# Assignment on : Software Verification and Validation

Mehedi Hasan Shifat

*2017831017*

8th Dec,2020

## Contents

## My Project Info

I have done some my pet projects and University projects. One of them is **CodeNerd** which is a Online Educational Platform where students can learn tech stuffs from articles and tutorials and teachers can make tutorials.

## 1 Q : Define the goals of software testing. Explain the goals in terms of a project you have done. How to goals impact your project ?

**Answer :**

### 1.1 Define the goals of software testing

**Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free.
The goals of software testing are classified into three major classes. They are

1. **Short-term or immediate goals :** They are the immediate results after software testing. Some of them are ...
   (a) Bug discovery
   (b) Bug prevention
2. **Long-term goals :** They effects the product quality in the long run. Some of them are ...
   (a) Reliability
   (b) Quality
   (c) Customer satisfaction
   (d) Risk management
3. **Post-implementation goals :** These goals are important after the product is released. Some of them are ...

   (a) Reduced maintenance cost
   (b) Improved testing process

## 1.2 The goals of software testing of my project

1. **Short-term or immediate goals :**
   (a) **Bug discovery :** The immediate goal of testing was to find errors at any stage of our project. More the bugs discovered at an early stage, better will be the success rate of software testing.
   (b) **Bug prevention :** It is the consequent action of our bug discovery. From the behaviour and interpretation of bugs discovered, we get to learn how to code safely such that the bugs discovered should not be repeated in later stages or future projects. Though errors cannot be prevented to zero, they can be minimized. In this sense, bug prevention is a superior goal of testing.
2. **Long-term goals :**
   (a) **Quality :** The first goal of understanding and performing the testing process is to enhance the quality of the software product.Though quality depends on various factors,such as correctness, integrity, efficiency, etc., reliability is the major factor to achieve quality. We focus on the quality of our product.
   (b) **Reliability :** The software should be passed through a rigorous reliability analysis to attain high quality standards. Reliability is a matter of confidence that the software will not fail, and this level of confidence increases with rigorous testing. The confidence in reliability, in turn, increases the quality.
   (c) **Customer satisfaction :** Testing should be complete in the sense that it must satisfy the user for all the specified requirements mentioned in the user manual, as well as for the unspecified requirements which are otherwise understood. A complete testing process achieves reliability, reliability enhances the quality, and quality in turn,in- creases the customer satisfaction.
   (d) **Risk management :** Risk is the probability that undesirable events will occur in a system. These undesirable events will prevent the organization from successfully implementing its business

initiatives. Thus, risk is basically concerned with our project.The only maintenance cost in a software product is its failure due to errors. Post-release errors are costlier to fix, as they are difficult to detect. Thus, if testing has been done rigorously and effectively, then the chances of failure are minimized and in turn, the maintenance cost is reduced.

3. **Post-implementation goals :**

   (a) **Reduced maintenance cost :** The maintenance cost of any software product is not its physical cost, as the software does not wear out.

   (b) **Improved testing process :** A testing process for one project may not be successful and there may be scope for improvement. Therefore, the bug history and post-implementation results can be analysed to find out snags in the present testing process, which can be rectified in future projects. Thus, the long-term post-implementation goal is to improve the testing process for future projects.

### 1.3 Impact of Software Testing goals on my project

1. Software testing helps to find bugs at a early state of our project.
2. It makes easy to find bugs and debug them.
3. Achieves reliability, reliability enhances the quality, and quality in turn creases the customer satisfaction.
4. The chances of failure are minimized.
5. The maintenance cost is reduced.

## 2 Q : Provide Verification and validation planning based on the project you have done.

**Answer**

## 2.1 Definition

**Verification** in Software Testing is a process of checking documents, design, code, and program in order to check if the software has been built according to the requirements or not. The main goal of verification process is to ensure quality of software application, design, architecture etc. The verification process involves activities like reviews, walk-throughs and inspection.

**Validation** in Software Testing is a dynamic mechanism of testing and validating if the software product actually meets the exact needs of the customer or not. The process helps to ensure that the software fulfills the desired use in an appropriate environment. The validation process involves activities like unit testing, integration testing, system testing and user acceptance testing.

