Assignment 9 (Software Design)

Objective Identify the elements and relationship by analysing the class diagram of easy shop retail application case study.

Background Software Design describes how software is decomposed and organized into components and the interfaces between those components. It also describes the components at a level of detail that enable their construction

Problem Description

1. Study the class diagram of Easy shop retail application depicted in the "CCFP4.0-SE Easy Shop

Manager-Phase1-ClassDiagram.docx" and identify

the Following for the class "Customer".

1. Class Name

2. Private attributes

3. Public attributes

4. Private methods

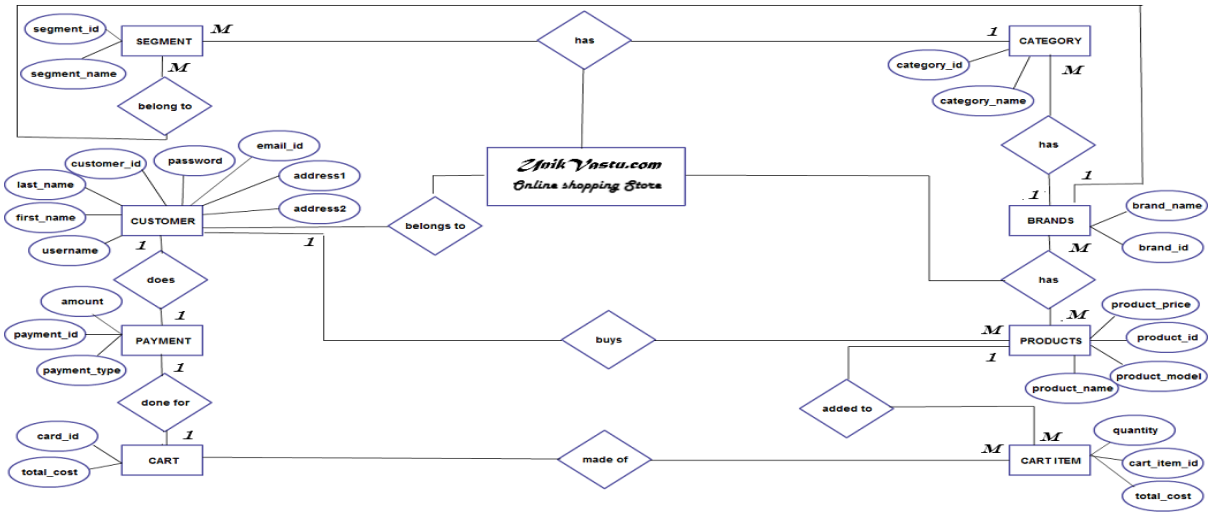
5. Public Methods

2. In above document, identify the relationship between the elements given below:

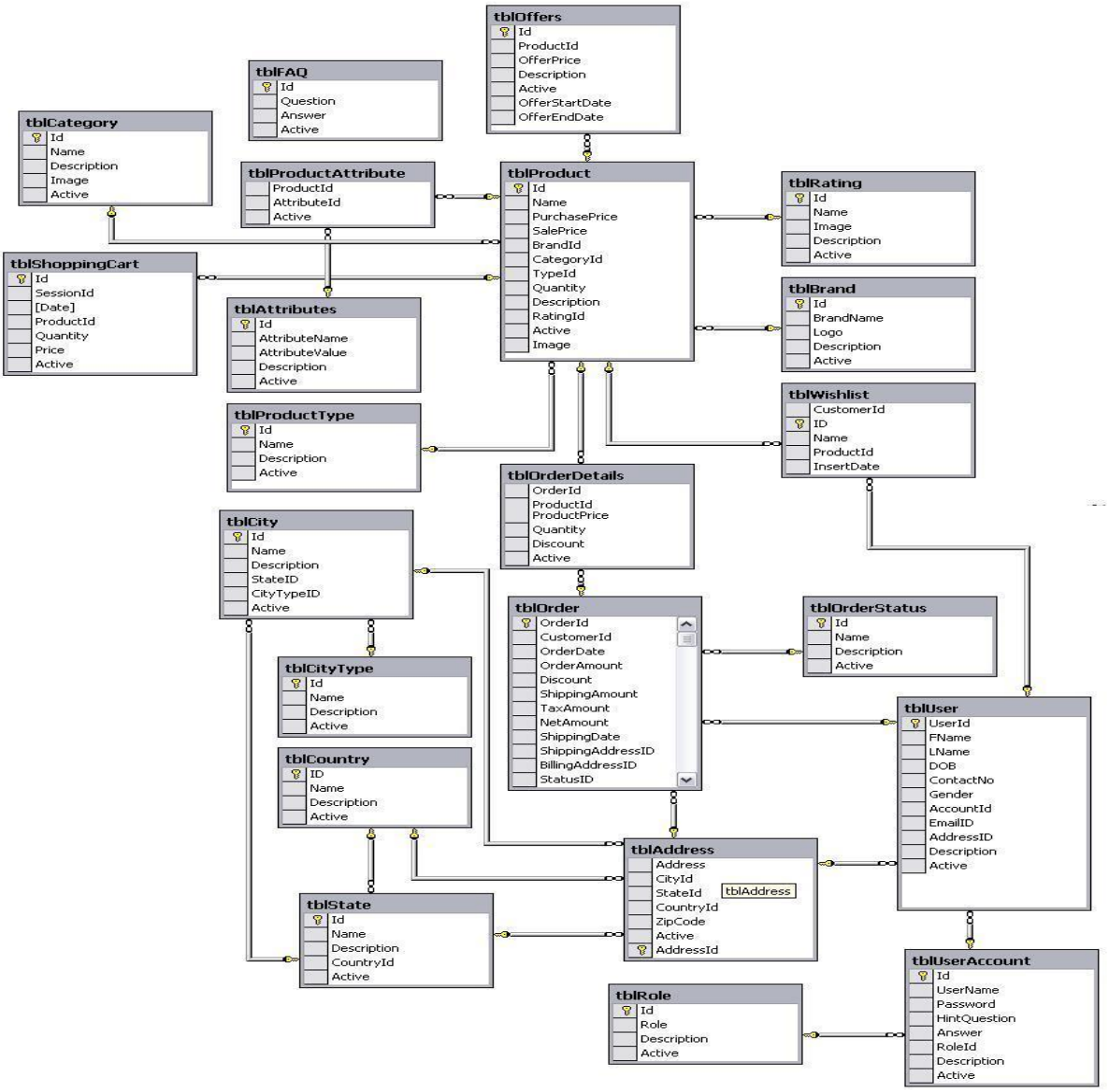
Class 1	Class 2	Relationship
Item	Apparels	
Retail Outlet	Item	
Customer	Regular Customer	
Customer	Address	
Purchase Bill	Line Item	
Tax	Purchase Bill	

Assignment 9 solution

First we have created simple use case diagram.



Now Class Diagram



Q1 **Study the class diagram**

Class	Public Methods & Attributes	Private Methods & Attributes
Product	Id,Name,PurchasePrice, SalerPrice,Description ,Image , Active	Brand id,Category id,Type ID,Quantity,Rating ID
Customer	User id , Active , Description	F name , L name , DOB , Contact number ,Gender , Account id ,Email id ,
Order	Order id , Customer id , Order Date , Order Account , Discount , shipping amount ,Tax amount , Net amount , shipping date , status id, Billing address	

2. In above document, identify the relationship between the elements given below

Class 1	Class 2	Relationship
Item	Apparels	**ONE TO MANY**
Retail Outlet	Item	**MANY TO ONE**
Customer	Regular Customer	**MANY TO ONE**
Customer	Address	**ONE TO MANY**
Purchase Bill	LineItem	**ONE TO ONE**
Tax	Purchase Bill	**MANY TO ONE**

Assignment 10 (Software Design Principles)

Objective Identify the design principle that is being violated in relation to the given scenario.

Background A good object oriented design not only meets the specified requirements but also addresses implicit requirements. There are five design principles which address most of the implicit requirements:

Software Design Principles

1. Abstraction: Focus on solving a problem by considering the relevant details and ignoring the irrelevant 2. Encapsulation: Wrapping the internal details, thereby making these details inaccessible. Encapsulation separates interface and implementation, specifying only the public interface to the clients, hiding the details of implementation. 3. Decomposition and Modularization: Dividing the problem into smaller, independent, interactive subtasks for placing different functionalities in different components 4.Coupling & Cohesion: Coupling is the degree to which modules are dependent on each other. Cohesion is the degree to which a module has a single, well defined task or responsibility. A good design is one with loose coupling and strong cohesion. 5. Sufficiency, Completeness and Primitiveness: Design should ensure the completeness and sufficiency with respect to the given specifications in a very simple way as possible.

Problem Description

Which of the following design principle(s) have been violated in the following scenarios?

1. Abstraction

2. Decomposition and Modularization

3. Coupling & Cohesion

4. Encapsulation

5. Sufficiency, Completeness and Primitiveness

6. All

Assignment 10 solution

No	Description	Principle Being Violated
1	Important information of a module is directly accessible by other modules	Encapsulation
2	Too many global variables in the program after implementing the design	Coupling & Cohesion, Decomposition and Modularization
3	Code breaks in unexpected places	All
4	Unfulfilled requirements in the code after the design has been implemented	All
5	Cyclic dependency among classes	Coupling & Cohesion, Decomposition and Modularization
6	Huge class doing too many unrelated operations	Abstraction, Decomposition and Modularization
7	Several un-related functionalities/tasks are carried out by a single module	Decomposition and Modularization, Abstraction
8	All data of all classes in public	Encapsulation
9	Design resulting in spaghetti code	Sufficiency, Completeness and Primitiveness
10	An algorithm documented as part of design is not understandable by the programmers	Sufficiency, Completeness and Primitiveness

Assignment 11 (Integration Testing)

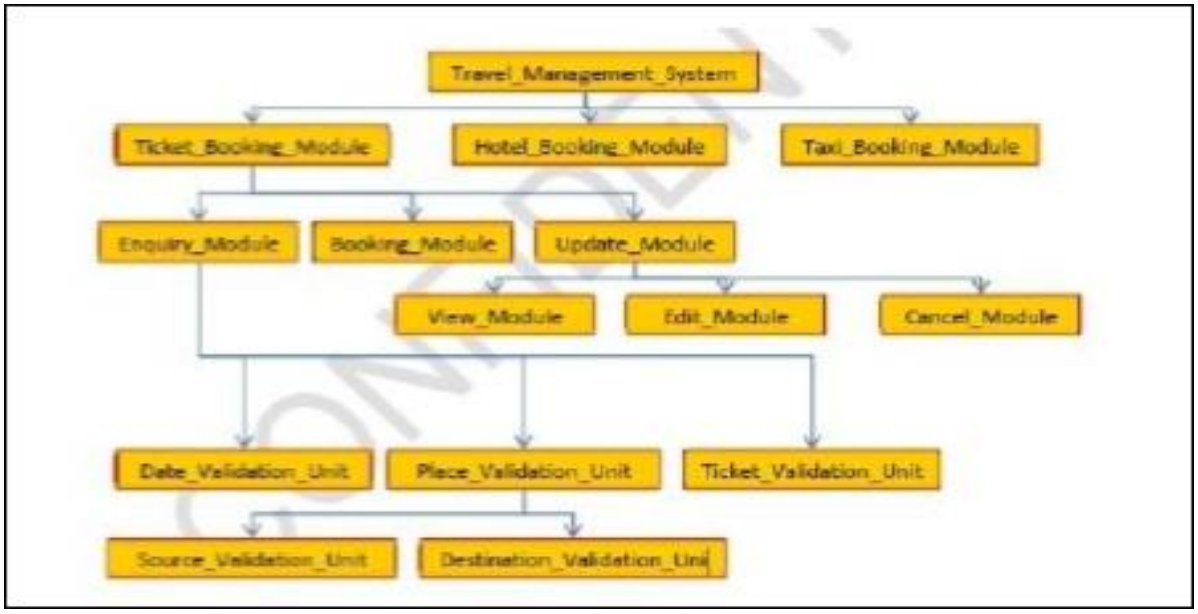
Objective: To identify the usage of stubs or drivers in the context of an integration testing scenario.

