**SDLC:** SDLC stands for Software development life cycle, it is an approach to developing or building software with high quality and correctness so it can meet the requirements of the customer.The aim of SDLC is to build software in a predefined time frame and cost. The phases involved in SDLC are

1. **Requirement gathering** (collection) and analysis (This stage gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives that triggered the project)
2. **Feasible studies** (Once the requirement analysis phase is completed the next step is to define and document software needs. This process was conducted with the help of the 'Software Requirement Specification' document also known as the 'SRS' document. It includes everything which should be designed and developed during the project life cycle)
3. **Design** (the system and software design documents are prepared as per the requirement specification document. Which helps define the overall system architecture)
4. **Coding** (In this phase, developers start building the entire system by writing code using the chosen programming language. In the coding phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process)
5. **Testing** (During this phase, QA and testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and sends it back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system)
6. **Implement /deployment** (Once the software testing phase is over and no bugs or errors are left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final software is released and checked for deployment issues if any)
7. **Maintenance** (Once the system is deployed, and customers start using the developed system, the following 3 activities occur

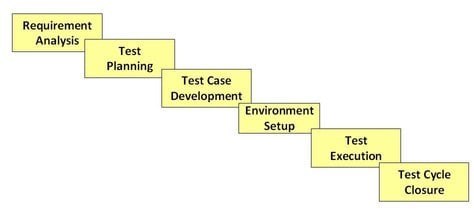
* Bug fixing - bugs are reported because of some scenarios which are not tested at all
* Upgrade - Upgrading the application to the newer versions of the Software
* Enhancement - Adding some new features to the existing software)

The main focus of this SDLC phase is to ensure that needs continue to be met and that the system continues to perform as per the specification mentioned in the first phase.

**The Advantages of SDLC**

* It gives the basic blueprint of software building .
* To enhance and improve development speed.
* Provides a framework for a standard set of activities and deliverables.
* Improved client relations.
* Helps to decrease project risk and management overhead.

**STLC:**



Understand client’s business & what does he/she wants

1. Plan it how you are going to deliver, the activities involved
2. Draft the detailed steps to-be-completed
3. Where: Identify the vendors & procure all the necessary items
4. Execute the detailed steps using items identified above
5. Deliver the quality product
6. Support your deliverable for certain duration (training, etc.)
7. Take a Retrospective look as to what can be improved the next time

***Software Testing Life cycle***

Generally, this is the ideal process for any activity. Now fitting this in Software Testing perspective, what we get is “***Software Testing Life Cycle***”, popularly known as ***STLC***. ***Software Testing Life cycle*** as in how a product or Testing process grows to maturation, from inception till the end. And at every ***Software Testing Life cycle*** phase you keep the client updated of the progress via a set of deliverables.

**Note***:* Ideally all the phases in ***Software Testing Life cycle*** have to be in the order listed with definite Entry and Exit criteria; but this is not an ideal world. Practically some of the phases can sometimes run in parallel.

1. **Requirement Analysis (What)**

Consider a situation where you sign a contract with the client to deliver some software. After signing, there are no further talks & you don’t know what does that ‘some software’ is :-( Funny? No! Your contract deadline is approaching fast. Serious Issue? Yeah!

As you might have guessed, this is the most important phase for ANY activity (specially ***Software Testing Life cycle***). Unless you know WHAT to develop or test, you cannot do much about it. So the first step to kick-off any Software Testing project is “Requirement Analysis”!

* Analyze the requirements from a testing point of view (testable requirements)
* Interact with various stakeholders (Client, Business Analyst, Technical Leads, System Architects etc.) to understand the requirements in detail
* Brainstorm & clarify ANY ambiguous requirement before the start of next phase
* Check requirements for Automation Test feasibility

***Software Testing Life cycle******Tip***: Make sure to start testing activities from the requirements phase itself because the cost of fixing defect is very less, when compared to future phases.

