**Table of Contents:**

Contents

[Day 1-2 4](#_Toc108687268)

[ASQA Engineer Training 4](#_Toc108687269)

[Main Objectives for an SQA Engineer 5](#_Toc108687270)

[Software Quality 5](#_Toc108687271)

[SDLC vs STLC: 6](#_Toc108687272)

[SDLC: 6](#_Toc108687273)

[STLC: 6](#_Toc108687274)

[BRD vs SRS vs FSD: 7](#_Toc108687275)

[1. Business Requirement Document (BRD): 7](#_Toc108687276)

[2. Software Requirement Specification (SRS): 7](#_Toc108687277)

[3. Functional Specifications Document (FSD): 7](#_Toc108687278)

[Days 3-5 8](#_Toc108687279)

[QA vs QC 8](#_Toc108687280)

[What is Quality Assurance (QA)? 8](#_Toc108687281)

[What is Quality Control (QC)? 8](#_Toc108687282)

[What is The Difference between QA/QC? 8](#_Toc108687283)

[Defect Life Cycle 9](#_Toc108687284)

[States in Defect Life Cycle: 9](#_Toc108687285)

[Bug Reporting 10](#_Toc108687286)

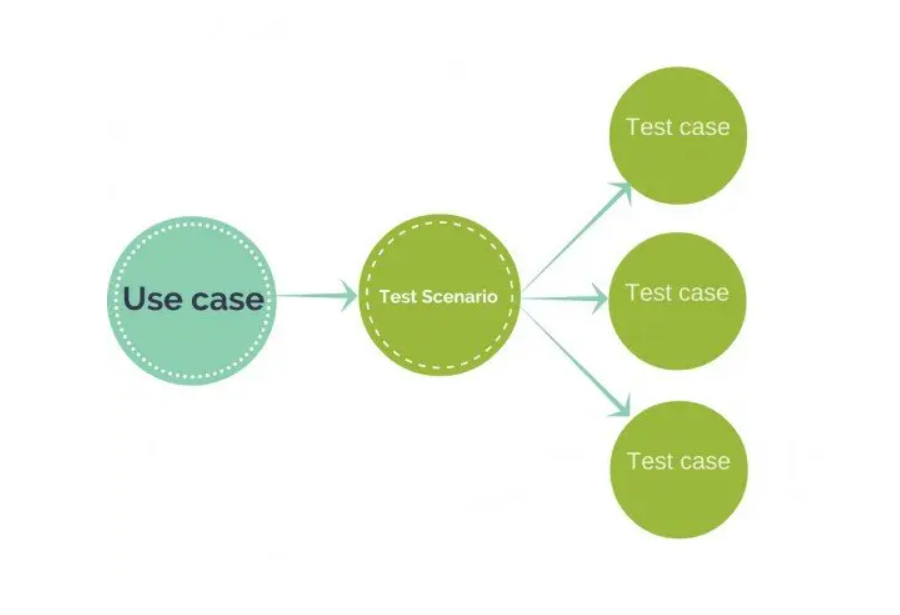
[Bug Triage: 10](#_Toc108687287)

[Test Plan and Test Strategy: 10](#_Toc108687288)

[Test Plan: 10](#_Toc108687289)

[Test Strategy: 10](#_Toc108687290)

[Use Cases vs Test Scenarios vs Test Cases: 11](#_Toc108687291)

[ Use Case: 11](#_Toc108687292)

[Test Scenario: 12](#_Toc108687293)

[Test Case: 12](#_Toc108687294)

[Days 6-7 13](#_Toc108687295)

[**Test Suite - Creation and Maintenance**: 13](#_Toc108687296)

[**Test Suite:** 13](#_Toc108687297)

[TDD vs BDD: 13](#_Toc108687298)

[Types of Testing in Software Engineering 15](#_Toc108687299)

[Testing Types 16](#_Toc108687300)

[Functional Testing: 16](#_Toc108687301)

[Reading (Optional, no extra time): 17](#_Toc108687302)

[Non-Functional Testing: 17](#_Toc108687303)

[Days 8-10 18](#_Toc108687304)

[Automation Tools: 18](#_Toc108687305)

[API Testing: 18](#_Toc108687306)

[REST vs SOAP APIs: 19](#_Toc108687307)

[SOAP (Simple Object Access Protocol) 19](#_Toc108687308)

[API Testing Tools: 20](#_Toc108687309)

[**Postman** 20](#_Toc108687310)

[API Documentation: 20](#_Toc108687311)

[Days 11-12 21](#_Toc108687312)

[Load/Stress Testing Tools: 21](#_Toc108687313)

[CI/CD 21](#_Toc108687314)

[Continuous Integration (CI) 21](#_Toc108687315)

[Continuous Delivery (CD) 21](#_Toc108687316)

[CI/CD Tools and Frameworks: 22](#_Toc108687317)

[Database Testing: 22](#_Toc108687318)

[SQL Injection: 22](#_Toc108687319)

[Days 13-14 23](#_Toc108687320)

[Penetration Testing: 23](#_Toc108687321)

[JS Attacks/Injection: 23](#_Toc108687322)

[Why is it Important to Test JS Injection? 23](#_Toc108687323)

[Software Development and Version Controlling: 24](#_Toc108687324)

[Days 15-17 24](#_Toc108687325)

[Documentation Tools/Platforms 24](#_Toc108687326)

[Confluence by JIRA: 24](#_Toc108687327)

[Days 18-19 25](#_Toc108687328)

[QA Management Tools/Plugins: 25](#_Toc108687329)

[Effective Client Communication: 26](#_Toc108687330)

[Client Communication Tools: 26](#_Toc108687331)

[Slack: 26](#_Toc108687332)

[Microsoft Teams: 26](#_Toc108687333)

[Other tools: 27](#_Toc108687334)

[Random: 27](#_Toc108687335)

# Day 1-2

## ASQA Engineer Training

The main focus and purpose of this training document is to ensure that you get the main idea as well as hands-on experience of the Software Quality Assurance essentials for your professional and personal development to grow your career as an SQA Engineer.