Background: Integration testing is carried out after the completion of unit testing and before the software is delivered for system testing. In top down integration testing, dummy stubs are required for bottom level modules. Similarly in bottom up testing, dummy drivers are required for top level modules.

Problem Description

Consider the scenario of development of software for Travel, Management System (TMS) is in progress. The TMS software has 3 major modules namely Ticket_Booking_Module, Hotel_Booking_Module and Taxi_Booking_Module. The Ticket_Booking_Module has 3 sub modules namely Enquiry_Module, Booking_Module and Update_Module. The enquiry

Module uses Date_Validation_Unit, Ticket_Validation_Unit and Place_Validation_Unit.



In the context of the given scenario, identify the usage of stub or driver for the following situations.

1. Except the Ticket_Validation_Unit, the coding and unit testing of all other modules, sub modules and units of TMS are completed. The top-down integration is in progress for the TMS software. To carry out the integration testing, which among the following is necessary?

- A Stub for Ticket_Validation_Unit
- A Driver For Ticket_Validation_Unit
- A Stub for Enquiry_Module
- A Driver for Enquiry_Module
- A Stub For Ticket_Booking_Module
- A Driver For Ticket_Booking_Module

2. The coding and unit testing of all the module, sub modules and units of TMS are completed except the Update_Module (coding and testing for Edit Module, Cancel Module and View Module are also completed). The bottom-up integration is to be started for the TMS software. Mention any stub or driver needed to carry out the integration testing?

3. Except the Taxi_Booking_Module, the coding and unit testing of all other modules, sub modules and units of TMS are completed. The top-down integration is to be started for the TMS software. Mention any stub or driver needed to carry out the integration testing?

Assignment 11 solution

In the context of the given scenario, identify the usage of stub or driver for the following situations.

1. Except the Ticket_Validation_Unit, the coding and unit testing of all other modules, sub modules and units of TMS are completed. The top-down integration is in progress for the TMS software. To carry out the integration testing, which among the following is necessary?
   - **A Stub for Place_Validation_Unit**
   - **A Driver For Ticket_Validation_Unit**
   - **A Stub for Enquiry_Module – YES**
   - **A Driver for Enquiry_Module**
   - **A Stub For Ticket_Booking_Module -YES**
   - **A Driver For Ticket_Booking_Module**
2. The coding and unit testing of all the module, sub modules and units of TMS are completed except the Update_Module (coding and testing for Edit Module, Cancel Module and View Module are also completed). The bottom-up integration is to be started for the TMS software. Mention any stub or driver needed to carry out the integration testing?
   - **A driver for Ticket_Booking_Module**
   - **A driver for Enquiry_Module**
   - **A driver for Place_Validation_Unit**
3. Except the Taxi_Booking_Module, the coding and unit testing of all other modules, sub modules and units of TMS are completed. The top-down integration is to be started for the TMS software. Mention any stub or driver needed to carry out the integration testing.
   - **A Stub for Place_Validation_Unit**
   - **A Stub for Enquiry_Module**
   - **A Stub for Update_Module**
   - **A Stub For Ticket_Booking_Module**

Assignment 12 (Performance Testing)

Objective: Identify the different types of performance testing

Background: Performance testing tests the non-functional requirements of the system. The different types of performance testing are load testing, stress testing, endurance testing and spike testing.

Problem Description: explain with example and tools

Assignment 12 solution

What is Performance Testing?

Performance Testing is performed to evaluate the performance of components of a particular system under a particular workload. During this testing, system components are monitored to verify the stability of the system under test.

Performance Testing is the type of **Non-Functional Testing**. It strives to build performance standards into the implementation, design and architecture of a system.

Performance Testing Attributes: Speed, Scalability, Stability & Reliability.

Speed - Determines whether the application responds quickly.

Scalability - Determines maximum user load the software application can handle.

Stability - Determines if the application is stable under varying loads.

Reliability - can generally be looked at as any interruptions in usage

Or **performance** during. The lifetime span of a product, part, material, or

System.

Features or Characteristics of Performance Testing Tools are:

- To generate load on the system to be tested.
- To measure the timing of specific transactions as the load on the system varies.
- To measure average response times.
- To produce graphs or charts of responses over time.

Objective of Performance Testing

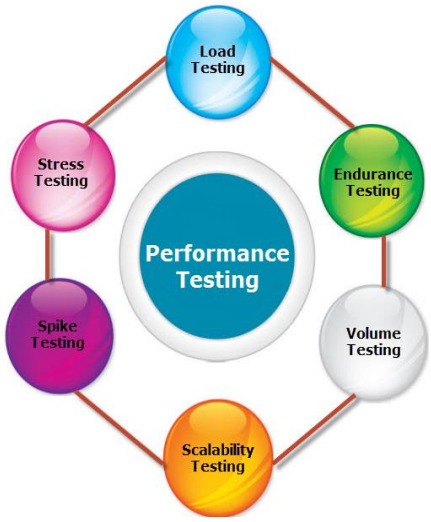
The primary objective is to establish the benchmark behaviour of the system. It does not aim to find defects in the application rather this focus on measuring characteristics, such as ***response times***, ***throughput*** or the ***mean time***. This can be done in different ways depending on the tool, such as different user profiles, different types of activity, timing delays and other parameters.

Performance Testing Techniques:

- **Load Testing** – Testing the behaviour of the system under a specific load or to get the breakeven point where system starts downgrading its performance.

- **Stress Testing:** It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.
- **Usability Testing:** Testing to determine the extent to which the software product is understood, easy to learn, easy to operate and attractive to the users under specified conditions.
- Security Testing: This intends to uncover vulnerabilities of the system and determine that its data and resources are protected from possible intruders.
- **Portability Testing:** Software reliability is the probability that software will work properly in a specified environment and for a given amount of time.

- **Soak Testing**: Soak Testing is a type of performance test that verifies a system's stability and performance characteristics over an extended period of time or for long durations.
- **Spike Testing:** Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system.
- **Isolation Testing:** Isolation testing is not unique to performance testing but involves repeating a test execution that resulted in a system problem. Such testing can often isolate and confirm the fault domain.



Example of Performance Test Cases

Writing test cases for performance testing requires a different mind-set compared to writing functional test cases.

Read more: How to Write Functional Test Cases.

- To verify whether an application is capable of handling a certain number of simultaneous users
- To verify whether the response time of an application under load is within an acceptable range when the network connectivity is slow
- To verify the response time of an application under low, normal, moderate and heavy load conditions
- To check whether the server remain functional without any crash under high load
- To verify whether an application reverts to normal behaviour after a peak load
- To verify database server and CPU and memory usage of the application under peak load

Difference between Functional and Non Functional Testing in Performance Testing

FUNCTIONAL TESTING	NON-FUNCTIONAL TESTING
What the system actually does is functional testing	How well the system performs is non-functionality testing
To ensure that your product meets customer and business requirements and doesn't have any major bugs	To ensure that the product stands up to customer expectations
To verify the accuracy of the software against expected output	To verify the behaviour of the software at various load conditions
It is performed before non-functional testing	It is performed after functional testing
Example of functional test case is to verify the login functionality	Example of non-functional test case is to check whether the homepage is loading in less than 2 seconds
Testing types are • Unit testing • Smoke testing • User Acceptance • Integration Testing • Regression testing • Localization • Globalization • Interoperability	Testing types are • Performance Testing • Volume Testing • Scalability • Usability Testing • Load Testing • Stress Testing • Compliance Testing • Portability Testing • Disaster Recover Testing
It can be performed either manual or automated way	It can be performed efficiently if automated

Performance Testing Tools:

1. Load Runner (HP Performance Tester)

Load Runner is a Load Testing Software from Micro Focus for application load testing. As a performance testing tool, it is used to test applications, measuring system behaviour, and performance under load. It can create and handle thousands of concurrent users to put the application through the rigors of real-life user loads while gathering required information with respect to the performance and also based on the infrastructure components (Web servers, database servers etc.). It reports the end user response times for business processes and transactions to compare them against the service level agreements (SLA). The results can then be analysed in detail to explore the reasons for a particular behaviour. It is compatible with operating systems like Microsoft Windows, and Linux.

Load Runner Features:

Following are some of the most important features of Load Runner

- It supports performance testing of the latest technologies as well as the legacy applications.
- Supports both Browser-based and Native Mobile applications tests using advanced Network behaviour and Service Virtualization
- Integrate load testing into your development tools: IDE, jUnit, nUnit, Jenkins, Selenium and Microsoft Visual Studio
- It identifies performance bottlenecks by using integrated performance monitors in real-time that leverage application-layer and code-level data for root cause and analytics

Protocols:

Protocols and technologies supported by Load Runner are as follows.

Load Runner supports performance testing for a wide range of application environments and protocols more than 50 which includes Ajax, Flex, HTML 5, Microsoft Silverlight, HTTP/2, MQTT, Web, SOAP, ERP, Web services, GWT, RDP, Database, Terminal, Citrix, Java, .NET, Oracle, and SAP. It supports the latest application technologies as well as the legacy ones.

Download link: https://www.microfocus.com/products/loadrunner-load-testing/free-trial

2. Apache JMeter

Apache JMeter application is a free and open source load testing tool. It was written in Java. It was originally designed for testing web applications but later its scope has expanded. It is designed to load test for analysing and measuring the performance of a

variety of services. It can be used to test performance on both static as well as dynamic resources such as dynamic web applications. It is useful to simulate a heavy load on a server, group of servers, network to review the performance of an application under different load conditions. It works on Linux, Windows & Mac OS X.

