What is the STLC

- **S**oftware **T**esting **L**ife **C**ycle
- The process of testing a software in a **well planned** and **systematic** way

Generic Phases of the STLC

1. Requirements Analysis
2. Test Planning
3. Test Analysis
4. Test Design
5. Test Construction/Verification
6. Test Execution & Bug Reporting
7. Final Testing & Implementation
8. Post Implementation

Requirements Analysis

- Testers analyze the requirements and work with BAs and Developers during the design phase to see which requirements are testable and how they are going to test those requirements
- Testers ask Business Analysts questions about the requirements to ensure the requirements are clear
- Vague or unclear requirements are one of the leading causes for bugs to slip past the testing team and into production!
- Ask questions until you understand the requirements!!!

Test Planning

- Plans are made in this phase to determine:
  - What needs to be tested
  - How the testing will be done
  - Test strategy to be followed
  - Test environment
  - Testing methodologies
  - Hardware/software requirements
  - Resources (human, technologies, etc.)
  - Risks
- IEEE 829 – Software Test Documentation Standards
  - Standard doesn't require the use of any specific document, but lays out standards for the format of various test documents
- **Test Strategy VS. Test Plan**
- Test Plan is a **Dynamic Document**

Test Analysis

- Determine what testing needs to be completed in each SDLC phase
- Automation activities can be decided in this phase
  - How will the automation be done
  - How much time will it take to automate
  - What features will be automated

Test Design

- Testers design their test cases for both black and white box testing
  - Every test case must have at least the following:
    - Test Case ID
    - Test Case Name
    - Test Case Description
    - Step Name
    - Step Description
    - Expected Result
    - RTM Information
  - If automation testing will be done, those scripts will be written in this phase

Test Construction and Verification

- Test cases are reviewed in this phase by test leads or managers and other stakeholders
- Test Case Review meetings are held to ensure that the test cases are properly testing the requirements and that appropriate data is being used
- *Before going to such a meeting with your test lead, go through your test cases and associated requirements to make sure you've written the test case correctly!*

Test Execution and Bug Reporting

- Test cases are executed AFTER unit testing is completed and the build has been released to the QA environment
- Defects are reported using a defect tracking tool (ALM, JIRA, etc.)
- Test Execution Reports are generated and distributed to all necessary parties (Development team, BAs, PMs, Test Lead/Managers)
- Once developers have fixed the bugs raised by testers, the testers do re-testing and then regression testing to ensure that the defect has been fixed AND that the fix has not affected any other area of the software

Re-Testing VS Regression Testing

Regression Testing	Re-Testing
Carried out to confirm whether a recent program or code change has not adversely (negatively) affected existing features	Carried out to confirm the test cases that failed in the final execution are passing after the defects are fixed
Purpose: New Code changes should not have any side effects to existing functionalities	Done on the basis of the defect fixes
Defect Verification is NOT part of Regression Testing	Defect Verification is part of Re-Testing
Based on the project and availability of resources, regression testing can be carried out in-parallel with Re-testing	Priority of Re-Testing is higher than Regression testing, so it is generally carried out before Regression Testing
You can do automation for regression testing, manual testing could be expensive and time consuming	You cannot automate test cases for re-testing

Re-Testing VS Regression Testing con't

Regression Testing	Re-Testing
Regression is known as generic testing	Re-testing is planned testing
Regression testing is done for passed test cases	Re-testing is done for Failed test cases
Regression testing checks for unexpected side effects	Re-testing ensures that the original fault (error) has been corrected
Done only when there is modification or changes become mandatory in existing project	Executes a test case that contains a defect with the same data and the same environment with different inputs with the new build
Test cases for regression testing can be obtained from the functional specification, user tutorials and manuals, and defect reports in regards to corrected problems	Test cases for re-testing cannot be obtained before testing has started

Final Testing & Implementation

- Final testing is conducted:
  - Stress, Performance, Load testing
- Software is verified in an environment similar to production
  - Some organizations set up their QA or UAT environments to be just like production to save time

Post Implementation

- Test environment is cleaned up and restored to a default state
- Process review meetings are conducted
- Lessons Learned Document is created
  - Retrospective meeting – led by Scrum Master

Test Strategy

- Test planning activities guide the testing team to define test coverage testing scope

- What is a Test Strategy?
  - Plan for defining the testing approach
  - Answers the questions:
    - What you want to get done
    - How you are going to accomplish it
  - This is 1 of the most important documents for any testing team