You are advised to properly read and implement the concepts discussed in this document and ask your work buddy in case of any queries/confusion.

Moreover, please don’t forget to explore the provided links and video tutorials when required. Self-learning is highly encouraged.

1. **Explain the Roles and Responsibilities**

* Briefing Job Description
* Primary Role of A-SQA Engineer
* Quality Management

1. **Explain Thread Management**

* Profile Management
* Effective Client Communication
* Team Coordination

1. **Team Structure and Hierarchies across Teams**

* Explain Team Structure
* Introduction to the Devsinc’s Culture
* Responsibilities towards the Devsinc family

1. **Familiarity with different business processes being followed**

* Explaining the different business processes being followed across many teams in the Devsinc family
* Getting to know the basic requirements and responsibilities within a certain team or across multiple teams in Devsinc
* Familiarity with ‘Reporting to’ mechanism

1. **Introduction to the QA Department in Devsinc**

* Goals of the QA Department
* Goals for oneself during the training and after the training
* Focus on the core ideology of the QA Department to achieve the essence of it

## Main Objectives for an SQA Engineer

* How to **bridge the gap of testing** from a developer’s perspective vs. from a tester’s perspective.
* **Effective Communication** across each end and Norms for different Cultures
* Knowledge of **Manual and Automated** Testing - where to use what?
* **When to stop testing**? Knowing when testing is done?
* **Bugs that unit tests** can expose and ones that they will miss
* **How to leverage** (as a tester, not a developer) unit tests to find more bugs earlier
* **How to plan testing** among several development models
* Working knowledge of the **Testing Tools** but what are Testing Tools?
* Designing and developing the **Test Suites**
* New and sometimes f**orgotten strategies** and test design methods including attack-based testing, model-based testing, and keyword-driven testing
* Why is **documentation** important for an SQA Engineer?
* How do you select the most **effective practices** to find bugs, optimize test planning and execution time?
* Understanding of different **types of Documentations**.
* Components/Attributes of **effective bug logging**.
* **Handling test cases** with repetition.
* Extending functionality in depth and finding out the **edge/corner cases** for effective testing.
* **Detailed regression** testing approaches.
* **Acceptance criteria** and their impact.

## Software Quality

It's a very subjective term to discuss. People have a perception of quality with respect to their own learning and experiences. It depends upon the purpose that we want to achieve, and It is relative with respect to time.

**Anything that creates comfortability for the end-user could be termed quality.  
If Correctness, Usefulness, and Goodness are woven together into a product, That product could be of good quality.**  
  
**Software Testing** is the art through which a tester evaluates a product in order to verify requirements and mitigate risks so that stakeholders could make better-informed decisions.  
It is the process of executing a program or application with the intent of finding software bugs.  
  
**According to ANSI/IEEE 1059 standard – A process of analyzing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item.**  
 **Why Software testing** is needed and carried out:

1. For error Identification.
2. To avoid risks and extra costs.
3. To ensure customer satisfaction.
4. For understanding what kind of value can be provided to customers.
5. To explore how comfortable it is to use the software and its overall usability.
6. To verify if it has any hidden features/bugs.
7. For assessing the robustness of the software.
8. For business optimization.

##### Task 1:

Prepare a document stating why software quality is important, and what impact it has over the development teams?

## SDLC vs STLC:

### SDLC:

SDLC **(Software Development Life Cycle)** defines all the standard phases which are involved during the software development process. It is a process of developing software in a phased manner in the following order.

1. Requirements Gathering
2. Design the software
3. Build the Software
4. Test the software
5. Deployment
6. Maintenance.

### STLC:

**Software Testing Life Cycle** (STLC) is the testing process that is executed in a well-planned manner. In the STLC process, various activities are carried out to improve the quality of the product. However, **STLC phases only deal with testing and detecting errors** but not development itself.

Different companies define different phases in STLC. However, the generic Software Test Life Cycle has the following stages.

1. Requirement Analysis
2. Test Planning
3. Test Development
4. Test Environment Setup
5. Test Execution & Closure

#### Useful links:

1. [SDLC VS STLC](https://www.guru99.com/sdlc-vs-stlc.html)
2. [SDLC vs STLC](https://www.softwaretestingmaterial.com/sdlc-vs-stlc)

##### Task 2:

Prepare a document listing the parallel comparison of the steps in SDLC and STLC, with examples.

## BRD vs SRS vs FSD:

The role of formulating a document is to understand fundamentals that will be compelled to develop robust software. Type of record expectation depends upon business type, their criteria, how the company processes, and what kind of software is to be developed.

### 1. Business Requirement Document (BRD):

A business requirements document describes the business solution for a project (i.e., what a new or updated product should do), including the user's needs(requirements) and expectations.

### 2. Software Requirement Specification (SRS):

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill all stakeholders’ (business, users) needs.

### 3. Functional Specifications Document (FSD):

A functional specification is a formal document used to describe a product's intended capabilities, appearance, and interactions with users in detail for software developers.

#### Useful links:

1. [BRS vs SRS: Know the Difference](https://www.guru99.com/brs-vs-srs-the-myth-busted.html)
2. [What is a Functional Specification Document?](https://searchsoftwarequality.techtarget.com/definition/functional-specification)
3. [What is a Functional Requirement? Specification, Types, EXAMPLES](https://www.guru99.com/functional-requirement-specification-example.html)

##### Task 3:

Prepare a document and enlist 3 major differences between BRD, SRS, FSD, with examples.

# Days 3-5

## QA vs QC

### What is Quality Assurance (QA)?

Quality assurance (QA) is a way of preventing mistakes and defects in manufactured products and avoiding problems when delivering products or services to customers; which [ISO 9000](https://en.wikipedia.org/wiki/ISO_9000) defines as **"part of** [**quality management**](https://en.wikipedia.org/wiki/Quality_management) **focused on providing confidence that quality requirements will be fulfilled".**

**Quality planning**

**Quality Assurance**

**Quality control**

**Quality improvement**

### What is Quality Control (QC)?

Quality control (QC) is a process by which entities review the quality of all factors involved in production. [ISO 9000](https://en.wikipedia.org/wiki/ISO_9000) defines quality control as **"A part of quality management focused on fulfilling quality requirements"**.