Apache JMeter Features:

Following are some of the most important features of JMeter

- It is a free open source software
- It comes with a simple and intuitive GUI.
- It is a platform independent tool. It is written and developed using Java. It can run on any environment which accepts JVM (Java Virtual Machine).
- It is highly extensible and supports different server types.
- Its full multi-threading framework allows concurrent sampling by many threads and simultaneous sampling of different functions by separate thread groups.

A complete and ready to present dynamic HTML report

Easy correlation through an ability to extract data from most popular response formats, HTML, JSON, XML or any textual format

It supports multiple protocols

Protocols & Technologies:

Protocols supported by JMeter are as follows.

Web – HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, etc.,)

Web Services – SOAP/REST , FTP Service

Database via JDBC drivers

LDAP directory

Mail Services – SMTP, POP3, IMAP

Messaging-oriented middleware (MOM) via JMS

Native command or shell scripts

TCP

Java objects

Download Link: https://jmeter.apache.org/

3. NeoLoad

NeoLoad is a powerful load and performance testing software solution designed for web and mobile applications. It simulates a large number of users and analyses server behaviour. It identifies performance bottlenecks and provides a solution to optimize the

design and development of the application before they become expensive issues in production. It allows users to conduct load tests quickly, efficiently, and frequently. This means you can confidently deploy high-performance internet, intranet or mobile applications regardless of what technologies. It even supports the newest technologies such as Flex, Silverlight, GWT, SPDY, JSON, and AJAX Push. It integrates with Continuous Delivery platform. It is compatible with operating systems like Microsoft Windows, Linux, and Solaris.

NeoLoad Features:

Following are some of the most important features of NeoLoad

- It supports latest technologies like HTML5, Push, Web Socket, AngularJS, Oracle Forms and many more
- Script less design and visual programming
- Integration with Continuous Integration servers. It provides an out-of-the-box CI plugin for Jenkins, TeamCity, Bamboo, and XebiaLabs XL Release.
- Native GIT support
- It integrates with functional testing tools like Selenium, Appium, Perfecto Mobile Cloud etc.,

Protocols:

Protocols and technologies supported by NeoLoad are as follows.

SAP GUI Web, HTTP, HTTPS, Web Socket, SOAP, REST, Silverlight, Java Serialization, Java Message Service, GWT, AJAX, Oracle Forms, Push technologies, etc.,

Download Link: https://www.neotys.com/neoload/overview

4. StresStimulus

StresStimulus is a load testing tool for web applications, mobile, and Enterprise apps. It determines the web performance and scalability of an application under the rigors of heavy traffic load. It collects real-time server monitoring information to pinpoint application performance bottlenecks and isolate web speed issues. It supports native app framework which saves time on mobile app load testing. It records user actions and replays them in order to emulate variable usage patterns. It also monitors a load impact on application responsiveness and server infrastructure. It automatically fixes playback errors. It comes with both free and commercial versions. You can get a 7-day extendable trial with up to 10k virtual users and full support in the free trial.

StresStimulus Features:

Following are some of the most important features of StresStimulus

- It can be used on-premises or from the cloud.
- It works with or without Fiddler. Fiddler helps to create, analyse and debug HTTP traffic.

- Scripting is not required but is available. We can create the entire test case through a GUI and wizard-based interface without touching script.
- Multiple secure authentication methods are supported, including Web Forms, Basic, NTLM, and Kerberos.
- It supports parameterization with external data (CSV files)
- It gives in-depth reporting. It generates a test summary report that presents key performance metrics.

Protocols:

Protocols and technologies supported by StresStimulus are as follows.

HTTP, HTTPS, AJAX, SOAP, WCF, binary WCF, and XML over HTTP.

Download Link: https://www.stresstimulus.com/

5. LoadUI Pro

LoadUI Pro by Smart bear is a load testing tool for REST & SOAP APIs, Databases, and Micro services. It runs on Windows, Linux and Mac OS. It allows users to create script less and sophisticated load tests in the shortest time. It allows users to test the speed and scalability of APIs, preview API performance behaviours before releasing to production environments and shift performance insights to the left. Users can access detailed reports and automate load tests on Bamboo, Jenkins, TFS, and other automation frameworks. Additionally, SoapUI functional tests can be quickly converted into load tests using LoadUI Pro without writing a single line of a script.

LoadUI Pro Features:

Following are some of the most important features of LoadUI Pro

- Cloud-based API load tests
- Script less load test creation
- Reuse existing functional tests without modifying the original tests
- Parallel API load testing
- Server monitoring gives visibility into how servers respond to traffic

Protocols:

Protocols and technologies supported by LoadUI Pro are as follows

HTTP, REST, SOAP, JSON, JMS, JSON Schema, XML Schema, WSDL etc.,

Download Link: https://smartbear.com/product/ready-api/loadui/overview/

6. WebLOAD

WebLOAD is an enterprise-grade load and performance testing tool which is designed for web applications. It supports over a hundred technologies from web protocols to enterprise applications and has built-in integration with Jenkins, Selenium and many

other tools to enable continuous load testing for DevOps. It combines performance, scalability, and integrity as a single process for the verification of web and mobile applications. It can simulate hundreds of thousands of concurrent users making it possible to test large loads and report bottlenecks, constraints, and weak points within an application. The results of the tests are collected from the Load Machines and it can be viewed in real-time in a tabular or graphical format. It is compatible with operating systems like Microsoft Windows, and Linux.

WebLOAD Features:

Following are some of the most important features of WebLOAD

- It generates load from on-premises machines or from the cloud.
- A set of predefined analysis reports provides performance data, helping users identify bottlenecks. Reports and analysis data can also be viewed remotely via a customizable Web Dashboard.
- It supports a wide range of web, mobile, and enterprise protocols and technologies.

Powerful correlation engine recognizes both server side and client side dynamic values.

Protocols:

Protocols and technologies supported by WebLOAD are as follows.

It supports a wide range of web, mobile, and enterprise protocols and technologies such as HTTP/HTTPS, Web Socket, PUSH, AJAX, SOAP, HTML5, WebDAV and many others.

Download Link: https://www.radview.com/webload-download/

7. Rational Performance Tester

IBM Rational Performance Tester (RPT) is a performance and load testing tool that identifies the presence and cause of system performance bottlenecks. It allows the development team to validate the scalability and reliability of web-based applications before deployment into a production environment.

It can be used for both web-based applications and server-based applications. It identifies and rectifies leakages in the websites and the servers. It is compatible with operating systems such as AIX, Mac OS X, and Microsoft Windows.

Rational Performance Tester Features:

Following are some of the most important features of Rational Performance Tester

- Offers real-time reporting for immediate awareness of performance problems at any time during a test.
- Root cause analysis helps to identify both the source code and physical application tier that are causing the bottleneck.
- Supports load testing against applications such as HTTP, SAP, Siebel, SIP, TCP Socket, Citrix

- Offers emulation of user populations while minimizing the memory and processor footprint
- Automates test data variation and enables insertion of custom Java code

Protocols:

Protocols and technologies supported by Rational Performance Tester are as follows.

HTTP, Citrix, SOA, SOP, Socket Recording etc.,

Download Link: http://www.ibm.com/developerworks/downloads/r/rpt/

8. AppLoader

AppLoader is a load testing tool to make sure the readiness of an application. It allows you to test any application by reproducing the same user experience from all your access points: Thin, fat clients, and web portals. It tests any business application that is accessed through thin, fat clients and web portals. Entire business flow can be tested, including all third-party apps, without adding plugins or writing a single line of code. It is compatible with all versions of Citrix, Cloud-Based & Hybrid infrastructures, EHR systems, custom apps

AppLoader Features:

Following are some of the most important features of AppLoader

- It supports Citrix, PeopleSoft, Java, .NET, Adobe, client-server, Oracle, Siebel, SAP, web, custom apps etc.,
- It is protocol independent
- Load Test any application, any environment
- Simulate user actions on any kind of application for any number of users.
- Build any scenario without scripting, (including mouse events, keyboard inputs, if conditions, etc.), and replay these actions as an automat.

Protocols:

AppLoader is protocol independent

Download Link: https://www.nrgglobal.com/apploader-performance-testing-download

9. SmartMeter.io

SmartMeter.io is a multi-platform load and performance testing tool. It features fast and easy test creation and execution, test management and generating of test reports with a focus on testing in a distributed mode. It is based on Apache JMeter but adds new features such as one-click test reports, advanced scenario recorder, acceptance criteria,

and others. It is compatible with operating systems such as Linux, Mac OS, and Microsoft Windows.

SmartMeter.io Features:

Following are some of the most important features of SmartMeter.io

- Scriptless test recording
- It fits very well into a continuous integration process
- Automatically generated test reports with test details and results
- Real-time test results on multiple monitors
- Combine load tests with Selenium

Protocols:

Protocols and technologies supported by SmartMeter.io are as follows.

HTTP, HTTPS, FTP, JDBC, LDAP, SOAP, and JMS

Download Link: https://www.smartmeter.io/download

10. Silk Performer

Silk Performer is a load and stress testing tool for optimizing business application performance from Micro Focus. It has the ability to test multiple application environments with thousands of concurrent users. Silk Performer can considerably speed up testing cycles by allowing you to reuse your existing functional tests (Silk Test or Selenium) for performance testing and synthetic monitoring purposes as well. It also supports the widest range of protocols.

Silk Performer Features:

Following are some of the most important features of Silk Performer

- Re-Use Functional Test Assets for Performance Testing and Monitoring
- Cloud integration and unlimited scalability from the cloud
- Supports Server monitoring, reporting, and analysis
- It provides extensive support for mobile web and native applications, with profiles for all popular mobile devices, application types and connection speeds.

Protocols:

Protocols and technologies supported by Silk Performer are as follows.

HTTP(S)/HTML, HTTP/2, IPv6, Ajax, Silverlight, mobile devices, Java over HTTP, HTTP Live Streaming (HLS), Adobe Flex/AMF3, Granite DS Flex, Unicode (UTF-8), SOAP (XML), FTP, LDAP, MAPI, IMAP, SMTP/POP, SSL, CORBA (IIOP), Java RMI (EJB/J2EE), .NET Remoting, Oracle Forms, Citrix, VMWare Horizon View, ODBC, Oracle Call Interface (OCI), DB2 CLI, TCP/ IP, UDP, Tuxedo ATMI, Jolt, TN3270E, TN5250, T100/200+, and UI-Level (Silk Test, Selenium)

Download Link: https://www.microfocus.com/products/silk-portfolio/silk-performer/

Example for Performance Testing

Identify the type of performance testing for the following:

1. A space craft is expected to function for nearly 8 years in space. The orbit control system of the spacecraft is a real-time embedded system. Before the launch, the embedded software is to be tested to ensure that it is capable of working for 8 years in the space. Identify the suitable performance testing category to be carried out to ensure that the space craft will be functioning for 8 years in the space as required.