**Deliverables**: Requirement Traceability Matrix (RTM), Automation feasibility report, Requirements Issue tracker (updated with resolutions)

**2. Test Planning (What, How, When, Where & Who)**

Once you know what is it that client wants – Plan out your activities. This is where you define What, When, Where & Who of the future Software Testing.

* What: The different types of Testing to be conducted, Automation feasibility
* How: What will be the Test strategy & methodology and Reporting (to client) structure
* When: Drafting the schedule, activities & milestones
* Where: Identify different in-scope Test environments, Hardware and Tools requirements
* Who: Resource identification & management and defining roles & responsibilities

**Deliverables**: Test Plan / Strategy, Test schedule

**3. Test Designing (How)**

Once you have planned everything, draft the detailed steps to-be-executed. And how do we do that in terms of Software Testing?

* Derive high-level Test Conditions from the requirements
* Write, review, rework & get sign-off for Test scenarios (based on above identified Test conditions)
* Write detailed Test cases (steps) covering ‘how’ to test different aspects of the application-under-test
* Identify the Test data to be used
* Update the Requirement Traceability Matrix (RTM) — an industry-accepted format where each test case is mapped with the requirement
* Automation Test scripts (if in-scope)

**Deliverables**: Test cases, Automation scripts, Updated Requirement Traceability Matrix (RTM)

**4. Test Environment Setup (Where)**

This is a simple concept – An App will run on Mobiles or Smartphones; Applications can run on any device with different hardware configurations. The test environment decides the software and hardware conditions under which a work product is tested. In the ***Software Testing Life cycle***, practically Test environment set-up is done in parallel with the Test Design phase. The test team may not be involved in this activity, generally, a release-management team is responsible for all the environments.

* Setup Test Environment
* Smoke Test the environment
* Test data setup

**Deliverables**: Test Environment, Test Data

**5. Test Execution (Just do it)**

No explanations required I guess – Just do it. Go on to execute the Test cases to find defects. If required, do some ad-hoc tests as well. After all the bottom line of ***Software Testing Life cycle*** is – To find defects!

Bugs will be reported back to the development team for correction and retesting will be performed.

* Execute the test cases
* Log defects in case of any discrepancy (defect management system)
* Retest defects (after fix)
* Simultaneously fill the traceability metrics
* Perform Regression tests

**Deliverables**: Daily Status Reports, Defects Report

**6. Test Closure**

What do you do once all testing is completed? Summarize!

Why did client hire you? For testing of course. But Client sits at his/her office, how do you prove it to them that Testing has been completed ‘successfully’? SUCCESSFULLY. Client wants a documented report that states all signed-off Test cases have been executed, any associated risks, defects list and assurance that all the defects have been fixed, retested and closed. After all, what Client eventually wanted is quantitative & qualitative summary of the developed software, to ease the decision-making!

* Evaluate cycle completion criteria based on Test coverage, Quality, Cost, Time, Critical Business Objectives, and Software.
* Share the Test Completion Report & Test Metrics

**Deliverable**: Test Summary Report, Test Metrics

**7. post-implementation support**

What if you tested & delivered software, and on the first day itself the end-user encounters a defect? What do you do? Raise your hands & say that ‘*Boss, my work was till delivery, now you take care*? Business ethics says No, right? Exactly! Most of the projects have a warranty period post-implementation wherein customer support is part of the contract.

* Customer support during Warranty
* Test environment clean-up and restoration to default state
* Process review meetings
* Lessons learnt & Best practices are documented

After all, you learn from your experience. What can be done better from the next time? What were the mistakes? Any process improvement ideas? The idea is to remove the ***Software Testing Life cycle*** process bottlenecks for future test cycles and share best practices for any similar projects in future.

This summarizes the main chapters of our ‘***Software Testing Life cycle***’. Almost anything to everything in Software Testing comes under the purview of one of these phases. Hope this helps you to get you started!