### What is The Difference between QA/QC?

QA and QC are closely related concepts and both are **aspects of quality management**, they are fundamentally different in their focus: QC is used to verify the quality of the output; QA is the process of managing for quality.  
**Example:**  
Quality Assurance is like following all the Quality processes for any project or product from development to deployment by following the Quality standards like ISO standards. Quality Control is like testing the deliverable product/project. it should be according to the client’s requirements.

Examples of quality assurance activities include process checklists, process standards, process documentation, and project audit. Examples of quality control activities include inspection, deliverable peer reviews, and the software testing process.  
  
**Real-World Example:**  
Let’s suppose we are developing any Software/Web App. Quality Assurance in this is that we should follow the quality standards/format at every stage like development(best coding practices), SRS/Test Planning document should be according to the industry standards. In QC we only perform the testing to make sure that it should not have any bugs.

#### Useful links:

1. [Quality Assurance Vs Quality Control: What's the Difference?](https://www.guru99.com/quality-assurance-vs-quality-control.html)
2. [Quality Assurance vs. Quality Control: What's the Difference?](https://www.qualio.com/blog/quality-assurance-vs-quality-control)
3. [Difference Between QA and QC](https://www.youtube.com/watch?v=j_KdgdLscU4)

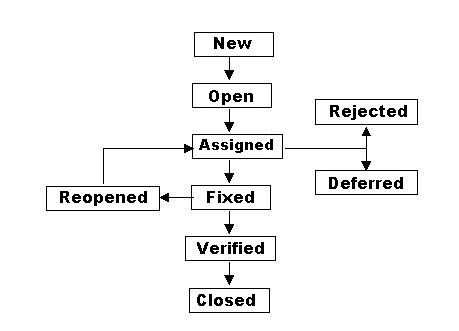
##### Task 4:

Statement Coverage and Path Finding is performed in QA or QC? Explain with justifications according to your understanding.

## Defect Life Cycle

The [defect](https://www.guru99.com/defect-management-process.html) life cycle, **also known as the Bug Life cycle** is the journey of a defect cycle, which a defect goes through during its lifetime.

Defect Life Cycle - Workflow:



### States in Defect Life Cycle:

* **New** - Potential defect that is raised and yet to be validated.
* **Assigned** - Assigned against a development team to address it but not yet resolved.
* **Active** - The Defect is being addressed by the developer and the investigation is in progress. At this stage there are two possible outcomes; viz - Deferred or Rejected.
* **Test** - The Defect is fixed and ready for testing.
* **Verified** - The Defect that is retested and the test has been verified by QA.
* **Closed** - The final state of the defect that can be closed after the QA retesting or can be closed if the defect is duplicate or considered as NOT a defect.
* **Reopened** - When the defect is NOT fixed, QA reopens/reactivates the defect.
* **Deferred** - When a defect cannot be addressed in that particular cycle it is deferred to a future release.
* **Rejected** - A defect can be rejected for any of the 3 reasons; viz - duplicate defect, NOT a Defect, Non Reproducible.

#### Useful Links:

1. [Defect Life Cycle](https://www.tutorialspoint.com/software_testing_dictionary/defect_life_cycle.htm)
2. [Defect Life Cycle(blog)](http://istqbrm.blogspot.com/2014/07/defect-life-cycle.html)
3. [Defect life Cycle](https://www.getsoftwareservice.com/defect-life-cycle/)

## Bug Reporting

[How to report a Bug?](https://www.softwaretestingnews.co.uk/write-effective-bug-report/)

##### Task 5:

Identify and report maximum bugs present on<https://www.demoblaze.com/> by performing manual testing using the bug reporting template, spend ~2-3 hours on this activity and try to find maximum bugs and cover maximum functionality in maximum depth.

## Bug Triage:

The term Triage is used in Software Testing to define **the severity and priority of defects**. Defect triage is a process or mechanism where each defect is prioritized based on its severity, risk, reoccurrence, etc.

#### Useful Links:

* [Bug triage](https://blog.testlodge.com/bug-triage/)
* [Defect triage process meeting](https://www.softwaretestinghelp.com/defect-triage-process-meeting/)

## Test Plan and Test Strategy:

### Test Plan:

A test plan is **a document detailing the objectives, resources, and processes** for a specific test for a [software](https://en.wikipedia.org/wiki/Software) or hardware product. The plan typically contains a detailed understanding of the eventual [workflow](https://en.wikipedia.org/wiki/Workflow).   
Example: [Test Plan Template: Sample Document with Web Application Example](https://www.guru99.com/test-plan-for-project.html)

### Test Strategy:

**Test Strategy** is a plan for **defining an approach to the Software Testing Life Cycle** (STLC). It guides QA teams to define [Test Coverage](https://www.guru99.com/test-coverage-in-software-testing.html) and testing scope. It helps testers get a clear picture of the project at any instance. The possibility of missing any test activity is very low when there is a proper test strategy in place.

#### Useful links:

1. [Test Plan vs Test Strategy](https://www.guru99.com/test-plan-v-s-test-strategy.html)
2. [How to create a Test Strategy](https://www.guru99.com/how-to-create-test-strategy-document.html)
3. [How to create a Test Plan](https://www.guru99.com/test-plan-for-project.html)
4. [Test Plan Templates](https://strongqa.com/qa-portal/testing-docs-templates/test-plan)

## Use Cases vs Test Scenarios vs Test Cases:

### Use Case:

**A use case is a description of how a person who actually uses that process or system will accomplish a goal**. It's typically associated with software systems, but can be used in reference to any process. For **example**, imagine you're a cook who has a goal of preparing a grilled cheese sandwich. The use case would describe through a series of written steps how the cook would go about preparing that sandwich. A use case helps you understand where errors could occur in the process and design features to resolve those errors.

### Test Scenario:

**A Test Scenario is a statement describing the functionality of the application to be tested**. It is used for end-to-end testing of a feature and is generally derived from the use cases.  
  
