**Quality Assurance (QA): 🡪 ConceptWeb folder chk**

* **QA vs Software Testing**
  + **QA**: monitor the quality of process
  + **Software Testing:** ensure the functionality of the final product
* What is **testware**?
  + Test case + test data + test plan used to design and execute a test
* **Build vs Release?**
  + **Build:** development team **–**number🡪 testing team
  + **Release**: development team **–**number🡪 customer
* **Automation challenges SQA facing while testing**
  + Mastering the automation tool
  + Reusability of automation script
  + Adaptability of test case for automation
  + Automating complex test cases
* **Bug leakage vs Bug release**
  + **Bug release:** testing team knows bug presents in the current release
  + **Bug leakage**: bug discovered be the user/customer
* **Bug Cycle**
  + Tester found bug report 🡪 development manager
  + If (bug is valid) 🡪 development team fix the bug
    - Else (invalid bug) 🡪 development team ignore and reject
  + If (bug is NOT in current release) 🡪 postpone the bug-fixing
  + If (bug has been raised earlier) 🡪 tester assign **duplicate** **status**
  + If (bug is assigned) 🡪 tester assign **in-progress status**
  + if (bug is fixed) 🡪 tester assign **fixed** then **closed** status
* **Test strategy** includes**:**
  + **Intro**
  + **Resources**
  + **Scope & schedule for test activities**
  + **Test tools**
  + **Test priorities**
  + **Test planning**
  + **Types of test must be performed**
* **Software testing types:**
  + Unit testing
  + Integration testing and regression testing
  + Shakeout testing
  + Smoke testing
  + Functional testing
  + Performance testing
  + White box & black box testing
  + Alpha and Beta testing
  + **Load & stress & volume** testing
  + System testing
* **Branch & Boundary testing**
  + **Branch testing:** test all branches of the code once
  + **Boundary testing:** while testing, focusing on the **limit conditions** of the software
* **AGILE testing: start testing along with the beginning of** 
  + Do not wait for the development team finish the coding first
  + Code and testing goes simultaneously; requires continuous customer interaction
* **Test case**
  + **Specific condition** to check against program
  + Contains: **info of test steps + prerequisites + testing environment + outputs**
* **Test plans & test cases content**
  + Testing objectives
  + Testing scope
  + Testing the frame
  + The environment
  + Reason for testing
  + Criteria for entrance and exit
  + Deliverables
  + Risk factors
* **Strategy for automation test plan**
  + Preparation
  + Recording scenario
  + Error handler incorporation
  + Script enhancement by inserting check points and looping constructs
  + Debugging script and fixing issues
  + Run script again
  + Report the result
* **Quality audit**
  + The systematic and independent examination for determining the effectiveness of quality control procedures
* **Tools used by tester:**
  + Selenium
  + FireBug
  + OpenSTA
  + WinSCP
  + YSlow for FireBug
  + Web Developer toolbar for firebox
* **Stress vs. Load vs. Volume testing**
  + **Load testing:** testing application under heavy but expected load
  + **Stress testing:** when load placed on the system is raised or accelerated beyond the normal range
  + **Volume testing:** process of checking the system, whether the system can handle the required amount of data and user request
* **5 common solutions for software development problems**
  + setting up the requirements criteria, the requirements of a software should be complete, clear and agreed by all
  + realistic schedule: time for planning, designing, testing, fixing bugs and re-testing
  + adequate testing: start testing immediately after one or more modules development
  + use rapid prototype during design phrase so that it can be easy for customers to find what to expect
  + use group communication tools
* **what is USE case?**
  + Doc describes user ***action*** and system ***response*** for a specific functionality
  + **Including**: revision history, table of contents, flow of events, cover page, special requirements, pre-conditions and post-conditions
* **User stories**
  + Describes what **user (may want to) will do** when using your software
* **CRUD** testing and how to test **CRUD**?
  + Create + Read + Update + Delete; can be done using SQL statements
* **Thread testing:**
  + Top-down testing, where the progressive integration of components follows the implementation of subsets of the requirements
  + As opposed to the integration of components by successively lower levels
* **Configuration Management**
  + Process to control and record any changes made during the life of a project.
  + 3 aspects: release control + change control + revision control
* **Ad Hoc testing**
  + **A test phase** where the tester tries to **break the system** by randomly trying the system’s functionality (may include negative testing)
* **Roles of SQA engineer**
  + Writing source code
  + Control of source code
  + Reviewing code
  + Change management
  + Configuration management
  + Integration of software
  + Program testing
  + Release management process
* **What are test driver & and test stub? Why required?**
  + **Stub** is called from the software component to be tested (**top-down approach**) by **driver**
  + **Driver** calls the component to be tested (**bottom-up approach**)
  + Required when we need to test the interface between modules X and Y and we have developed only module X, thus we must test both
* **Why is bug triage (classified)?**
  + Ensure bug report completeness
  + Assign and analyze the bug
  + Assigning bug to proper bug owner
  + Adjust bug severity property
  + Set appropriate bug priority
* Tools required to **support** testing during software development:
  + **Test** Management Tools: **JIRA, Quality Center**
  + **Detect** Management Tools: **Test Director, Bugzilla**
  + **Project** Management Tools: Sharepoint
  + **Automation** Tools: RFT, QTP, WinRunner
* **“Test Metric is Software testing”**
  + the standard of test measurement
  + contains: **total test / test run / test passed / test failed / test deferred / test passed (1st time)**
  + **traceability matrix:** map scripts to requirements
* **Regression testing vs. retesting**
  + **Regression testing:** performed to check whether the bug-fix have any impact on other functionality
  + **Retesting**: recheck if the bug is fixed
* **Software Quality practices** thru the software development cycle
  + Review the requirements before starting the development phase
  + Code Review
  + Write comprehensive test cases
  + Session based testing
  + Risk based testing
  + Prioritize bug based on usage
  + Form a dedicated security and performance testing team
  + Run a regression cycle
  + Perform sanity tests on production
  + Simulate customer accounts on production
  + Include software QA Test Reports
* **Test Driven Development** (TDD) rule:
  + Prepare test cases before writing the actual code
* **Data Driven App**
  + Application flow is governed by data it processes
  + Input data set can change the behavior of the application
* **Document types in SQA:**
  + Requirement Document
  + Test Metrix
  + Test cases and Test plan
  + Task distribution flow chart
  + Transaction Mix
  + User profiles
  + Test log
  + User profiles
  + Test incident report
  + Test summary report
* **SQA testing document must include:**
  + List the number of bugs detected as per severity level
  + Explain each requirement or business function in detail
  + Designs
  + Specifications
  + Inspection reports
  + Configurations
  + Code changes
  + Business rules
  + Test plans and test cases
  + Bug reports
  + User manuals
  + Prepare separate reports for managers and users
* **What is MR? What does MR contain?**
  + MR: **Modification Request** / Defect Report
  + Reports errors/problem/suggestions in the software
* **How to conduct validation activities?**
  + Hire 3rd party independent verification and validation
  + Assign internal staff who are not involved in this
  + Independent evaluation