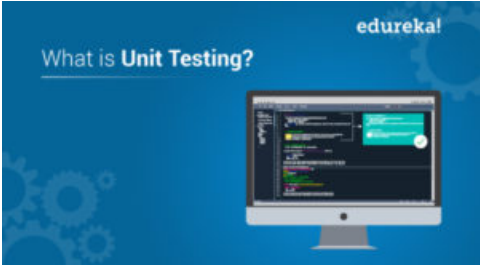
[Watch Now]

‹›

Recommended blogs for you

JMeter Correlation: The Best Way for Extracting Variables

[Read Article]

What Is Unit Testing? Everything That You Need To Know About Unit Testing

[Read Article]

Test Automation Frameworks: How to Build a Framework in Selenium?

[Read Article]

Top 50 Selenium Int Questions And Answ Must Prepare In 202

[Read Article]

‹›

Comments   0 Comments

Join the discussion

Selenium Certification Training Course

👤 35k Enrolled Learners
📅 Weekend/Weekday
🎥 Live Class

Reviews

⭐⭐⭐⭐⭐ 5 (13650)

Manual Testing Certification Training Course ...

👤 6k Enrolled Learners
📅 Weekend
🎥 Live Class

Reviews

⭐⭐⭐⭐⭐ 5 (2300)

Software Testing Fundamentals Course

👤 2k Enrolled Learners
📅 Weekend
🎥 Live Class

Reviews

⭐⭐⭐⭐⭐ 5 (600)

Performance Test JMeter

👤 3k Enrolled Learners
📅 Weekend
🎥 Live Class

Reviews

⭐⭐⭐⭐⭐ 5 (950)

‹›

Browse Categories

Artificial Intelligence | BI and Visualization | Big Data | Blockchain | Cloud Computing | Cyber Security | Data Science

Data Warehousing and ETL | Databases | DevOps | Digital Marketing | Enterprise | Front End Web Development | Mobile Development

Operating Systems | Programming & Frameworks | Project Management and Methodologies | Robotic Process Automation

Systems & Architecture

---

edureka!

TRENDING CERTIFICATION COURSES

DevOps Certification Training

AWS Architect Certification Training

Big Data Hadoop Certification Training

Tableau Training & Certification

Python Certification Training for Data Science

Selenium Certification Training

PMP® Certification Exam Training

Robotic Process Automation Training using UiPath

Apache Spark and Scala Certification Training

Microsoft Power BI Training

Online Java Course and Training

Python Certification Course

TRENDING MASTERS COURSES

Data Scientist Masters Program

DevOps Engineer Masters Program

Cloud Architect Masters Program

Big Data Architect Masters Program

Machine Learning Engineer Masters Program

Full Stack Web Developer Masters Program

Business Intelligence Masters Program

Data Analyst Masters Program

Test Automation Engineer Masters Program

Post-Graduate Program in Artificial Intelligence & Machine Learning

Post-Graduate Program in Big Data Engineering

COMPANY

About us

News & Media

Reviews

Contact us

Blog

Community

Sitemap

Blog Sitemap

Community Sitemap

Webinars

WORK WITH US

Careers

Become an Instructor

Become an Affiliate

Become a Partner

Hire from Edureka

DOWNLOAD APP

CATEGORIES ⌄

Subscribe to our Newsletter, and get personalized recommendations. ✕

G   Sign up with Google

f   Signup with Facebook

Already have an account? Sign in.

TRENDING BLOG ARTICLES

Selenium tutorial | Selenium interview questions | Java tutorial | What is HTML | Java interview questions | PHP tutorial | JavaScript interview questions |

Spring tutorial | PHP interview questions | Inheritance in Java | Polymorphism in Java | Spring interview questions | Pointers in C | Linux commands |

Android tutorial | JavaScript tutorial | jQuery tutorial | SQL interview questions | MySQL tutorial | Machine learning tutorial | Python tutorial |

What is machine learning | Ethical hacking tutorial | SQL injection | AWS certification career opportunities | AWS tutorial | What Is cloud computing |

What is blockchain | Hadoop tutorial | What is artificial intelligence | Node Tutorial | Collections in Java | Exception handling in java |

Python Programming Language | Python interview questions | Multithreading in Java | ReactJS Tutorial | Data Science vs Big Data vs Data Analyt… |

Software Testing Interview Questions | R Tutorial | Java Programs | JavaScript Reserved Words and Keywor… | Implement thread.yield() in Java: Exam… |

Implement Optical Character Recogniti… | All you Need to Know About Implement…

Legal & Privacy

Subscribe to our Newsletter, and get personalized recommendations.

Sign up with Google

Signup with Facebook

Already have an account? Sign in.