“Test scenarios can serve as the basis for lower-level test case creation. A single test scenario can cover one or more test cases. Therefore a test scenario has a one-to-many relationship with the test cases.”

### Test Case:

A **Test case is a set of actions executed to verify a particular feature or functionality of your software application**. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenarios to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.

##### Examples of Test Scenario and Test Cases:

As an example, consider a test scenario – “Verify that the user is not able to login with incorrect credentials”. Now, this test scenario can be further broken down into multiple test cases like-

1. Checking that users with the correct username and incorrect password should not be allowed to log in.
2. Checking that a user with an incorrect username and correct password should not be allowed to log in.
3. Checking that a user with an incorrect username and incorrect password should not be allowed to log in.

#### Useful links:

1. [A Quick Guide to Test Scenario & Test Cases in Software Testing](https://reqtest.com/testing-blog/test-scenario-test-case/)
2. [Test Case vs Test Scenario: What's the Difference?](https://www.guru99.com/test-case-vs-test-scenario.html)
3. [23 thoughts on “What is the difference between Test Cases vs Test Scenarios?”](https://www.softwaretestingclass.com/what-is-difference-between-test-cases-vs-test-scenarios)
4. [Test Scenario Vs Test Case](https://www.youtube.com/watch?v=ePGAEJURzqU)
5. [Test Scenario Template](https://docs.google.com/spreadsheets/d/1RvR54gywTcrZQy-fOpr5NTbQceWg5edhwZtbLBvtS5A/edit#gid=0)
6. [How to Write Test Cases: Sample Template with Examples](https://www.guru99.com/test-case.html)

##### Task 6a:

From your work in task 5, continue with the same and prepare Use cases, test scenarios and test cases for the same.

# 

# Days 6-7

## **Test Suite - Creation and Maintenance**:

### **Test Suite:**

**A Test Suite is a collection of test cases**. In automated testing, it can mean a collection of test scripts. In a test suite, the test cases/scripts are organized in a logical order. For example, the test case for registration will precede the test case for login.

For example, you could have a test suite for each of the core features of the software or you could have a separate test suite for a particular type of testing (for example, smoke test suite or security test suite).

An example of a Test suite for purchasing a product could comprise of the following test cases:

* Test Case 1: **Login**
* Test Case 2: **Add Products**
* Test Case 3: **Checkout**
* Test Case 4: **Logout**

Note that each of the test cases above is dependent on the success of the previous test cases. For instance, it’s no use checking out if one cannot add products. Hence, if you are running a test suite in sequential mode, you can choose to stop the test suite execution if a single test case does not pass.

#### Useful Links:

1. [Test suite](https://www.tutorialspoint.com/software_testing_dictionary/test_suite.htm)
2. [Test Suite Fundamentals](https://softwaretestingfundamentals.com/test-suite/)
3. [Difference between test-case test-suite test-run](https://www.testmonitor.com/blog/test-case-test-suite-test-run-whats-the-difference)

##### Task 6b:

Continue with work in tasks 5 and 6a, and create a Test Suite (A folder in your google drive and sheets within that folder) containing the test scenarios, test cases, and relevant stuff.

## TDD vs BDD:

Test-Driven Development (TDD) is a development practice while Behavioral Driven Development (BDD) is a team methodology. In TDD, the developers write the tests while in BDD the automated specifications are created by users or testers (with developers wiring them to the code under test). For small, co-located, developer-centric teams, TDD and BDD are effectively the same.

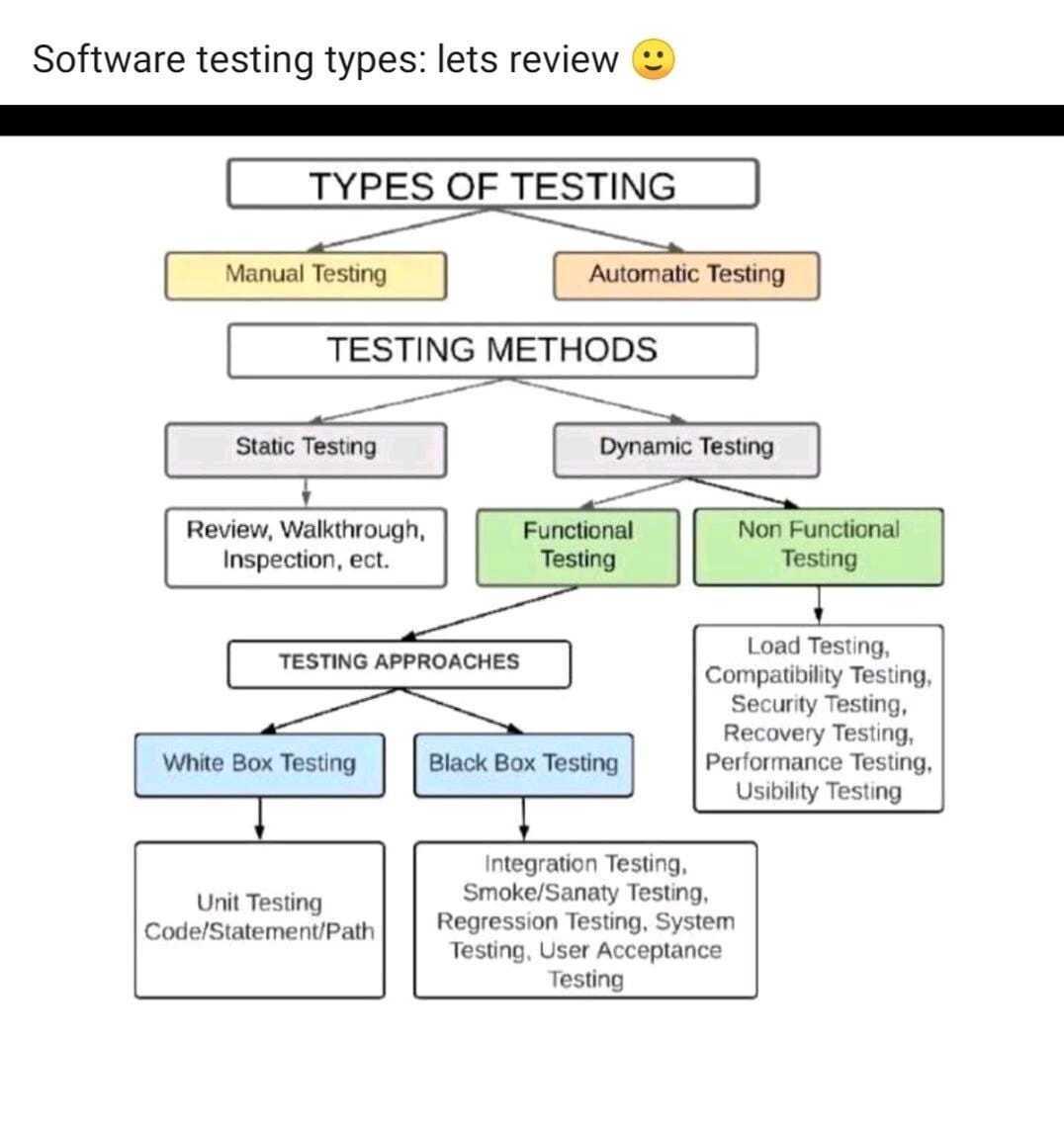
Useful links:

