Software Testing

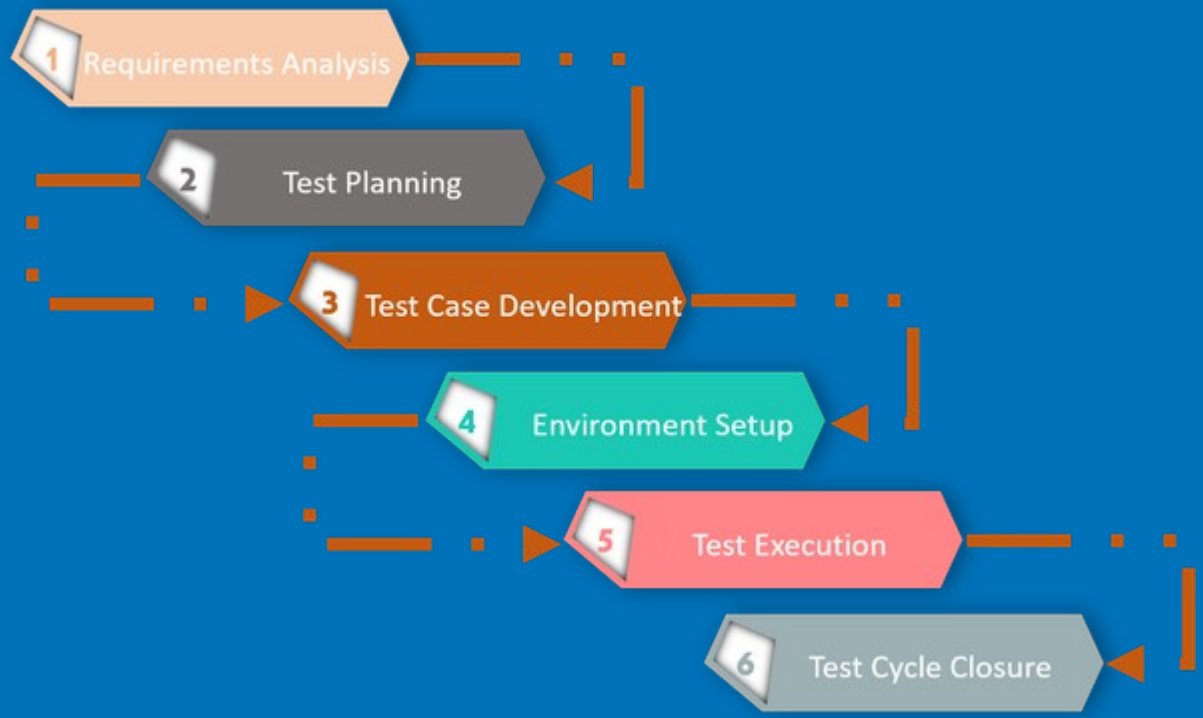
Software Testing Life Cycle

Agenda

- STLC Overview
- STLC Phases
- Practice session - Requirement analyses. Test Planning

Software Development Life Cycle - Overview

The **Software Testing Life-Cycle (STLC)** identifies and manages test activities – what and when – in order to accomplish them. Each organization has such, even though it differs between them.



STLC - Requirement analyses phase

- Research
  - **Entry criteria:** Requirements Specification, Application architectural, Acceptance criteria
  - Starts before specification approval
  - A lot of communication with different stakeholders – PO, Client, Technical lead, etc.
- Two main questions from testing perspective:
  - Who are we doing this project for?
  - What are the problems we're trying to solve?
  - What will need to be tested?
- Specification review should start as soon as possible:
  - Find potential problems/gaps
  - Share ideas/suggestions

STLC - Requirement analyses phase

When reviewing specification documents, try asking yourself these questions:

- What is the main idea behind the new functionality/project?
- What are the different user roles/workflows?
- Which is the most critical part of the functionality?
- How this idea is going to be implemented technically – what are the components, what are the communication flows/logic inside the system?
- Can you extract expected results without assumption?
- Are there unclear cases/scenarios you can think of?
- Do new requirements affect existing system and how?

STLC - Test planning phase

- Test planning is the phase in which overall testing strategy and approach are defined and communicated with stakeholders
- This strategy is often documented in a Software Test Plan (STP)

- Software Test Plan (STP)
  - Useful for the synchronization of test objectives, approach and deliverables with project stakeholders
  - Usually created by QA Manager / Lead
  - Contains high-level description of the testing approach and planned activities on the project

Software test plan - content

- Scope and Objectives
- Roles and responsibilities
- Test approach
- Entry & Exit criteria
- Suspension & Resumption criteria
- In & Out of Scope
- Communication and status reporting
- Industry standards to follow
- Test automation scope and tools
- Testing measurements and metrics
- Test deliverables
- Training needs
- Environment needs
- Risks and mitigation
- Defect reporting and tracking
- Change and configuration management
- Estimations and activity schedule
- Etc.