2. Global Education Centre (GEC) at Infosys Mysore provides the training for fresh entrants. GEC uses an automated tool for conducting objective type test for the trainees. At a time, a maximum of 2000 trainees are expected to take the test. Before the tool is deployed, testing of the tool was carried out to ensure that it is capable of supporting 2000 simultaneous users. Indicate the performance testing category?

3. A university uses its web based portal for publishing the results of the students. When the results of an examination were announced on the website recently on a pre-planned date, the web site crashed. Which type of performance testing should have been done during web-site development to avoid this unpleasant situation?

4. During unexpected terrorist attack, one of the popular websites crashed as many people logged into the web-site in a short span of time to know the consequences of terrorist attack and for immediate guidelines from the security personnel. After analysing the situation, the maintenance team of that website came to know that it was the consequences of unexpected load on the system which had never happened previously. Which type of performance testing should have been done during web-site development to avoid this unpleasant situation?

Scenarios	Performance Testing Type
Scenario 1	Performance testing
Scenario 2	Load testing
Scenario 3	Stress testing
Scenario 4	Endurance testing

Assignment 13 (Regression Testing)

Objective: To identify the usage of regression testing**.**

Background: Enhancements are introduction of new features to the software and might be released in different versions. Whenever a version is released, regression testing should be done on the system to ensure that the existing features have not been disturbed.

Problem Description: explain with example and tools

Assignment 13 solution

What is Regression Testing?

Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to make sure the product works fine with new functionality, bug fixes or any change in the existing feature. Previously executed test cases are re-executed in order to verify the impact of change.

Regression Testing is a Software Testing type in which test cases are re-executed in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs.

This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix.

Regression means retesting the unchanged parts of the application.

Need of Regression Testing

Regression Testing is required when there is a

- Change in requirements and code is modified according to the requirement
- New feature is added to the software
- Defect fixing
- Performance issue fix

Steps to take in Regression Testing:

- Prepare a Regression Test Suite by following above techniques of "How to select Test for Regression Test Suite"
- Automate the Regression Suite using the choice of Automation Tool.
- Manage Regression Suite and update it when required.
- Execute Regression Suite on the above events of above "When to perform Regression Testing?.
- Analyse the test execution report.

Types of Regression Testing

Given below are the various types of Regression:

Unit Regression

Partial Regression

Complete Regression

1) Unit Regression

Unit Regression is done during the Unit Testing phase and code is tested in isolation i.e. any dependencies on the unit to be tested are blocked so that the unit can be tested individually without any discrepancy.

2) Partial Regression

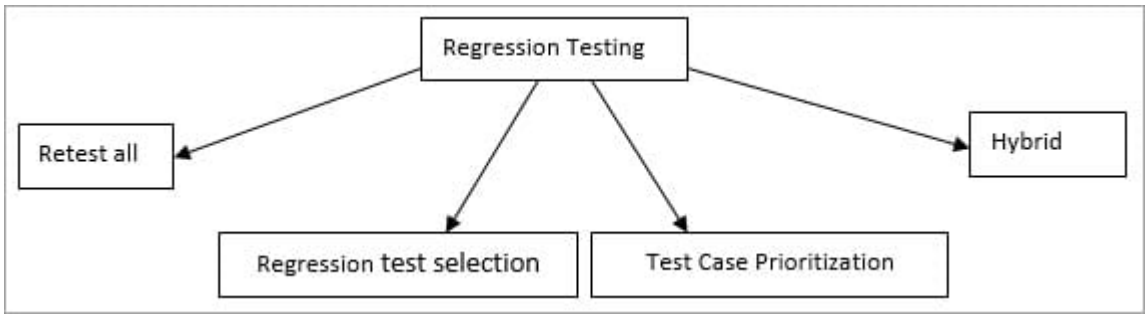
Partial Regression is done to verify that the code works fine even when the changes have been done in the code and that unit is integrated with the unchanged or already existing code.

3) Complete Regression

Complete Regression is done when a change in the code is done on a number of modules and also if the change impact of a change in any other module is uncertain. Product as a whole is regressed to check any changes because of the changed code.

Regression Testing Techniques

Given below are the various techniques.



Regression testing techniques

1) Retest All

As the name itself suggests, the entire test cases in the test suite are re-executed to ensure that there are no bugs that have occurred because of a change in the code. This is an expensive method as it requires more time and resources when compared to the other techniques.

2) Regression Test Selection

In this method, test cases are selected from the test suite to be re-executed. Not the entire suite is re-executed. Selection of test cases is done on the basis of code change in the module.

Test cases are divided into two categories, one is Reusable test cases and another one is Obsolete test cases. The reusable test cases can be used in future regression cycles whereas obsolete ones are not used in the upcoming regression cycles.

3) Test Case Prioritization

Test cases with high Priority are executed first than the ones with medium and low priority. The priority of test case depends on its criticality and its impact on the product and also on the functionality of the product which is used more often.

4) Hybrid

The hybrid technique is a combination of Regression test selection and Test case Prioritization. Rather than selecting the entire test suite, select only the test cases which are re-executed depending on their priority.

Regression Testing Tools:

1) Ranorex Studio

Shorten your regression testing cycles with Ranorex Studio, your all-in-one solution for test automation of desktop, web, and mobile apps. Used by over 4,000 companies worldwide, Ranorex Studio is easy for beginners with a codeless click-and-go interface and helpful wizards, but powerful for automation experts with a full IDE.

Features include:

- Reliable object identification, even for web elements with dynamic IDs.
- Shareable object repository and reusable code modules for efficient test creation and reduced maintenance.
- Data-driven and keyword-driven testing.
- Customizable test report with video reporting of test execution – see what happened in a test run without having to re-run the test!
- Run tests in parallel or distribute them on a Selenium Grid with built-in Selenium Web driver support.
- Integrates with tools like Jira, Jenkins, TestRail, Git, Travis CI, and more.

Download link: =>https://www.ranorex.com/regression-testing-tool/?utm_source=softwaretestinghelp&utm_medium=cpc&utm_campaign=softwaretestinghelp_regression-testing

2) Katalon Studio

Katalon studio new logo

Katalon Studio is an automated testing solution built on top of Selenium and Appium for web, API, mobile, and desktop. It is recommended as a top Customer's Choice by Gartner Peer Insights.

Tool highlights:

- Lightweight. Deployable on Windows, macOS, and Linux
- Supports end-to-end testing for web, API, mobile and desktop applications
- Easy to use for beginners with robust spying & recording functionalities
- Infinite testing extension for experts with plugin platform
- Supports various testing methodologies: keyword-driven, data-driven, and TDD/BDD testing
- Seamlessly integrate into CI/CD systems such as Jira, Jenkins, CircleCI, Bamboo, Selenium Grid, and more
- Global community and experts support

Download link=> https://www.katalon.com/

3) **Sahi Pro**

Sahi Pro is a tester focused automation regressing testing tool. It is one of the most popular regression testing tools that suits best for testing large web applications quickly and with lesser maintenance effort.

Tool Highlights:

- The coolest feature of this tool is the smart accessor mechanism that does not let the test script to fail even if there are slight changes in UI.
- Inbuilt Logging and reporting
- Distributed and parallel playback
- Data-driven suites
- Cross-browser & OS support
- Email reports
- Inbuilt Excel framework.

Official website: http://sahipro.com/

4) Selenium

It is one of the top automated regression testing tools for web application testing. Selenium WebDriver can be employed to build powerful, browser-based regression automation suites and tests.

Tool Highlights:

- Selenium has cross-environment, OS & browser support.
- It is compatible with multiple programming languages and other testing frameworks.
- No doubt, it's a great tool for doing frequent regression testing.

Visit Selenium Website: http://www.seleniumhq.org/

5) Watir

Watir (pronounced as water) is a short form for Web Application Testing in Ruby. It uses the Ruby programming language. Watir can be used to automate regression testing suites

Tool Highlights:

- Very light-weight and easy to use tool
- This tool has great browser interaction capabilities.
- Intended for testing web apps.
- Allows you to design simple, complaint, readable and maintainable automated tests.
- Technology independent
- Cross-platform OS support
- Used by many big companies like SAP, Oracle, Facebook, etc.

Official website: http://watir.github.io/

Link for download: https://github.com/watir/watir

6) Test Complete

Regression tests can be easily and quickly automated by using the Test Complete Platform. It also integrates very well with defect tracking tools.

Tool Highlights:

- Execute parallel regression tests with automated builds.
- Identifies and fixes buggy code quickly.
- Let's you to create regression tests that are stable enough that they do not break on UI changes.
- Automatically schedules and runs the regression tests without any manual intervention.
- Supports desktop, web and mobile apps.
- Useful for GUI testing.
- Cuts down training cost and testing time significantly.

Official website: https://smartbear.com/product/testcomplete/overview/

7) IBM Ra----tional Functional Tester

IBM rational functional tester is mainly meant automated functional testing & regression testing.

Tool Highlights:

- Built over script assure technology
- The call Script capability of IBM RFT facilitates creating and running the regression test suite.
- Improved testing efficiency and easy script maintenance.
- Also, supports data-driven and GUI testing.
- Supports a wide range of apps like web-based, terminal emulator based apps, NET, Java, Ajax, etc.

Official website: http://www-03.ibm.com/software/products/en/functional

8) TimeShiftX

TimeShiftX is a date shifting software that lets you time travel apps to perform temporal testing.

Tool Highlights:

- Uses virtual times so no system clock changes are required.
- Allows for time travel inside Active Directory, Kerberos, LDAP, and other domain authentication protocols.
- Enables time shift testing for all applications & databases such as SAP, SQL, Oracle, WAS, and .NET.
- Supports all platforms & operating systems and can be run in the cloud or containers.

Official website: https://www.vornexinc.com/

#9) Test-drive

It is an automated software quality (ASQ) solution that lets you do automated regression testing rapidly. It lets you come up with dynamic, flexible and easy deployable tests.

Tool Highlights:

- Code-free test automation
- Modular scripts
- Reduced testing time
- Easy to handle changes in the application
- Allows human input
- Supports multiple technologies and interfaces
- Useful for testing browser apps, legacy apps, and GUIs.
- Also, supports manual testing.

Official website: http://origsoft.com/products/testdrive/

10) Advent Net QEngine

QEngine can be used for regression testing of web applications. It is an extensive, platform-independent automation testing tool.