1. [TDD vs BDD: What's the Difference?](https://www.pluralsight.com/blog/software-development/tdd-vs-bdd)
2. [TDD Vs BDD - Analyze The Differences With Examples](https://www.softwaretestinghelp.com/tdd-vs-bdd)
3. [TDD vs BDD vs ATDD : Key Differences](https://www.browserstack.com/guide/tdd-vs-bdd-vs-atdd)

##### Task 7:

Prepare a short summary describing the advantages and disadvantages of TDD and BDD.

## Types of Testing in Software Engineering



#### Useful Links:

[Types of Software Testing Video Tutorial](https://www.youtube.com/watch?v=T0TynxN77oY)

### Testing Types

#### Static Testing:

[Static Testing](https://www.guru99.com/testing-review.html) is a software testing technique that is used to **check defects in software applications without executing the code**. Static testing is done to avoid errors at an early stage of development as it is easier to identify the errors and solve the errors. It also helps to find errors that may not be found by Dynamic Testing.

#### Dynamic testing:

[Dynamic Testing](https://www.guru99.com/dynamic-testing.html.) is a software testing method used to **test the dynamic behavior of software code.** The main purpose of dynamic testing is to test software behavior with dynamic variables or variables which are not constant and finding weak areas in the software runtime environment. The code must be executed in order to test the dynamic behavior.

**Example**:  
Let's suppose we have a submit button to submit the form. Static Testing is like that we should make sure that the submit button’s color and size should be according to designs given by the client. Dynamic Testing in this regard would be to make sure that the submit button should submit the form. Dynamic Testing is further divided into Functional and Non Functional testing.

### Functional Testing:

FUNCTIONAL TESTING is a type of software testing that **validates the software system against the functional requirements/specifications**. Following are the approaches of functional testing:

#### 1. Black box testing:

Testing, either functional or non-functional, **without reference to the internal structure** of the component or system.

**Black-box test design technique:** Procedure to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.

**Example:** A tester, without knowledge of the internal structures of a website, tests the web pages by using a browser; providing inputs (clicks, keystrokes), and verifying the outputs against the expected outcome.

#### 2. White-box testing:

Testing based on an analysis of the **internal structure of the component or system.**

**White-box test design technique:** Procedure to derive and/or select test cases based on an analysis of the internal structure of a component or system.

**Example:** A tester, usually a developer as well, studies the implementation code of a certain field on a webpage, determines all legal (valid and invalid) AND illegal inputs, and verifies the outputs against the expected outcomes, which is also determined by studying the implementation code.

White Box Testing is like the work of a mechanic who examines the engine to see why the car is not moving.

### Reading (Optional, no extra time):

The Art of Software Testing [Book by Glenford Myers](http://barbie.uta.edu/~mehra/Book1_The%20Art%20of%20Software%20Testing.pdf)

### Non-Functional Testing:

NON-FUNCTIONAL TESTING is defined as a type of Software testing to check **non-functional aspects (performance, usability, reliability, robustness, etc) of a software application**. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing.

#### Performance Testing

[Performance Testing](https://www.guru99.com/performance-testing.html#:~:text=Performance%20Testing%20is%20a%20software,software%20application%20under%20particular%20workload.) is a software testing process used for **testing the speed, response time, stability, reliability, scalability, and resource usage of a software application** under a particular workload.

It has the following types:

##### Load Testing

[Load Testing](https://www.guru99.com/load-testing-tutorial.html) is a non-functional software testing process in which the performance of a software application is tested under a **specific expected load**

##### Stress Testing

[Stress Testing](https://www.guru99.com/stress-testing-tutorial.html) is a type of software testing that verifies the stability & reliability of software applications. The goal of Stress testing is measuring software on its robustness and error handling capabilities under **extremely heavy load conditions** and ensuring that software doesn't crash under crunch situations.

#### Compatibility Testing

[Compatibility Testing](https://www.guru99.com/compatibility-testing.html#:~:text=Compatibility%20Testing%20is%20a%20type,type%20of%20Non%2Dfunctional%20testing) is a type of Software testing to **check whether your software is capable of running** on different hardware, operating systems, applications, network environments, or [Mobile](https://www.guru99.com/mobile-testing.html) devices.

#### Security Testing

[Security Testing](https://www.guru99.com/what-is-security-testing.html) is a type of Software Testing that uncovers vulnerabilities, threats, risks in a software application and prevents **malicious attacks** from intruders.

#### Recovery Testing

[Recovery Testing](https://www.guru99.com/recovery-testing.html#:~:text=Recovery%20Testing%20is%20software%20testing,after%20disaster%20or%20integrity%20loss.) is a software testing technique that verifies software's ability to recover from failures like software/hardware crashes, network failures, etc. The purpose of Recovery Testing is to determine whether **software operations can be continued after disaster or integrity loss.**

#### Usability Testing

[Usability Testing](https://www.guru99.com/usability-testing-tutorial.html), also known as User Experience(UX) Testing, is a testing method for measuring how **easy and user-friendly** a software application is.

