* Unit testing, typically done by the developer, but sometimes tester also helps
* Feature acceptance testing, which may be broken into couple of activities:
  + Feature verification testing, which is often automated
  + Feature validation testing, which involves everyone and mostly done manually

Example of Feature acceptance testing:

As a new depositor, in order to protect my money, I want to save my money a bank account.  Since, I am a new depositor, my default opening balance in a new bank account is INR 0.00

Automated tests are often used for running parallel regression testing to test, if any of the previously developed features might have broken. The continuous integration framework also helps checking if any builds are failing to code compilation errors. The system level test may also get started as soon as user stories are done.

There are other tests like performance, reliability, usability tests which may also run in parallel to the system tests. Acceptance testing may also include alpha and beta tests based on the type of the product being developed, either in iteration or at the end of the release. User Acceptance Testing at the end of the iteration is good idea. In regulated environments, certain regulatory testing also may be norm to satisfy the audit compliance rules.

can mark a user story as complete. There are various test levels incorporated in definition of done in Agile software development. The following list gives examples of various test levels.

* Unit testing
  + 100% unit test coverage with reviews
  + Cyclomatic code complexity and analysis of code using tools like SONAR
  + Defects in ‘acceptable’ state to the Product Owner
  + Unit tests code reviewed
  + All unit tests checked-in
  + All unit tests automated
  + Performance characteristics are within agreed limits
* Integration testing
  + Defects found reported and counted.
  + No major regression found
  + All regression tests are automated and checked in into SVN.
  + Acceptance criteria tested for both positive and negative tests based on agreed parameters
  + Quality risks identified and in acceptable
* System testing
  + All stories in a release are tested en-to-end
  + All user persons covered, if applicable
  + Testing done in staging environment or Production like environment
  + Performance testing done
  + Quality risks covered and closed, if applicable.
  + Defects in “Acceptable” state to the Product Owner.
  + Regression tests are automated and checked-in
  + Release interfaces thoroughly checked.

User Story

The definition of done for user stories may be determined by the following criteria:

* Coding tasks completed.
* Code reviews completed.
* Exploratory Testing completed and signed.
* Regression test reviewed and passed.
* Unit testing – written and passed.
* Code reviews conducted.
* Technical Design Specification updated.
* Defects are in an “acceptable” state to the Product Owner.
* User story accepted by the product owner.

Feature

The **definition of done for features**, which may span multiple user stories or epics, may include:

* All user stories for various related to their parent epics are accepted
* The technical debt is fully paid off, without major deviations
* The code is completed
* Units were written and passed for all code with full coverage
* Defects in “Acceptable” state to the Product Owner

Iteration

The **definition of done for the iteration** may include the following:

* Regression tests run and passed
* Smoke / automation tests run (if applicable)
* Demo / Review completed
* Retrospective completed
* Documentation is approved and stored.

Release

The **definition of done for a release**, which may span multiple iterations, may include the following areas:

* **Coverage:** The extent of coverage is determined by new or changed contents for the release and its complexity, size and associated risks
* **Quality:** The number of defects found per day or transaction is called defect intensity and the number of defects compared to number of user stories is called defect density. These both parameters should remain within permissible limits. The consequences of these limits which may raise residual risk may be fully understood and accepted
* **Time:** Release go/no go business decisions as per the pre-set delivery date may be evaluated.
* **Cost:** The positive return on investment which is calculated development and maintenance cost of the product may be significantly lower than the projected total sales of the product. The escaped defects after product has been released may yield lower return on investment.