Tool Highlights:

- Easy to use UI.
- Supports IE and FF browsers.
- Event Recording and distributed playback support
- Session tracking
- Server monitoring capability
- Virtual user simulation
- Parameterization to support dynamic values

Official website: https://www.manageengine.com/products/qengine/qengine-eol.html

Example for Performance Testing

Consider the scenario of development of software for Travel Management System (TMS) discussed in previous assignment. TMS has been developed by Infosys and released to its customer Advance Travel Solutions Ltd. (ATSL). Integration testing, system testing and acceptance testing were carried out before releasing the final build to the customer. However, as per the customer feedback during the first month of usage of the software, some minor changes are required in the Enquiry Module of the TMS. The customer has approached Infosys with the minor changes for upgrading the software. The development team of Infosys has incorporated. Those changes, and delivered the software to testing team to test the upgraded software. Which among the following statement is true?

1 Since minor changes are there, integration of the Enquiry Module and quick system testing on Enquiry module should be done. - **True**

2 The incorporation of minor changes would have introduced new bugs into other modules, so regression testing should be carried out. - **True**

3 Since the acceptance testing is already carried out, it is enough if the team performs sanity testing on the Enquire module. - **True**

4 No need of testing any module - **False**

Statement: The incorporation of minor changes would have introduced new bugs into other modules, so regression testing should be carried out

Assignment 14 (Defect Classification)

Objective: To classify the given defects into different defect types.

Background: Defect detection activities like reviews and testing help in identifying the defects in the artifacts (deliverables). These defects must be classified into various buckets before carrying out the root cause analysis.

Following are some the defect Categories.

1. Logical

2. User interface

3. Maintainability

4. Standards

Problem Description

In the context of the above defect categories, classify the following statements under the defect categories and mention in the table given below.

1. Divide by Zero Error is not guarded

2. Usage of 3.14 in the statement Circle_Area = 3.14 * Radius * Radius;

3. 3500 lines of code in a single function

4. A pointer is declared but not initialized. It is used in the program for storing a value.

5. A program designed to handle 1000 simultaneous users, crashed when 1001 the user logged in.

6. A “while” loop never exits

7. User interface displays “MALFUNCTION 54” when something goes wrong in the back-end

8. No documentation (comments) for the source code

9. Hungarian Notation not followed while coding, even though the coding guidelines mandate to use Hungarian Notation

10. Pressing of “Tab” key moves the cursor in different fields of a web form randomly.

Assignment 14 solution

Statement	Statement Defect Category	Defect Name
1	Logical	Technical Error
2	Standards	Programming
3	Maintainability	Coding Standards
4	Logical	Technical Error
5	Maintainability	Limited User
6	Logical	Technical Error
7	User interface	Define Defects in ui
8	Standards	No source availability
9	Standards	Coding Standards
10	Logical	Technical Error

Assignment 15 (agile, scrum and DevOps)

Objective: To brief about agile, scrum and DevOps

Background: Enhancements are introduction of new features to the software and might be released in different versions.

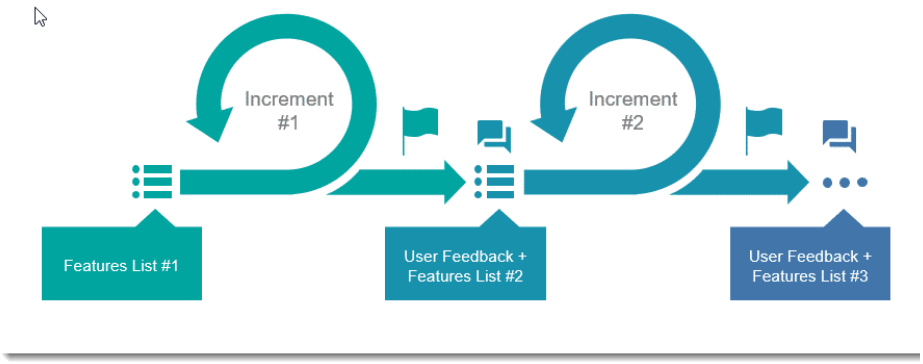
Problem Description: explain with example and tools

Assignment 15 solution

What is Agile?

The first thing that we need to understand about Agile is that it's an approach and a mind-set. It's not a list of instructions to follow. Neither is it a textbook nor a certificate. If we try to turn Agile into a black and white paper document, then it goes against everything that Agile is.

Agile Process



- The agile model adopts iterative development, and each iteration is designed to be small and manageable that can be delivered in a specific short period of time. i.e., a week or a couple of weeks.

- An agile model is a group of development processes, and its main motive is to remove/avoid activities that may not be required for the project and to remove anything which is a waste of time and effort.

- Agile gives us an idea about how much of our delivered product is valuable and whether we've missed out onto something that counts.

- Agile is a framework that defines how software development needs to be done. It's not a single or specific method, and it is the collection of various methodologies and best practices that follow the value statement signed with the customer.

How does agile work?

When you have many things to do but do not have enough time to complete them all, what will you do? You will make a list of all things, prioritize them, assign time and try to complete all the tasks.

Agile works like this. We will discuss the agile process step-by-step.

To understand it better, we can take the help of an example. Let's consider a customer who wants you to develop software for them. Below would be the steps that you will follow as per Agile:-

Create a List: This step includes meeting customer and taking down all the details and features that they want to see in their software. This list of requirements and functions is called the User Requirements List, and this becomes the to-do list for this project.

Estimating and Sizing: After making the user list, you will size stories relative to each other using Agile Estimation techniques and come up with an estimate of how many days each story would take to complete.

Setting Priorities: After giving an estimate you can take customer's help to know their preference. You can ask them to prioritize their user list so that you can finish the most critical task first and save the least important for the end.

Execution: Now when you know the customer's priority and have a master story list in place, you can finally start delivering some value to the customer. You start from the top, with the most prior task and work your way down to the bottom for the least prior task. After every delivery, you take feedback from your customer.

Updating Plan: As you start delivering the product to the customer, he will come up with some changes. You can include those changes in your list. If required, you will reprioritize the list of items. There might as well be a case where you are not fast enough and may not deliver the project in time. In such a case, you will always be delivering the highest priority and most important features. The prioritized list is used to define work order.

However, these practices and methods of Agile do not claim to solve every problem present in the software development industry because of the ever-changing environment and demands. But they help to establish a culture and provide an environment where tangible solutions emerge.

Why Agile?

In earlier days Waterfall Model was one of the most popular Software Development Models to deliver any project, but software developers faced many issues while using the waterfall model to develop new software. There were many drawbacks in the waterfall model which used to create difficulties around including the changes and the time and cost required for that.

To deal with these drawbacks, the Agile Software Development Model was suggested. In its initial phase, it was designed only to help any software development project to adapt to the client's and markets frequently changing requirements and to provide quick project completion.

How does Agile Help:

As customers and developers are involved in all the steps, this reduces the chances of any mistakes and end-result for the client is much more rewarding.

Below are the factors using which Agile addresses all the problems mentioned above:-

Characteristics	Agile approach
User requirement	Detailed, interactive inputs taken
Involvement of client	High engagement
Customer involvement	Customer involves as soon as the work starts
Escalation management	The entire team works to resolves problems as they occur
Model preference	Agile model favours adaption and accepts changes
Product or process	Less focused on formal and directive methods, so less documentation
Test documents	Comprehensive test planning before every value delivery
Effort estimation	Scrum master facilities and the team does the estimation
Reviews, Approvals, and feedback	Inspections are done after each iteration and feedback is also provided immediately. Requires fewer approvals.

Other factors that make Agile more adaptable include:

High Quality

More Productivity

Better Business Values

Fewer Costs

Quicker Time-to-Market

Considering the dynamic nature of businesses, where changes are happening every moment. Agile method is an excellent option for Software Development.

Agile Tools

Product name	Features and price	A good fit for
1. Wrike	• Split large goals into separate chunks, scale your project with dynamic request forms • Employ Gantt Charts, track time and budget • Communicate with your colleagues, get timely proofing and approval. • Look into granular group and team reporting. Price: Free/up to $34.60 per user per month	• Creative, marketing, business, IT teams without limitation.
2. Atlassian JIRA	• A major issue tracking and project management tool for agile teams. • Use customizable Scrum and Kanban boards to help with iterative delivery. • Track project performance with multiple predefined and custom reports. Price: $10 (up to 10 users/per user), $75 (from 11 users/per user)	• IT professionals, instructional designers, e-Learning authors working in a shared environment.
3. Freedcamp	• Get insights into current projects and workflows. • Log work time and bill fulfilled tasks Price: free (200MB storage)/from $2.49 per month (1GB storage).	• IT professionals, instructional designers, students and teachers working in a shared environment.
4. VersionOne	• Work with priorities, releases and sprints. • Connect with other team members, internal and external stakeholders, customers, etc. Price: free/paid plans from $29 per user per month	• IT professionals, tech savvy instructional designers
5. Taiga	• Track issues • Make video calls to fellow team members. • Employ built-in wikis for each project. Price: free/paid plans from $19 per user per month	• Agile developers and project managers, e-Learning authors working in a shared environment.

What is Scrum?

Scrum is a framework in which the team solves complex problems and comes up with productive & innovative solutions that are of the highest possible quality. It is a framework that assists the smooth & productive way to deliver end products & helps in building agile principles. Scrum is one of the most widely used frameworks which employs various processes and techniques.

Scrum and Agile are two terms that are often confused because Scrum works around continuous improvement, which is a core principle of agile.

Scrum is an agile way to manage a project say software development; It's a framework for getting work done, whereas agile is an attitude or a mind-set. You can't become "agile" as such, but you can always use a framework like Scrum to help you start thinking in a way that revolves around continuous improvement. Scrum assists in building agile principles into your everyday work.

Scrum is:

- Lightweight
- Simple to understand
- Difficult to master

Scrum Values

The entire Scrum framework is based on mainly 5 Scrum Values, which are discussed below. These values provide a standard to define what our actions & our work should adhere to. Hence, it's essential to reinforce theses values time and again.

- Commitment – This means dedication to providing high quality, working software with active collaboration.
- Courage – Means to accommodate client's new requirements & to deliver high-quality products every time, in every iteration.
- Focus – Means to concentrate on current priorities irrespective of their changeability.
- Openness – Means transparency in doing work with stakeholders, consistency within the team, openness to accept the new features.
- Respect – Means respecting the roles, norms, needs & expectations of various scrum partners.

Scrum Framework

The Scrum framework is heuristic in nature means it acknowledges the fact that a team doesn't know everything at the beginning of a project & the learning evolves gradually during a project. It is fabricated in a way where teams keep refreshing their priorities according to end-user requirements and continuously adjust to the fluctuating conditions.

Scrum is structured but not rigid in its application. Its application & execution can be customized to the needs of any organization.