# Days 8-10

## Automation Tools:

Some of the most widely used tools/frameworks for Automation Testing are:

* Selenium
* Cypress
* Capybara
* JMeter
* JUnit
* LoadRunner+
* RSpec

(Just get basic intro of available tools, we will dig in details later)

## API Testing:

API testing is a type of software testing that involves **testing application programming interfaces (APIs) directly and as part of integration testing** to determine if they meet expectations for functionality, reliability, performance, and security.

Since APIs lack a GUI, API testing is performed at the message layer.

#### Useful links:

1. [API Testing Tutorial: What is API Test Automation? How to Test](https://www.guru99.com/api-testing.html)
2. [What Is API Testing? | API Testing Basics](https://smartbear.com/solutions/api-testing)
3. [10 API Testing Tips for Beginners (SOAP & REST) | Complete Guide](https://www.katalon.com/resources-center/blog/api-testing-tips)

Video link: [What is an API?](https://www.youtube.com/watch?v=C_rkJRSlJS8)

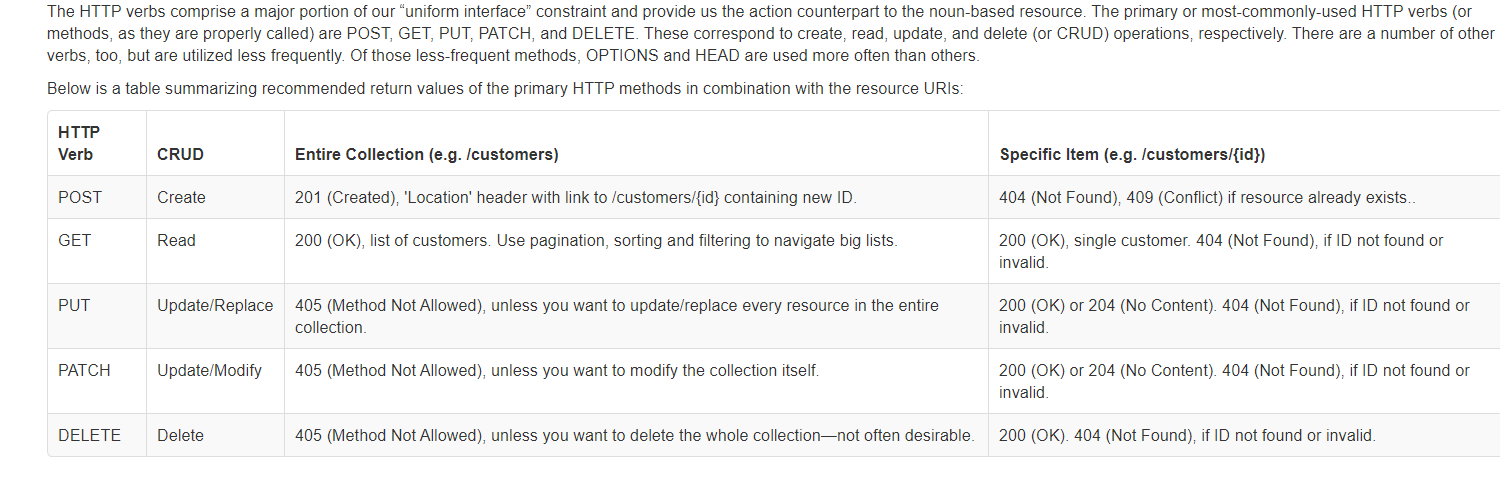
Video Link: <https://www.youtube.com/watch?v=s7wmiS2mSXY>

### REST vs SOAP APIs:

REST (Representational State Transfer) is another standard, **made in response to SOAP’s shortcomings.** It seeks to fix the problems with SOAP and provide a simpler method of accessing web services.

**Using HTTP Methods for RESTful Services**

The primary or most commonly used HTTP verbs (or methods, as they are properly called) are **POST, GET, PUT, PATCH, and DELETE.** These correspond to create, read, update, and delete (or CRUD) operations, respectively. There are a number of other methods, too, but are utilized less frequently. Of those less-frequent methods, OPTIONS and HEAD are used more often than others.



Reference: [HTTP Methods for RESTful Services](https://www.restapitutorial.com/lessons/httpmethods.html#:~:text=The%20primary%20or%20most%2Dcommonly,but%20are%20utilized%20less%20frequently.)

Reading: [HTTP request methods - HTTP](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods)

#### HTTP response status codes

HTTP response status codes indicate whether a specific [HTTP](https://developer.mozilla.org/en-US/docs/Web/HTTP) request has been successfully completed. Responses are grouped into five classes:

1. Informational responses (100–199)
2. Successful responses (200–299)
3. Redirects (300–399)
4. Client errors (400–499)
5. Server errors (500–599)

Reference:<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

### SOAP (Simple Object Access Protocol)

is a standards-based web services access protocol that has been around for a long time. Originally developed by Microsoft, SOAP isn’t as simple as the acronym would suggest.

Useful links:

1. [Understanding SOAP vs REST: Basics And Differences](https://smartbear.com/blog/test-and-monitor/soap-vs-rest-whats-the-difference)
2. [SOAP vs REST APIs: Which Is Right For You?](https://www.soapui.org/learn/api/soap-vs-rest-api)

## API Testing Tools:

The most commonly used tool for API Testing is:

### **Postman**

1. It allows users to set up all the headers and cookies the API expects and checks the response.
2. Productivity can be increased using some of the Postman features, which are listed below.

Useful links:

1. [API Testing using Postman.](https://medium.com/aubergine-solutions/api-testing-using-postman-323670c89f6d)
2. [How to Use Postman for API Testing Automation](https://www.postman.com/use-cases/api-testing-automation)
3. <https://www.youtube.com/watch?v=juldrxDrSH0&list=PLhW3qG5bs-L-oT0GenwPLcJAPD_SiFK3C&index=1>

##### Task 8:

Setup Postman on your system/web browser and take 5 sample APIs. Perform API testing with the help of Get, Post and Put methods in Postman and prepare a report on your findings.

