Software Development & Testing Methodologies

# Software Development

# Agile

## Sprints

## Scrum

* An incremental approach to development using cross-functional teams
* Scrum foster flexibility and encourages team members to self-organize by daily communication
* Scrums aims at a quick a response to changes

## Scrum Artifacts

### Product Backlog Items (PBI)

* The product owner (PO) defines the PBI

### Sprint Backlog (SB)

* Committing PBIs during Sprint Planning
* Referenced during Daily Scrum

### Sprint Task

* How to achieve the Backlog Items in the current sprint
* Estimated in hours

### Sprint Burndown Chart

* Indicates remaining hours for a sprint
* Re-estimated daily

### Product/Release Burndown Chart

* Tracks the remaining Product Backlog effort from one Sprint to the next

## Scrum Roles

### Product Owner (PO)

### Scrum Dev Team

Cross-functional team members

### Scrum Master

Single person to maximize ROI

### Scrum Master

Facilitates the scrums process

## Scrum Meetings

* PO and team negotiate the PBIs to be included in the next Sprint.
* Hourly estimates or by points

### Sprint Planning

### Daily Scrum and Sprint Execution

* Stand ups

### Sprint Review/Showcase

### Sprint Retrospective

### Backlog Refinement (Grooming ?)

* Many PBIs need to be refined because they are too large and poorly understood
* This help to prepare the items for the next Sprint Meeting

# Waterfall Model

You can proceed to the next phase only when the previous phase is fully and successfully completed.

Recommended for larger projects where sophisticated documentation is important.

1. Requirements analysis
2. Design
3. Implementation
4. Verification
5. Maintenance

Testing

# Functional Testing

## Testing the application against the business requirements. Functional specs and design docs are used.

* System Testing
  + The entire system is tested for bugs
  + Interfacing hardware is also tested
  + Black box testing is included here, higher level testing is verified against user-expected working conditions
* Integration Testing
  + The individual modules already testing are now integrated with one another
  + Top-down or bottom-up approach is used
* Unit Testing
  + Testing individual components and modules
* Acceptance Testing
  + It’s the final phase of functional testing prior to the client hand-over

# Non-functional Testing

## Testing the non-functional requirements of the applications

**Security and Vulnerability Testing:**

Security testing tests the software for confidentiality, ***integrity, authentication, availability***, and non-repudiation. Individual tests are conducted to prevent any unauthorized access to the software code.

**Usability Testing:**

The usability testing part of a software testing methodology looks at the ***end-user usability aspect*** of the software. The ease with which a user can access the product forms the main testing point. Usability testing looks at five aspects of testing, – learnability, efficiency, satisfaction, memorability and errors.

**Acceptance Testing:**

The compatibility part of a software testing methodology tests that the product or application is **compatible with all the specified operating systems, hardware platforms, web browsers, mobile devices**, and other designed third-party programs (e.g. browser plugins). Compatibility tests check that the product works as expected across all the different hardware/software combinations and that all functionality is consistently supported.

# Testing Types

## Black Box

* Testing without knowledge of the internal workings of the system.
* AKA closed-box testing or functional testing.

## White Box

* Takes into account the internal logic of the code
* The tester should have knowledge of the code
* AKA open-box, glass-testing

## Gray Box

* part knowledge of the code is necessary to carry out the test
* referring to system documents and data flow diagrams