The Scrum framework consists of:

- Scrum Artifacts
- Scrum Roles
- Scrum Rules
- Scrum Events

Scrum Artifacts – Scrum Framework

Artifacts are something that we create, e.g., a tool to solve a problem or a value that adds transparency.

The main idea of defining an artifacts is that everyone in a team should be on the same page. The scrum team continuously visits these 3 Artifacts to make sure the process is smooth & that everyone in the group has the same understanding of how to go about the inspection and adaptation.

There are three primary Scrum artifacts, namely:-

- Product Backlog
- Sprint Backlog
- Sprint Goal (or increment)

Product Backlog

The Product Backlog is a dynamic list of features, requirements, improvements, etc. which acts as an input for the Sprint Backlog. It is a customer's requirement that gets captured to understand the client's need.

It is, practically, the team's "To Do" list. Each item logged in the product backlog is referred to as a Product Backlog Item (PBI). Product Owner creates Product Backlog. We will understand more about the responsibilities of the Product Owner later in this article.

Sprint Backlog

Sprint Backlog is the list of finalized user stories, work items, bug fixes, etc., completed & selected by the scrum team for implementation & execution in the current sprint cycle.

It is created by the Scrum Team (discussed later).

Hence, Sprint Backlog is a further division of requirements that are gathered in a Product Backlog, which is given an effort-hour estimate, i.e., an estimate of the effort required to finish each task. This effort is measured in the number of hours.

Sprint Goal

The Sprint Goal is the useable end product derived from the completion of a sprint. It is also called as an increment. This sprint goal is demonstrated at the "End-of-Sprint Demo," which will be discussed in the chapter further.

It defines the objective that has to be met as decided by the development team. It is derived after a discussion between the Product Owner (PO) & the Scrum Team (Development Team).

Scrum Roles – Scrum Framework

Each Scrum Team consists of three leading roles that are indifferent to the job titles of those members in the organization. Irrespective of what your job title is, you can play any of the below roles in Scrum.

The three leading roles in a Scrum are:-

Scrum Roles

Each of these roles has a defined set of responsibilities, and successful completion of the project depends on how closely do they work with each other & their level of collaboration.

Let's discuss them one by one to know more about their responsibilities.

Product Owner

The Product Owner, also known as PO in its abbreviated form, is a central role within the Scrum Framework. He represents the end customer or stakeholders. He is the main character in agile testing, which creates stories and features from the requirements provided by the stakeholders (also referred to as Epics/ User Stories). There is always one product owner in any scrum team.

Three main functions of a product owner are:-

- Defines the product in the customer-focused terminology (typically user stories)
- Adds them to the product backlog
- Prioritizes them based on importance and dependencies

The primary responsibilities of a product owner are:-

- Defines and announces releases
- He communicates delivery and team status to the Scrum Team. This acts as a reminder for the Sprint Goals to be achieved.
- Shares progress with the Development Team basis how well the work has been done.
- Negotiates priorities etc.
- Ensures that the product backlog's visibility is high and bright.

Scrum Team

A Scrum Team is a group of individuals like developers, testers, etc. who work together to deliver the requested and committed products. The scrum team prioritizes the items from the Product Backlog that can be delivered & committed to. The Scrum Team is responsible for delivering the work to the customer.

Scrum teams, also known as Development Teams, are structured and empowered to self-organize and manage their work.

Main features of any Scrum Team are:-

- **Self-organizing** – Scrum teams are self-organizing means the teams know how to accomplish their work on their own, without outsiders' interference.
- **Cross-functional –** Scrum teams are cross-functional means they have all the competencies needed to accomplish the task without depending on non-team members.
- The Scrum team is designed to optimize flexibility, creativity, and productivity.

Scrum Master

A Scrum Master, also known as SM in its abbreviated form, helps the scrum team perform at their highest level. He protects the team from both internal and external distractions. His role is not of a project manager or product owner but of a facilitator who ensures that there are no blockages that can hamper the ability of the team to deliver products.

The primary responsibilities of a scrum master are as below:-

- Helps the PO maintain the product backlog.
- Assisting in the determination of the definition of done (DOD) for the product.
- Coaching the team, within the Scrum principles.
- Promoting self-organization within the team
- Facilitation of Scrum Events like
- Daily Scrum Meetings
- Sprint Planning Meetings
- Sprint Review Meetings
- Sprint Retrospective Meeting

Scrum Rules – Scrum Framework

Scrum uses an incremental & iterative approach so that the predictability is optimized & risk is controlled. Three pillars uphold every implementation of scrum process control:

Transparency, e.g., Common language for all participants, i.e., those working on the project & those who are going to check the work done should have an ear understanding of the term "finished."

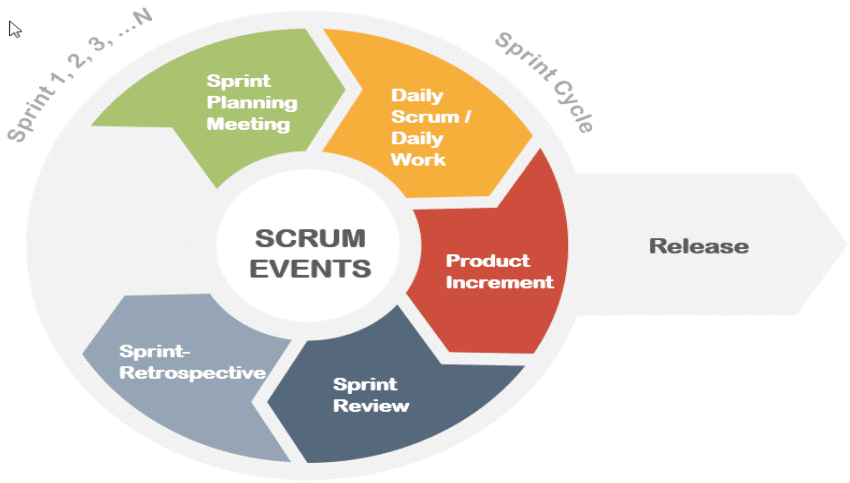
Inspection, e.g., regularly inspecting Scrum artifacts and progress toward a Sprint Goal to detect undesirable variances.

Adaptation, e.g., Following four things are a way to adapt to deviances:-

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Scrum Events – Scrum Framework

Scrum Events are prescribed events that are used to create regularity and to minimize the need for meetings which are not defined in a Scrum. All scrum events are time-boxed. These events are:



Before understanding these series of events let's see what a sprint is.

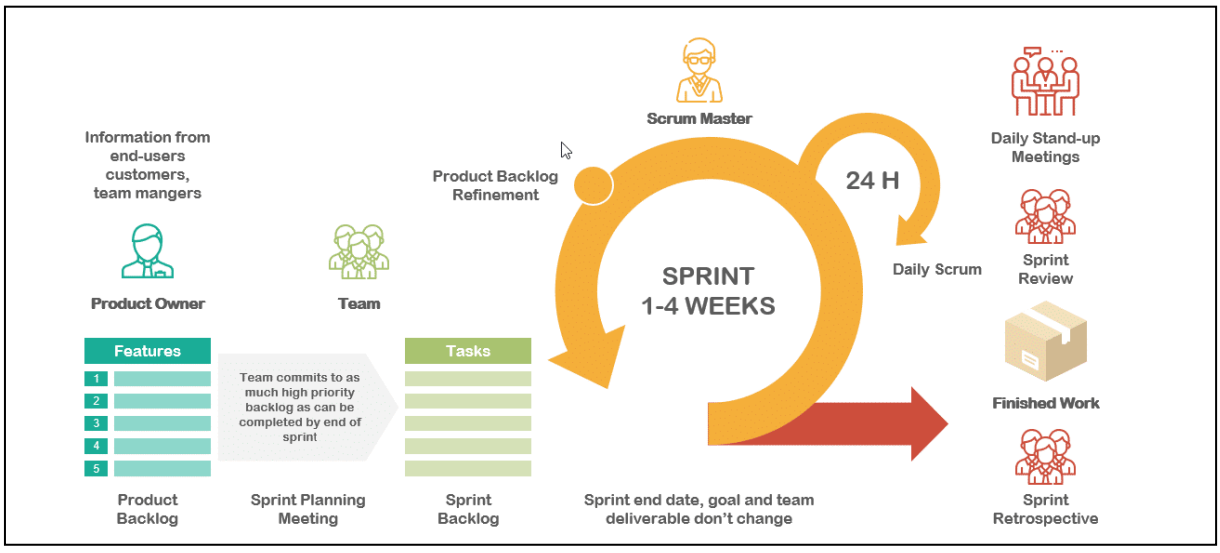
Scrum Framework – SPRINT

A sprint refers to a small, time-bound period in which a scrum team works to deliver the committed project to the user. Sprints are at the core of scrum and agile methodologies.

In the Scrum Framework, all activities which are shortlisted entries in the Scrum Backlog are executed in each Sprint. Sprints are also called 'Iterations.' Sprints are generally 2-4 weeks long, but their duration can be decided by the team consensually.

Once a Sprint starts, its duration is fixed and cannot be shortened or lengthened. The remaining events may end whenever the purpose of the Scrum event is achieved, ensuring judicious use of time.

Each Sprint follows a pre-defined series of steps, as shown below:



Scrum Framework

Each Sprint starts with two meetings to define the content of the Sprint:

- **WHAT-Meeting: Scrum Team commits to the User Stories from the Scrum Backlog.**
- **HOW-Meeting: To further divide the committed User Stories into smaller tasks**.

The culmination of these two meetings is also known as the Sprint Planning Meeting.

Under the deployment stage & during the Sprint, a short daily Stand-up-Meeting (Daily Scrum Meeting) is held to update the status of the user stories and to assist the self-organization of the team. Here, the Product Owner grooms the Product Backlog, also known as Product Backlog Grooming, wherein the requirements are prioritized and refined accordingly in consultation with the Scrum Team.

At the end of the Sprint, a Sprint Review Meeting is conducted to allow the Product Owner to check if all of the committed tasks are finished and implemented correctly. This is also called as DOD, i.e., Definition of Done, which means all the tasks under User Story have been accomplished & are ready to be checked by the PO.

Additionally, a Sprint Retrospective Meeting is conducted to understand what went well, what didn't, and what should be improved.

Now that we have discussed all the terminologies of a scrum process let's understand Scrum Events /

Scrum Process with an example.

Before starting the first Sprint

- Garry is assigned as the Product Owner of a new software project. One of his first tasks is to write down the high-level requirements called Epics, from the customer.
- Bossy is the Scrum Master.
- He writes them into the Product Backlog and initiates an estimation and prioritization session with senior developers & tester.
- Now he starts to break-down the high-level requirements into smaller user stories. Let's assume eight stories are created out of 1 Epic.
- With this list, he then plans & invites for the first Sprint Planning meeting.