## API Documentation:

API documentation is a **technical content deliverable, containing instructions** about how to effectively use and integrate with an API. ([HELP](https://youtu.be/rKyUA_nsCXA))

API description formats like the OpenAPI/Swagger Specification have automated the documentation process, making it easier for teams to generate and maintain them.

Useful links:

1. [What is API Documentation?](https://swagger.io/blog/api-documentation/what-is-api-documentation-and-why-it-matters)
2. [API Documentation Tool](https://www.postman.com/api-documentation-tool)
3. [API Documentation | Sample APIs Document, Docs & API Documentation Template](https://stoplight.io/api-documentation-guide/basics)
4. [API Documentation Solutions](https://medium.com/technical-writing-is-easy/api-documentation-solutions-d3719af2780f)

##### Task 9:

While performing the API testing, prepare the API documentation in parallel by using Postman’s documentation feature or Swagger Documentation.

# Days 11-12

## Load/Stress Testing Tools:

Following are some of the most commonly used Load/Stress Testing Tools:

* JMeter
* JUnit
* LoadRunner
* WebLoad
* NeoLoad
* GTMetrix

Useful links:

1. [Top 10 Performance Testing Tools | Load Testing Tools Guide](https://www.edureka.co/blog/performance-testing-tools)
2. [What is STRESS Testing in Software Testing? Tools, Types, Examples](https://www.guru99.com/stress-testing-tutorial.html)
3. [JMeter Tutorial](https://www.tutorialspoint.com/jmeter/index.htm)

##### Task 10:

Setup JMeter on your system. Perform Load and Stress Testing at [www.hotmail.com](http://www.hotmail.com) and prepare a report on your findings.

The core focus should be on the response time against a certain applied load over the sample website.

## CI/CD

### Continuous Integration (CI)

is a development practice **where developers integrate code into a shared repository frequently, preferably several times a day.** Each integration can then be verified by an automated build and automated tests. While automated testing is not strictly part of CI it is typically implied.

### Continuous Delivery (CD)

is a software engineering approach in which **teams produce software in short cycles, ensuring that the software can be reliably released at any time and,** when releasing the software, doing so manually. It aims at building, testing, and releasing software with greater speed and frequency.

Useful links:

1. [What is Continuous Delivery? - Continuous Delivery](https://continuousdelivery.com)
2. [Continuous integration vs. continuous delivery vs. continuous deployment](https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment)
3. [What is Continuous Integration: Testing, Software & Process Tutorial](https://codeship.com/continuous-integration-essentials)

### CI/CD Tools and Frameworks:

Following are the most commonly used CI/CD tools and frameworks:

* CircleCI
* Travis CI
* Gitlab CI
* Jenkins

#### Useful links:

1. [Top 20 Continuous Integration tools](https://www.guru99.com/top-20-continuous-integration-tools.html)
2. <https://travis-ci.com>
3. <https://circleci.com>

## Database Testing:

Database **Testing is a type of software testing that checks the schema, tables, triggers, etc. of the database under test.** It involves creating complex queries for performing the load or stress test on the database and checking its responsiveness. It checks the integrity and consistency of data.

Useful links:

1. [Software database testing](https://www.geeksforgeeks.org/software-testing-database-testing)
2. [Database (Data) Testing Tutorial with Sample Test Cases](https://www.guru99.com/data-testing.html)

### SQL Injection:

SQL **injection is a code injection technique that might destroy your database.** SQL injection is one of the most common web hacking techniques.

SQL injection is the placement of malicious code in SQL statements, via web page input.

Useful links:

1. [SQL Injection](https://www.w3schools.com/sql/sql_injection.asp)
2. [What is SQL Injection?](https://portswigger.net/web-security/sql-injection)
3. [What is SQL Injection (SQLi) and How to Prevent Attacks](https://www.acunetix.com/websitesecurity/sql-injection)

# Days 13-14

## Penetration Testing:

Penetration testing, also called pen testing or ethical hacking **is the practice of testing a computer system, network, or web application to find security vulnerabilities that an attacker could exploit.** Penetration testing can be automated with software applications or performed manually.

It is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system. Not to be confused with a vulnerability assessment.

Useful links:

1. [What is a pen test (penetration testing)? - Definition from WhatIs.com](https://searchsecurity.techtarget.com/definition/penetration-testing)
2. [What is Penetration Testing | Step-By-Step Process & Methods](https://www.imperva.com/learn/application-security/penetration-testing)
3. [Penetration Testing Tutorial: What is PenTest?](https://www.guru99.com/learn-penetration-testing.html)
4. [What Is Penetration Testing? What Is Pen Testing?](https://www.cloudflare.com/en-gb/learning/security/glossary/what-is-penetration-testing)

## JS Attacks/Injection:

Javascript is one of the most popular technologies and is most widely used for web pages and web applications.

It can be used for realizing different website functionalities. However, this technology can bring some **security issues**, which the developer and tester should be conscious about.

Javascript can be used not only for good purposes but for some malicious attacks too. One of them is Javascript Injection. The essence of JS Injection is to inject the Javascript code that will be run from the **client-side**.

### Why is it Important to Test JS Injection?

Many would ask if testing for JS Injection is really necessary. Checking for JS Injection vulnerabilities is a part of security testing. Security testing is usually performed only if it was included in the project planning, as it requires time, a lot of attention, and checking multiple details.

#### Useful links:

1. [JavaScript Injection Tutorial](https://www.softwaretestinghelp.com/javascript-injection-tutorial)
2. [Javascript-attacks-webviews](https://www.checkmarx.com/2017/12/07/javascript-attacks-webviews)

**Optional Task:**

It is advised that you also visit [Google Gruyere](https://google-gruyere.appspot.com/)

This web application provides a walkthrough for practical implementation which can help with the understanding and advantages of **Penetration Testing** and **JS Attacks**

## Software Development and Version Controlling:

Software Version Control (SVC), also called revision control, source control management, and Versioning Control, **is a management strategy to track and store changes to a Software Development document or set of files that follow the development project from beginning to end-of-life.**

Useful links:

1. [The Basics of Software Version Control](https://www.smartsheet.com/software-version-control)
2. [What is version control](https://www.atlassian.com/git/tutorials/what-is-version-control)?