Test Plan

- **Components:**
  - Test Plan ID
  - References
  - Introduction
  - Test Items
  - Risk
  - Items to be tested (what requirements should be tested)
  - Features excluded from testing (what requirements not to test)
  - Testing Approach
  - Test Pass/Fail criteria
  - Resumption/Suspension Criteria (entry/exit criteria)
  - Test Deliverables (RTM, Test Reports, Bug Reports, etc.)
  - Test Environment Setup
  - Training and Staffing
  - Team Member Responsibilities
  - Testing Schedule
  - Planning for Risks and Contingency Plans
  - Approvals

*****It's not required that everything here be included in a test plan**

How Test Planning Takes Place in other phases of the SDLC

- Planning/Req. Analysis Phase of SDLC –
  - QA team *should* get involved while the scope of the project is gathered from the client in the form of Business Requirements. (not the case in the real world)
- Requirement Analysis Phase of SDLC –
  - SRS is created from BRD. Test plan's **INITIAL** draft is created.
  - At this point, scope of testing is not clear because QA team is not done with the SRS review
  - Test plan will only contain information on when testing is going to happen, project information, and team information (if available)

How Test Planning Takes Place other phases of the SDLC *con't*

- Design Phase of SDLC –
  - SRS review is completed and the scope of testing is identified.
  - We (QA Team) have much more information on what to test and a good estimate of how many test cases we might get, etc.
  - **2nd Version of the Test Plan** is created incorporating all this information

Aspects of a Test Plan

Items in a Test Plan	What do they Contain
Scope	Test scenarios/Test Objectives that will be validated
Out of Scope	Enhanced clarity on what we are NOT going to cover
Assumptions	All the conditions that need to hold true for us to be able to proceed successfully
Schedules	- Test Scenario prep - Test Documentation – Test cases/Test data & setting up Test environment - Test Execution - Test Cycle – How many cycles - Start/End Date for cycles

Aspects of a Test Plan con't

Items in a Test Plan	What do they Contain
Roles & Responsibilities	- Team members are listed - Who is to do what tasks - Module owners are listed and their contact info
Deliverables	- What documents (Test artifacts) are going to be produced and in what time frames - What can be expected from each document
Environment	- What kind of environment requirements exist - Who is going to be in charge - What to do in case of problems
Tools	- For example: ALM, JIRA, UFT - Login specifications – process for acquiring a login credential to use a tool - How to use a tool (where to find tutorials etc)

Aspects of a Test Plan con't

Items in a Test Plan	What do they contain
Defect Management	- Who are we going to report defects to - How are we going to report defects - What is expected – do we provide screenshots, steps to reproduce, etc.
Risks and Risk Management	- Risks are listed - Risks are analyzed – likelihood and impact is documented - Risk mitigation plans are drawn
Exit Criteria	When to stop testing

