# Common Testing Concepts Cheat Sheet

by Natalya Rahmany (narbit) via cheatography.com/130639/cs/26070/

## Testing Types

1. **Sanity Testing** is type of testing that QA perform after receiving the software build, sanity testing is performed to ensure that the code changes introduced are working as expected .This testing is a checkpoint to determine if testing for the build can proceed or not.

2. **Smoke Testing** is is preliminary testing to reveal simple failures severe enough to, for example, reject a prospective software release.Smoke tests are a subset of test cases that cover the most important functionality of a component or system, used to aid assessment of whether main functions of the software appear to work correctly.

3. **Unit Testing** is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected.

4. **Integration Testing** is defined as a type of testing where software modules are integrated logically and tested as a group. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

5. **Functional Testing** is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

6. **Regression testing** is re-running functional and non-functional tests to ensure that previously developed and tested software still performs after a change.

7.**System Testing** is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.

8. **Load Testing** is a non-functional software testing process in which the performance of software application is tested under a specific expected load.

9. **Performance Testing** is is a testing measure that evaluates the speed, responsiveness and stability of a computer, network, software program or device under a workload.

10. **User Interface Testing** is a testing technique used to identify the presence of defects is a product/software under test by using Graphical user interface [GUI].

11. **Usability Testing** is the practice of testing how easy a design is to use with a group of representative users.

## Testing Documents

1. **STP** is the Software Test Plan document describes plans for qualification testing of Computer Software Configuration Items (CSCIs) and software systems. It describes the software test environment to be used for the testing, identifies the tests to be performed, and provides schedules for test activities.

2. **STD** is the The Software Test Description (STD) document describes the test preparations, test cases, and test procedures to be used to perform qualification testing of the Computer Software Configuration Item (CSCI) capabilities specified in the Software Requirements Specification (SRS) and Software Design Description (SDD) documents.

3. **STR** is the Software Test Report is a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software system or subsystem, or other software-related item.

4. **ATP** is the establishes the acceptance test plan used by the acceptance test team to plan, execute, and document acceptance testing. It describes the scope of the work performed and the approach taken to execute the tests created to validate that the system performs as required.

5. **RTM** is a requirements traceability matrix document that demonstrates the relationship between requirements and other artifacts. It's used to prove that requirements have been fulfilled. And it typically documents requirements, tests, test results, and issues.

## AGILE, V MODEL, WATERFALL, SPIRAL

**Agile**- the development model of software development is a conceptual framework for software engineering that promotes development iterations throughout the life-cycle of the project. The Agile approach meaning as the name refers implies something to do very quickly.

**Agile Advantages**

1.Saving time and money. 2.Focus more on the application rather than documenting things. 3.Daily meetings and discussions for the project following an Agile model can help to determine the issues well in advance and work on them accordingly. 4.Requirements changing even in the late stage of development .5.The end result is high-quality software in the least possible time duration and satisfied customers.

**Agile Disadvantages**

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

## AGILE, V MODEL, WATERFALL, SPIRAL (cont)

1.The project can easily get taken off track if the customer representative is not clear what final outcome that they want. 2.There is a lack of emphasis on necessary designing and documentation. 3. User involvement is often a potential problem especially in big and complex projects.4. Agile processes are really only applicable to products where reliability is not very critical.

**Waterfall** The original SDLC model was the Waterfall model. This model is very simple to understand and very well known in the development process. The main concept of this model is that only when one development level is completed will the next one be initiated.

**Waterfall Advantages**

1.The Waterfall method is also well known by software developers 2.The Waterfall Model works well for smaller projects where requirements are very well understood 3.Cost-effectiveness: Time spent early in the software production cycle can lead to the greater economy at later stages.

**Waterfall Disadvantages**

1.The crucial disadvantage of this model is that testing is understood as a “one-time” action at the end of the project just before the release of the operation. The test is seen as a “final inspection”, an analogy to a manufacturing inspection before handing over the product to the customer. 2. High amounts of risk and uncertainty 3.Inflexible – Poor model for complex and object-oriented projects.

**V-Model** This is the most familiar model in the development process. The model has the form of a “V”. The main idea in the V-Model is that development tasks and testing tasks are corresponding activities of equal importance, which is symbolised by the two sides of the “V”

**V-Model Advantages**

1.Due to the fact that in the V-Model defects are being repaired a short time after they have been detected, it is cheaper to fix them. 2.The model has a reputation for a very good base for the partitioning of testing. All the participants in the development of a system have a responsibility for quality assurance and testing. 3.Testing activities like requirements, test designing happen well before coding. This fact saves a lot of time and also helps in developing a very good understanding of the project at the initial stage. 4.The objectives of testing are changing, and specific for each test level.

**V-Model Disadvantages**

## AGILE, V MODEL, WATERFALL, SPIRAL (cont)

1.The V-Model is very rigid and the least flexible, which means that if one of the requirements is changing, the tester should update the test documentation as a whole. 2. This model applicable mostly to big companies because the model needs a lot of resources. 3.The amount and the intensity of the test levels should be tailored according to the specific needs of the project.

**Spiral** model is mostly used in large projects.In a spiral model, testing must be adapted to such development models, and continuous integration testing and regression testing are necessary. The tests should be a reusable test case for every component and increment, and it should be reused and updated for every additional increment.

**Spiral Advantages**

1.Spiral Life Cycle Model is a very flexible model. Development phases can be determined by the project manager, according to the complexity of the project. 2.Estimates (i.e. budget, schedule, etc.) get more realistic as work progresses because important issues are discovered earlier. 3.Good for large and mission-critical projects. 4.The software is produced early in the software lifecycle.

**Spiral Disadvantages**

1. Doesn’t work well for smaller projects. 2.Evaluating the risks involved in the project can shoot up the cost and it may be higher than the cost of building the system. 3.The spiral model is much customized for every project. 4.Risk analysis requires highly specific expertise.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com