Sprint 1 – Day 0

- During the Sprint Planning meeting, Garry presents the Product Backlog items from the highest priority to the lowest.
- The team clarifies open questions, and for each task, the team discusses if they have enough capacity, skills, etc.
- After this, they commit to complete the stories 1, 2,3,6,7 and 8 until the end of this sprint. Out of the total eight stories, numbers 4 & 5 cannot be done due to technical issues.
- After the Sprint Planning meeting Bossy – the Scrum Master of the team – calls the team to define the details of how the committed items are going to be implemented.
- The resulting tasks are written down & every member of the Scrum Team selects a task to work on.

Sprint 1 – Day 1

- In the morning the whole team comes together for their Daily Scrum Meeting.
- Everyone gives a short statement of what has been achieved so far, tells what he or she is planning to do today and shares blockages if any.
- Today one of the team members has a problem with licensing. Bossy checks if other team members have the same problem & say that he'll take care of that after the meeting.
- After 15 minutes, everyone goes back to work.

Sprint 1 – Day 2

- Today, again, the whole team carries out their Daily Scrum meeting.
- In the afternoon, one of the Scrum team members is stuck in a task. He calls Garry – Product Owner- and discusses.
- The scrum team mutually finds out a solution and carries on to work.

Sprint 1 – Day 14

- The final day of the first Sprint and Bossy –Scrum Master- has invited the team for the Sprint Review Meeting.
- The team has prepared a software. Garry – Product Owner- checks if the product meets his expectations and if the features are documented as required.
- At the end of the Review Session, he derives the following conclusions:
- Stories 1, 3, 6 are finished as expected
- Story 2 & 7 couldn't be finished in time
- Story 8 has some tasks that have to be redone
- In the afternoon, the team gets together for the Sprint Retro Meeting and discusses what went well during the sprint and what could be improved.
- One of the feedbacks is that the team faces trust issues. Bossy invites the people manager to guide the scrum team on mutual trust for better results.

Sprint 2 – Day 1

- Garry – Product Owner- adds new items to the Product Backlog based on his recent customer meetings.
- Moreover, he suggests additional items for the addition in story 8.
- Garry then invites the team for the Sprint Planning Meeting for Sprint 2.
- The team discusses and commits to stories with the guidance of Bossy –Scrum Master- and the second Sprint begins.

What Is DevOps?

DevOps is a term for a group of concepts that, while not all new, have catalysed into a movement and are rapidly spreading throughout the technical community. Like any new and popular term, people may have confused and sometimes contradictory impressions of what it is. Here's my take on how DevOps can be usefully defined; I propose this definition as a standard framework to more clearly discuss the various areas DevOps covers. Like "Quality" or "Agile," DevOps is a large enough concept that it requires some nuance to fully understand.

Definition in Depth

DevOps means a lot of different things to different people because the discussion around it covers a lot of ground. People talk about DevOps being "developer and operations collaboration," or it's "treating your code as infrastructure," or its "using automation," or "using kanban," or "a toolchain approach," or "culture," or a variety of seemingly loosely related items. The best way to define it in depth is to use a parallel method to the definition of a similarly complex term, agile development. Agile development, according to Wikipedia and the agile manifesto, consists of four different "levels" of concern. I've added a fifth, the tooling level – talk about agile and devops can get way too obsessed with tools, but pretending they don't exist is also unhelpful.

- **Agile Values –** Top level philosophy, usually agreed to be embodied in the Agile Manifesto. These are the core values that inform agile.
- **Agile Principles** – Generally agreed upon strategic approaches that support these values. The Agile Manifesto cites a dozen of these more specific principles. You don't have to buy into all of them to be Agile, but if you don't subscribe too many of them, you're probably doing something else.
- **Agile Methods** – More specific process implementations of the principles. XP, Scrum, your own homebrew process – this is where the philosophy gives way to operational playbooks of "how we intend to do this in real life." None of them are mandatory, just possible implementations.
- **Agile Practices** – highly specific tactical techniques that tend to be used in conjunction with agile implementations. None are required to be agile but many agile implementations have seen value from adopting them. Stand-ups, planning poker, backlogs, CI, all the specific artifacts a developer uses to perform their work.
- **Agile Tools** – Specific technical implementations of these practices used by teams to facilitate doing their work according to these methods. JIRA Agile (aka Green hopper), planningpoker.com, et al.

Ideally the higher levels inform the lower levels – people or organizations that pick up specific tools and practices without understanding the fundamentals may or may not see benefits but this "cargo cult" approach is generally considered to have suboptimal results. I believe the different parts of DevOps that people are talking about map directly to these same levels.

- **DevOps Values** – I believe the fundamental DevOps values are effectively captured in the Agile Manifesto – with perhaps one slight emendation to focus on the overall service or software fully delivered to the customer instead of simply "working software." Some previous definitions of DevOps, like Alex Honor's "People over Process over Tools," echo basic Agile Manifesto statements and urge devops collaboration.
- **DevOps Principles** – There is not a single agreed upon list, but there are several widely accepted attempts – here's John Willis coining "CAMS" and here's James Turnbull giving his own definition at this level. "Infrastructure as code" is a commonly cited DevOps principle. I've made a cut at "DevOps'ing" the existing agile manifesto and principles here. I personally believe that DevOps at the conceptual level is mainly just the widening of Agile's principles to include systems and operations instead of stopping its concerns at code checking.
- **DevOps Methods –** Some of the methods here are the same; you can use Scrum with operations, Kanban with operations, etc. (although usually with more focus on integrating ops

with dev, QA, and product in the product teams). There are some more distinct methods, like Visible Ops-style change control and using the Incident Command System for incident response. The set of these methodologies are growing; a more thoughtful approach to monitoring is an area where common methodologies haven't been well defined, for example.

- **DevOps Practices** –Specific techniques used as part of implementing the above concepts and processes. Continuous integration and continuous deployment, "Give your developers a pager and put them on call," using configuration management, metrics and monitoring schemes, a toolchain approach to tooling… Even using virtualization and cloud computing is a common practice used to accelerate change in the modern infrastructure world.
- **DevOps Tools** – Tools you'd use in the commission of these principles. In the DevOps world there's been an explosion of tools in release (Jenkins, Travis, teamcity), configuration management (puppet, chef, ansible, cfengine), orchestration (zookeeper, Noah, mesos), monitoring, virtualization and containerization (AWS, OpenStack, vagrant, Docker) and many more. While, as with Agile, it's incorrect to say a tool is "a DevOps tool" in the sense that it will magically bring you DevOps, there are certainly specific tools being developed with the express goal of facilitating the above principles, methods, and practices, and a holistic understanding of DevOps should incorporate this layer.

In the end, DevOps is a little tricky to define, just like its older brother Agile. But it's worth doing. To be a successful Agile or DevOps practitioner is to understand all the layers that go into it, and what a given DevOps implementation might contain or not contain. In the end, what DevOps hopes to bring to Agile is the understanding and practice that software isn't done until it's successfully delivered to a user and meets their expectations around availability, performance, and pace of change.

Specifically, I've come to believe there are three primary practice areas that are usually discussed in context of DevOps.

- **Infrastructure Automation** – create your systems, OS configs, and app deployments as code.
- **Continuous Delivery** – build, test, and deploy your apps in a fast and automated manner.
- **Site Reliability Engineering** – operate your systems; monitoring and orchestration, sure, but also designing for operability in the first place.

DevOps Toolchain

Since DevOps is collaborative across cross-functional teams, DevOps processes are structured to be cross-functional as well. Rather than using a single set of tools, there are multiple toolsets designed to promote collaborative, iterative software development.

There are seven stages in the software development and delivery process:

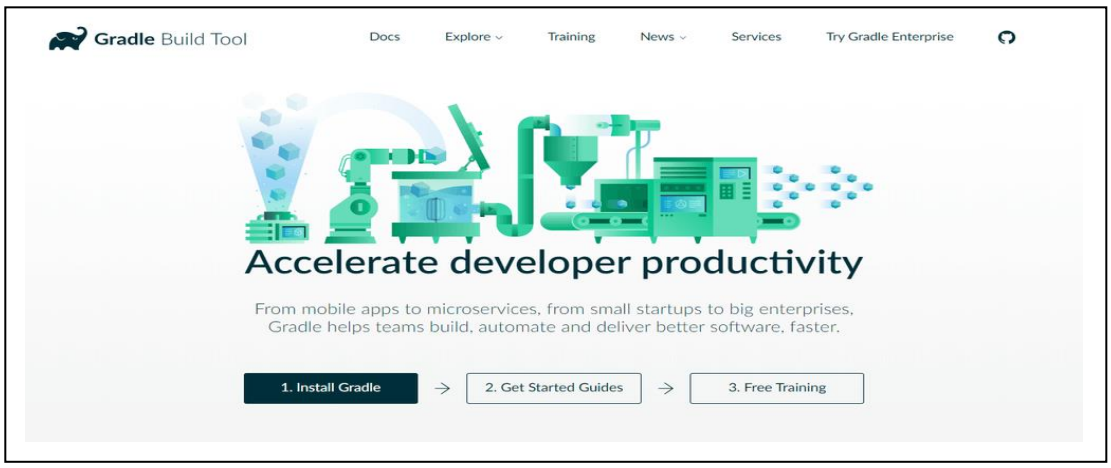
- **Plan:** Define the business value and application requirements. This involves a variety of IT personnel, including the application owners, software architects, security professionals, and the team responsible for maintaining the IT infrastructure.
- **Create:** This includes building and coding the application, and configuring the software development process.
- **Verify:** Testing and quality control is part of each development cycle, including acceptance testing, regression testing, security analysis, and performance testing.
- **Packaging:** Following testing, each software build is made ready for deployment. This step, also referred to as staging, requires approvals and steps to prepare for release. These first four stages are part of application development, the "Dev" of DevOps.
- **Release:** This is a transitional stage that hands the application from development to operations—"Ops"—for evaluation.
- **Configure:** Operations assume responsibility for IT provisioning and configuration.
- **Monitoring:** Once the software is released, it has to be monitored to understand the impact on end users, including performance problems. Feedback is offered to inform the next software release.

DevOps tools

1. Gradle

Gradle: DevOps tool

Your DevOps tool stack will need a reliable build tool. Apache Ant and Maven dominated the automated build tools market for many years, but Gradle showed up on the scene in 2009, and its popularity has steadily grown since then. Gradle is an incredibly versatile tool which allows you to write your code in Java, C++, Python, or other languages. Gradle is also supported by popular IDEs such as Netbeans, Eclipse, and IntelliJ IDEA. If that doesn't convince you, it might help to know that Google also chose it as the official build tool for Android Studio.