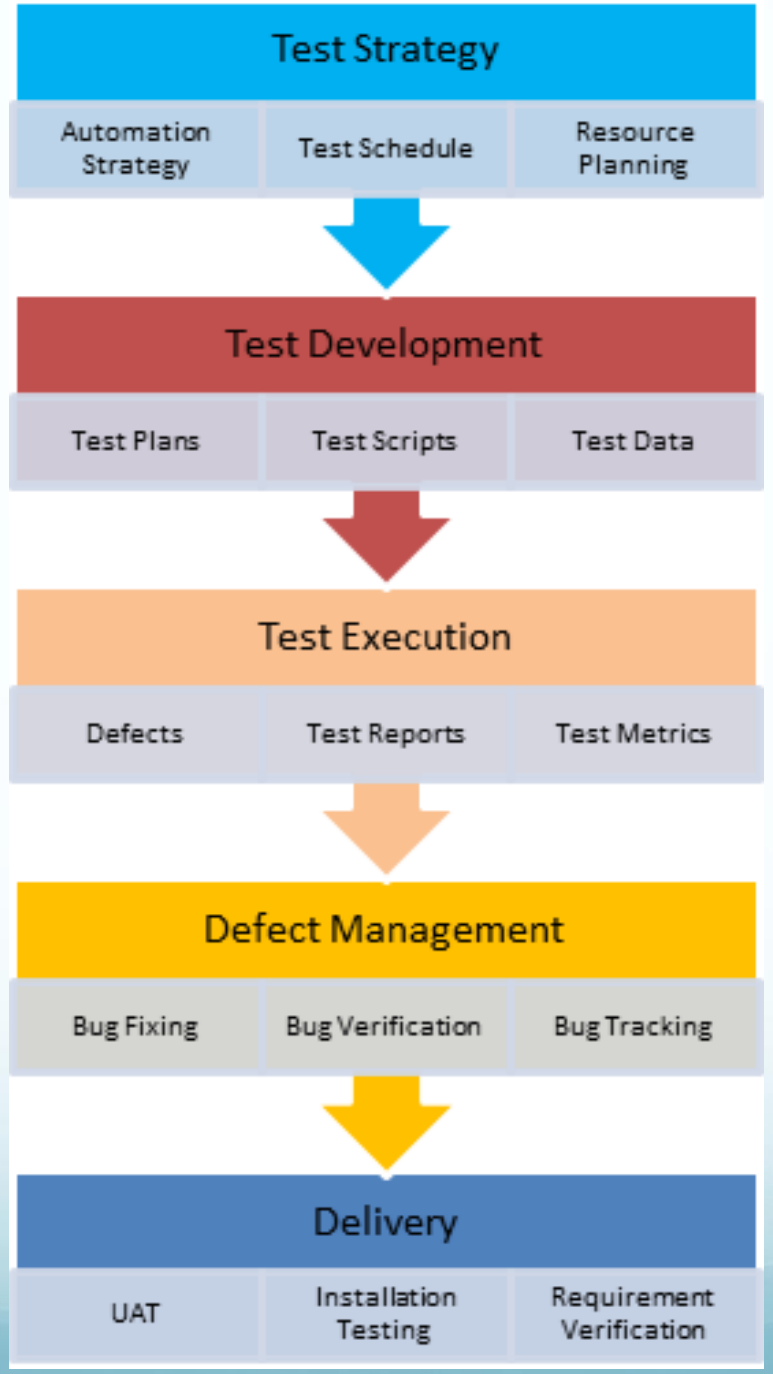
Important pointers about Test Plans

- Test Plan is a document that is the point of reference based on which testing is carried out within the QA team
- It is also a document we share with the BAs, PMs, Development team, and other teams
- It is documented by the QA manager/QA Lead based on the inputs from the QA team members
- Test planning is typically allocated 1/3rd of the time it takes for the entire QA engagement.
  - The other 1/3rd is for Test Designing and the rest is for Test Execution
- Test plan is not static; it is updated on an on-demand basis
- The more detailed and comprehensive the Test Plan, the more successful the testing activity

Test Plan VS. Test Strategy

- **Test Plan**:
  - Test focus and project scope are defined
  - Deals with:
    - Test Coverage
    - Scheduling
    - Features to be tested
    - Features not to be tested
    - Estimation
    - Resource Management

- **Test Strategy**:
  - A guideline to be followed to achieve the test objective and execution of test types mentioned in the test plan
  - Deals with:
    - Test Objective
    - Test Environment
    - Test Approach
    - Automation Tools/strategy
    - Contingency plans
    - Risk Analysis


Test Strategy
Automation Strategy
Test Schedule
Resource Planning
Test Development
Test Plans
Test Scripts
Test Data
Test Execution
Defects
Test Reports
Test Metrics
Defect Management
Bug Fixing
Bug Verification
Bug Tracking
Delivery
UAT
Installation Testing
Requirement Verification

Preparing a Test Strategy

- Steps:
  1. Scope
  2. Test Approach
  3. Test Environment
  4. Testing Tools
  5. Release Controls
  6. Risk Analysis
  7. Review and Approvals

Test Strategy - Scope

- Defines parameters like:
  - Who will review the document
  - Who will approve the document
  - Testing activities carried out with timelines

Test Strategy – Test Approach

- Defines:
  - Process of Testing
  - Testing Levels
  - Roles/Responsibilities of each team member
  - Types of Testing (Load, Security, Performance, etc.)
  - Testing Approach & automation tools if applicable
  - Adding New Defects, Re-testing, defect triage, regression testing, and test sign-off

Test Strategy - Test Environment

- Define the number of requirements and setup required for each environment
- Define backup of test data restore strategy

Test Strategy – Test Tools

- Automation and Test Management tools needed for test execution
- Figure out number of open-source as well as commercial tools required, and determine how many users are supported on it and plan accordingly

Test Strategy – Risk Analysis

- List all the risks that you can estimate
- Give a clear plan to mitigate the risks (contingency plan)

Test Strategy – Review and Approvals

- All these activities are reviewed and signed off on by the business team, PMO, Development team, etc.
- Summary of review changes should be traced at the beginning of the document along with approved date, name, and comment