## 2.2 Verification Planning based on my project

1. **Verification of Requirements:** We collect requirments based on our project.After gathering requirements, specific objectives are prepared considering every situation. And we made a Software Requirement Specification (SRS) from those objectives.

2. **Verification of High-Level Design :** All the requirements mentioned in the SRS document are addressed in this phase and work in the direction of designing the solution. The architecture and design is documented in another document called the software design document (SDD).here we divided into three parts :

   (a) Verification of Data Design

   (b) Verification of Architectural Design

   (c) Verification of Interface Design

3. **Verification of Low-Level Design :**

   This is the last pre-coding phase where internal details of each design entity are described. For verification, the SRS and SDD of individual modules are referred to. Some points to be considered are listed below:

   (a) Verify the SRS of each module.

   (b) Verify the SDD of each module.

   (c) In LLD, data structures, interfaces, and algorithms are represented by design notations; verify the consistency of every item with their design notations.

4. **Verification of Coding :** This is the last phase when we get the operational software with the source code.We checked that every design specification in HLD and LLD has been coded using traceability matrix. We examined the code against a language specification checklist.

### 2.3 Validation Planning based on my project

1. **Acceptance Test Plan :** This plan is prepared in the requirement phase according to the acceptance criteria prepared from the user feedback. This plan is used at the time of Acceptance Testing.

2. **System Test Plan :** This plan is prepared to verify the objectives specified in the SRS.

3. **Function Test Plan :** This plan is prepared in the HLD phase. In this plan, test cases are designed such that all the interfaces and every type of functionality can be tested.

4. **Integration Test Plan :** This plan is prepared to validate the integration of all the modules such that all their interdependencies are checked. It also validates whether the integration is in conformance to the whole system design. This plan is used at the time of Integration Testing.

5. **Unit Test Plan :** This plan is prepared in the LLD phase. It consists of a test plan of every module in the system separately.

## 3 Q : Explain the states of the bug

### 3.1 Definition

The **Bug** is the informal name of defects, which means that software or application is not working as per the requirement.A software bug can also be issue, error, fault, or failure.

## 3.2 The different states of a bug in the bug life cycle

- **New :** When a tester finds a new defect. He should provide a proper Defect document to the Development team to reproduce and fix the defect. In this state, the status of the defect posted by the tester is "New"
- **Assigned :** Defects that are in the status of New will be approved (if valid) and assigned to the development team by Test Lead/Project Lead/Project Manager. Once the defect is assigned then the status of the bug changes to "Assigned"
- **Open :** The development team starts analyzing and works on the defect fix
- **Fixed :** When a developer makes the necessary code change and verifies the change, then the status of the bug will be changed as "Fixed" and the bug is passed to the testing team.
- **Test :** If the status is "Test", it means the defect is fixed and ready to do test whether it is fixed or not.
- **Verified :** The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is "verified."
- **Closed :** After verified the fix, if the bug is no longer exits then the status of the bug will be assigned as "Closed."
- **Reopen :** If the defect remains the same after the retest, then the tester posts the defect using the defect retesting document and changes the status to "Reopen". Again the bug goes through the life cycle to be fixed.
- **Duplicate :** If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to "duplicate" by the development team.
- **Deferred :** In some cases, the Project Manager/Lead may set the bug status as deferred.

  - If the bug found during the end of the release and the bug is minor or not important to fix immediately. If the bug is not related to the current build.
  - If it is expected to get fixed in the next release.
  - The customer is thinking to change the requirement.
  - In such cases the status will be changed as "deferred" and it will be fixed in the next release.

- **Rejected :** If the system is working according to specifications and the bug is just due to some misinterpretation (such as referring to old requirements or extra features) then the Team lead or developers can mark such bugs as "Rejected"

  Some other statuses are:

- **Cannot be fixed :** Technology not supporting, Root of the product issue, Cost of fixing a bug is more

- **Not Reproducible :** Platform mismatch, improper defect document, data mismatch, build mismatch, inconsistent defects