While Maven and Ant use XML for configuration, Gradle introduces a Groovy-based DSL for describing builds. In 2016, the Gradle team also released a Kotlin-based DSL, so now you can write your build scripts in Kotlin as well. This means that Gradle does have some learning curves, so it can help a lot if you have used Groovy, Kotlin or another JVM language before. Besides, Gradle uses Maven's repository format, so dependency management will be familiar if you have prior experience with Maven. You can also import your Ant builds into Gradle.

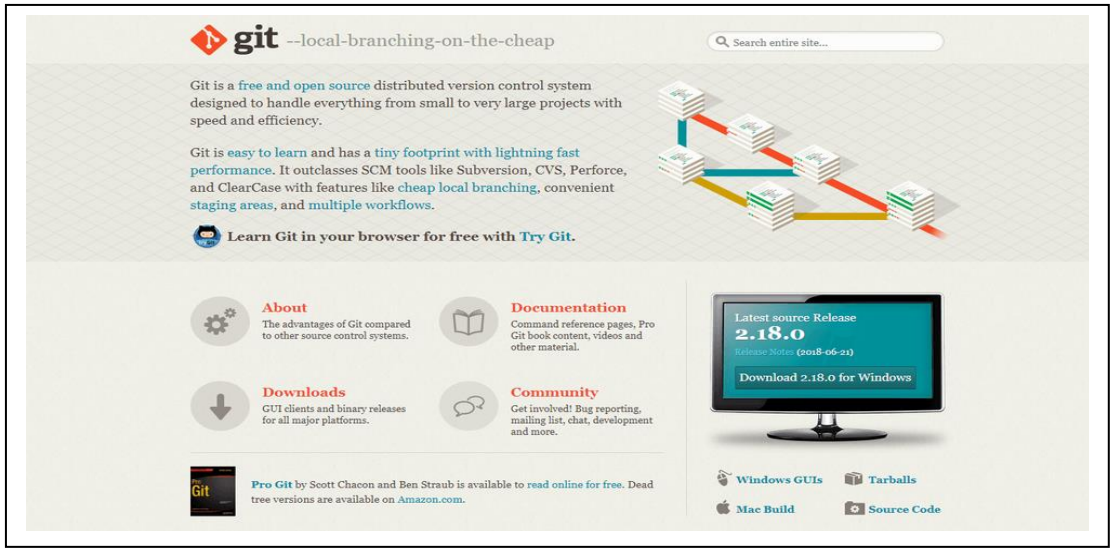
The best thing about Gradle is incremental builds, as they save a nice amount of compile time. According to Gradle's performance measurements, it's up to 100 times faster than Maven. This is in part because of incrementality, but also due to Gradle's build cache and daemon. The build cache reuses task outputs, while the Gradle Daemon keeps build information hot in memory in-between builds.

All in all, Gradle allows faster shipping and comes with a lot of configuration possibilities.

2. Git

Git: DevOps tool

Git is one of the most popular DevOps tools, widely used across the software industry. It's a distributed SCM (source code management) tool, loved by remote teams and open source contributors. Git allows you to track the progress of your development work. You can save different versions of your source code and return to a previous version when necessary. It's also great for experimenting, as you can create separate branches and merge new features only when they're ready to go.



To integrate Git with your DevOps workflow, you also need to host repositories where your team members can push their work. Currently, the two best online Git repo hosting services are GitHub and Bitbucket. GitHub is more well-known, but Bitbucket comes with free unlimited private repos for small teams (up to five team members). With GitHub, you get access only to public repos for free—which is still a great solution for many projects.

Both GitHub and Bitbucket have fantastic integrations. For example, you can integrate them with Slack, so everyone on your team gets notified whenever someone makes a new commit.

3. Jenkins

Jenkins: DevOps tool

Jenkins is the go-to DevOps automation tool for many software development teams. It's an open source CI/CD server that allows you to automate the different stages of your delivery pipeline. The main reason for Jenkins' popularity is its huge plugin ecosystem. Currently, it offers more than 1,000 plugins, so it integrates with almost all DevOps tools, from Docker to Puppet.

With Jenkins, you can set up and customize your CI/CD pipeline according to your own needs. I found the following example in the Jenkins Docs. And, this is just one of the possibilities. Nice, isn't it?

It's easy to get started with Jenkins, as it runs out-of-the-box on Windows, Mac OS X, and Linux. You can also easily install it with Docker. You can set up and configure your Jenkins server through a web interface. If you are a first-time user, you can choose to install it with frequently used plugins. However, you can create your own custom config as well.

With Jenkins, you can iterate and deploy new code as quickly as possible. It also allows you to measure the success of each step of your pipeline. I've heard people complaining about Jenkins' "ugly" and non-intuitive UI. However, I could still find everything I wanted without any problem.

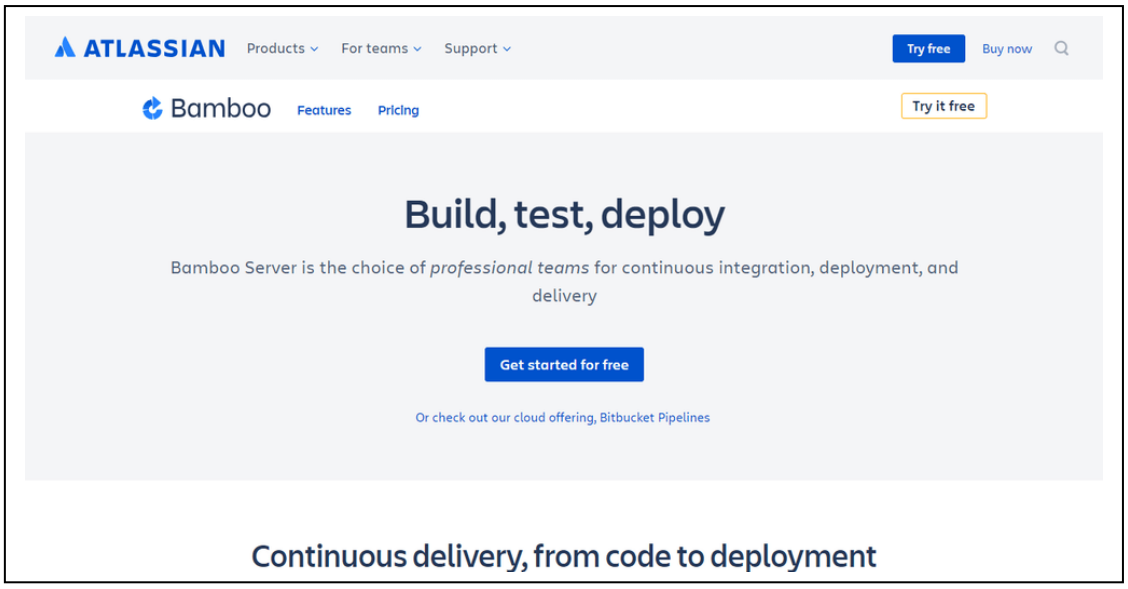
4. Bamboo

Bamboo: DevOps tool

Bamboo is Atlassian's CI/CD server solution that has many similar features to Jenkins. Both are popular DevOps tools that allow you to automate your delivery pipeline, from builds to deployment. However, while Jenkins is open source, Bamboo comes with a price tag. So, here's the eternal question: is it worth choosing proprietary software if there's a free alternative? It depends on your budget and goals.

Bamboo has many pre-built functionalities that you have to set up manually in Jenkins. This is also the reason why Bamboo has fewer plugins (around 100 compared to Jenkins' 1000+). In fact, you don't need that many plugins with Bamboo, as it does many things out-of-the-box.

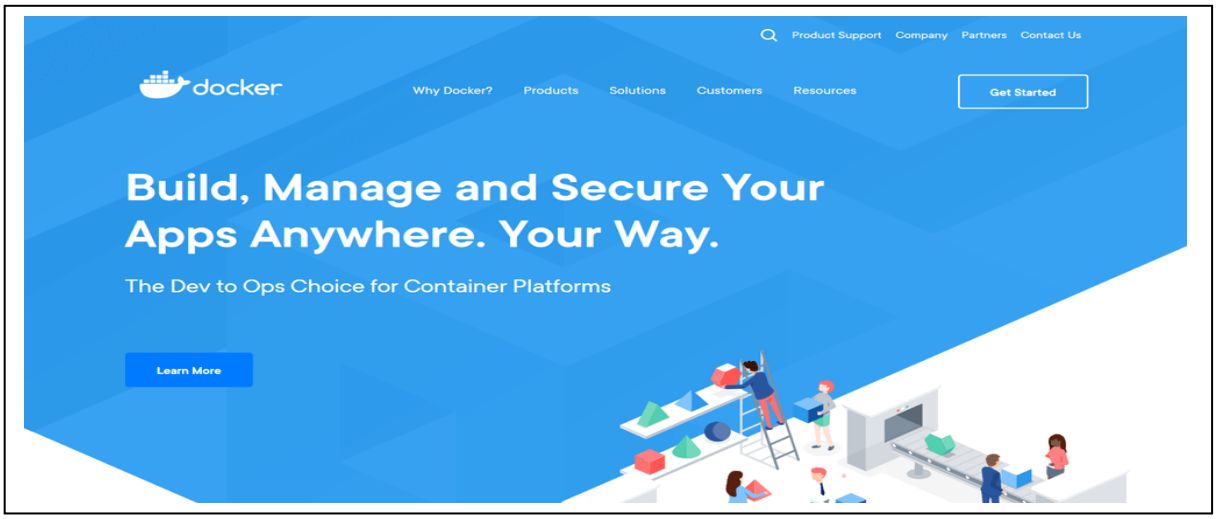
Bamboo seamlessly integrates with other Atlassian products such as Jira and Bitbucket. You also have access to built-in Git and Mercurial branching workflows and test environments. All in all, Bamboo can save you a lot of configuration time. It also comes with a more intuitive UI with tooltips, auto-completion, and other handy features.



5. Docker

Docker: DevOps tool

Docker has been the number one container platform since its launch in 2013 and continues to improve. It's also thought of as one of the most important DevOps tools out there. Docker has made containerization popular in the tech world, mainly because it makes distributed development possible and automates the deployment of your apps. It isolates applications into separate containers, so they become portable and more secure. Docker apps are also OS and platform independent. You can use Docker containers instead of virtual machines such as VirtualBox.



What I like the most about Docker is that you don't have to worry about dependency management. You can package all dependencies within the app's container and ship the whole thing as an independent unit. Then, you can run the app on any machine or platform without a headache.

Docker integrates with Jenkins and Bamboo, too. If you use it together with one of these automation servers, you can further improve your delivery workflow. Besides, Docker is also great for cloud computing. In recent years, all major cloud providers such as AWS and Google Cloud added support for Docker. So, if you are planning a cloud migration, Docker can ease the process for you.