#### Code Repositories - Github/Bitbucket/Gitlab:

Github: [Git and GitHub for Beginners - Crash Course](https://www.youtube.com/watch?v=RGOj5yH7evk)

Bitbucket: [Bitbucket Fundamental](https://www.youtube.com/watch?v=L0XVDTx-cA8)

Gitlab: [About Gitlab](https://about.gitlab.com/)

##### Task 11:

Create your Github/Gitlab or Bitbucket accounts and set up a demo project on your system, create some branches and PRs and get them approved and merged.

# Days 15-17

## Documentation Tools/Platforms

### Confluence by JIRA:

**Create, collaborate, and organize all your work in one place.** Confluence is a team workspace where knowledge and collaboration meet. Dynamic pages give your team a place to create, capture, and collaborate on any project or idea.

Useful links:

1. [How to get the most of Confluence | Atlassian](https://www.youtube.com/watch?v=sQD9HUvFjSo)

##### Task 12:

Watch Confluence’s tutorials in detail and try to create the same documents, using a couple of your already prepared docs on Google Docs(get help from buddy on which docs to replicate).

Creating/Maintaining Test Sheets:  
Before you can start making test cases, we have attached a template for you to follow. Study it and research why we use sheets and such format for maintaining test cases. You are not limited to just this template. There is always room for creativity and improvement. This is just to get you started.

Useful links:

#### [Test Case Template](https://docs.google.com/spreadsheets/d/1zA2OmzfjrA2pkiVrQYaSh2JsuMmsxSkqOsj4RNf970I/edit?usp=sharing)

##### Task 13:

Based on your current learning, create the following test cases sheets for your previous tasks:

1. Smoke Test Sheet
2. Sanity Test Sheet
3. Regression Test Sheet

#### Practice Sites:

1. [phptravels.com/demo](http://phptravels.com/demo/)
2. [thedemosite.co.uk](http://thedemosite.co.uk/)
3. [newtours.demoaut.com](http://newtours.demoaut.com/)
4. [www.way2automation.com/demo.html](http://www.way2automation.com/demo.html)
5. [automationpractice.com/index.php](http://automationpractice.com/index.php)
6. [demoqa.com](http://demoqa.com/)

##### Task 14a:

Explore the above-given sample sites and prepare Smoke, Sanity, and Regression test sheets in your Google drive accordingly for any of them. Don’t forget to PRIORITIZE the modules.

Subscribe to the live testing project on the [given link](https://www.guru99.com/live-testing-project.html). This will help you refine your testing skills. You will learn about how to test a web application according to its specific requirements.

# Days 18-19

## QA Management Tools/Plugins:

Following are the most popularly used QA/Test Management Tools:

* TestRail
* Zephyr
* TestPad
* TestMoniter
* QMetry by JIRA

Useful links:

1. [TestRail - Introduction](https://www.tutorialspoint.com/testrail/testrail_introduction.htm)
2. [Starting to Test with TestRail](https://www.youtube.com/watch?v=kBWwMUCYvMk)
3. [Defining Test Strategy](https://www.youtube.com/watch?v=bIt5jawrF8c)
4. [QMetry](https://www.qmetry.com/resources/videos/qmetry-test-management-for-jira-test-execution)

###### Task 14b:

Create a Test Suite On Test Rails containing the test scenarios, test cases, and screenshots of the relevant steps and attach screenshots if required. (In this task, the primary objective is to manually transfer the work done in 14a to the test management tool.)

## Effective Client Communication:

Effective **Client Communication is the way to win client satisfaction** that is one of the most important aspects of an SQA Engineer’s day-to-day life. Smart communication skills can make your day and win the client’s trust in your work giving your morale a real boost.

Useful links:

1. [15 Keys to Effective Client Communication](https://www.successagency.com/di/effective-client-communication)
2. [Client Communication Skills and Tips](https://www.teamgantt.com/blog/client-communication-skills)
3. [Five-strategies-client-communication](https://www.stptax.com/five-strategies-client-communication)

##### Task 15:

Prepare yourself for a client meeting considering all of the Effective Client Communication tips and tricks. To complete this task your buddy will align you with some other QA resources for a 15-20 minutes sync on the tasks that you have done till now. It will be a formal meeting where communication has to be done in English no matter what.

## Client Communication Tools:

### Slack:

Slack is a [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) [business communication platform](https://en.wikipedia.org/wiki/Business_communication) developed by American software company [Slack Technologies](https://en.wikipedia.org/wiki/Slack_Technologies). Slack offers many [IRC](https://en.wikipedia.org/wiki/Internet_Relay_Chat)-style features, including persistent chat rooms (channels) organized by topic, private groups, and direct messaging.

### Microsoft Teams:

Microsoft Teams is a proprietary business communication platform developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft), as part of the [Microsoft 365](https://en.wikipedia.org/wiki/Microsoft_365) family of products. Teams primarily compete with the similar service [Slack](https://en.wikipedia.org/wiki/Slack_(software)), offering workspace chat and videoconferencing file storage, and application integration.[[8]](https://en.wikipedia.org/wiki/Microsoft_Teams#cite_note-8) Teams are replacing other Microsoft-operated business messaging and collaboration platforms, including [Skype for Business](https://en.wikipedia.org/wiki/Skype_for_Business) and [Microsoft Classroom](https://en.wikipedia.org/wiki/Microsoft_Classroom).

[teams.microsoft.com](about:blank)

### Other tools:

Mattermost, skype, rocket chat, hip chat etc.

# Random:

[Top 8 Chrome Extensions for Website QA Testing](https://www.hostgator.com/blog/chrome-extensions-qa-testing-website/)