STLC - Test Design / Development phase

- Create high level test structure
- Develop detailed test cases
- Prioritize test cases
- Prepare test data
- Test environment setup
- Test procedures (detailed instructions on how tests should be executed)
- Automation tools / helpers preparation
- Create test suites and prepare test execution

STLC - Test Execution phase

- Execute test cases, according to approved test procedure
- Compare expected vs actual result
- Log defects
- Manage test incidents (example: changes in specification must be reflected in test cases)
- Re-testing
  - Verify defect fixes
  - Execute failed tests
- Report test results
- Iterate until meeting exit criteria
- Regression testing

STLC - Test closure phase

- Checking completed deliverables against plan
- Checking the closure of incident reports
- Report for still opened bugs
- Handover the test-ware to the maintenance organization
- Finalizing and archiving testing artifacts, the test environment and the test infrastructure for later reuse
- Retrospectives - lessons learned for future releases/projects

Test control

- Ongoing activity through all phases of the testing life-cycle
- Measure and analyze test results
- Monitor progress, test coverage and exit criteria
- Provide information and status of testing (reporting)
- Including issues and deviations from the plan
- Initiate corrective actions
- Assign extra resources or re-allocate resources
- Adjust the test schedule and test plan

Quality metrics

“If you cannot measure it, you can't manage it”

Software Testing Metrics are the bases on which decisions for the software project are taken.

Generation of testing metrics is responsibility of test lead/manager.

Example software testing metrics:

- Test coverage %
- Pass/Fail Ratio
- Defect density
- Defects found by client

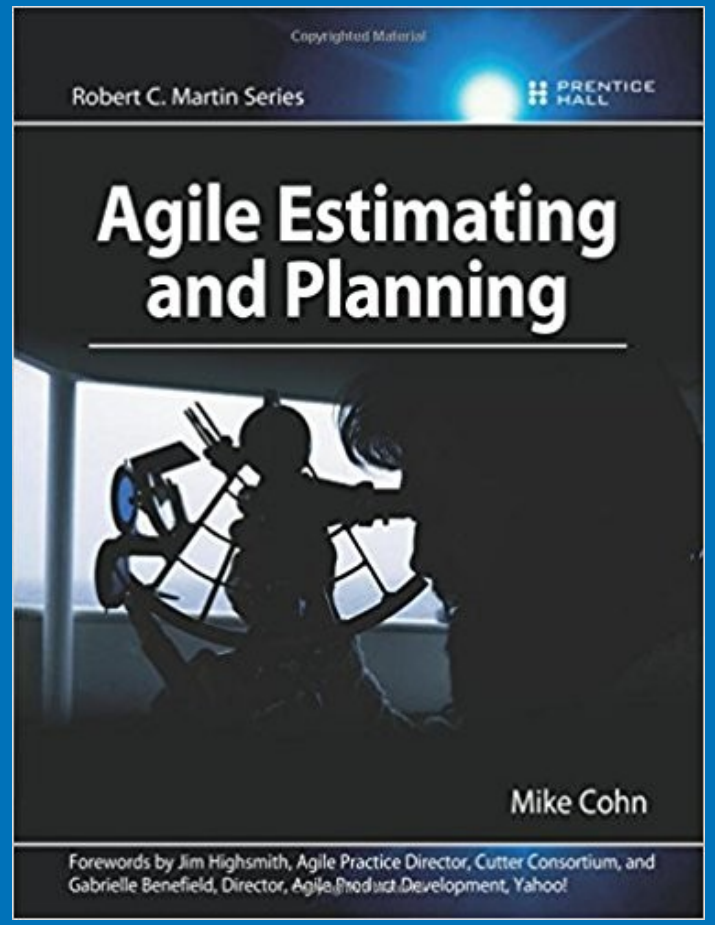
Providing effort estimations

- Project plan / delivery dates depend on the completion of required development and testing activities
- When estimating work from testing perspective, all related testing activities should be considered (for example: test environment and test data preparation, test design and execution, regression testing, etc.)
- It is often hard to estimate (i.e. predict) how complex or time consuming each task is
- Therefore, various estimation techniques exist (task breakdown, analogue, Functional points analyses (FPA), group estimations, etc.)

Planning Poker

- In agile methods, such as Scrum, teams often use a **group estimation technique** called “Planning poker”
- In Planning poker, each story is estimated with ‘points’, often called ‘Story points’
- Story points indicate complexity of the estimated user story

Further reading



SOFT ACADEMY