ST Assignment No.1

Testing can be defined in simple words as "Performing Verification and Validation of the Software Product" for its correctness and accuracyof working.

1. *Software testing is as old as the hills in the history of digital computers.*
2. *The testing of software is an important means of assessing the software to determine its quality.*
3. *Since testing typically consumes 40% to50% of development efforts, and consumes more effort for systems that require higher levels of reliability, it is a significant part of the software engineering.*
4. *With the development of fourth generation languages (4GL), which speeds up the implementation process, the proportion of time devoted to testing increased.*
5. *As the amount of maintenance and upgrade of existing systems grow, significant amount of testing will also be needed to verify systems after changes are made.*

Q1. List the error case studies and explain any 3 of them in details.

Ans.

The various error case studies related to software testing are

- *Disney Lion King*
- *Intel Pentium Floating Point Division Bug*
- *NASA Mars Polar Lander*
- *Y2K Bug*

The Disney Lion King case study identifies the problem that the software product which is developed should be compatible with the various configurations and the hardware compatibility should be tested on the various machines. The product developed by the company should be given for testing to various clients for testing.

*The Intel Pentium floating point division bug occurs in the rare case when we try to develop high end mathematical project. Enter the following equation into your PC's calculator :(4195835 / 3145727) * 3145727 – 4195835. If it doesn't give the error its fine, but in few older machines it shows the floating point division error. But stakes the reputation of the company as it cannot solve the problem and monetary loss to the company.*

The NASA Mars Polar Lander failed as each module that was designed for the launch was tested separately and functioned perfectly. But the various modules thatwere designed were never integrated and tested together. The switch installed to check the landing of the mars polar lander was tested separately and whenintegrated, the bit changed to 1 by the slight jerk caused all lander to fail.

The most common the Y2K bug was the recent one which caused lot of computers to give error as the year changed from 31.12.1999 to 1.1.2000 and the software's which were developed to take the last two digits of the years turned to 00. The changes in the

software's were not possible as the developer who developed it eitherleft the company and the development cost was too high.

Q2. Define Bug and list the terms that are related to software failure.

Ans.

A bug can be defined in simple term as any error or mistake that leads tothe failure of the product or software either due to the specification problem or due to communication problem, regarding what is developed and what had to be developed.

- ***Terms for software failure***

The various terms related to software failure with respect to the area of applicationare listed as Defect, Variance, Fault, Failure, Problem, Inconsistency, Error, Feature, Incident, Bug, and Anomaly.

1. Problem, error, and bug are probably the most generic terms used.

2. Anomaly, incident, and variance don't sound quite so negative and infer more unintended operation than an all-out failure.

3. Fault, failure, and defect tend to imply a condition that's really severe, maybe even dangerous. It doesn't sound right to call an incorrectly colored icon a fault. These words also tend to imply blame: "It's his fault that the software failed.

4. As all the words sound the same they are distinguished based on the severity andthe area in which the software failure has occurred.

5. When we run a program the error that we get during execution is termed on the basis of runtime error, compile time error, computational error, and assignment error.

6. The error can be removed by debugging, if not resolved leads to a problem and ifthe problem becomes large leads to software failure.

7. A bug can be defined as the initiation of error or a problem due to which fault, failure, incident or an anomaly occurs.

8. The program to find the factorial of a number given below lists few errors problem and failure in a program.

Q3. What are different factors that lead for bugs to occur in project.

Ans.

A bug can be defined in simple term as any error or mistake that leads tothe failure of the product or software either due to the specification problem or due to communication problem, regarding what is developed and what had to be developed.

A software bug occurs when one or more of the following factors are true:

1. *The software doesn't do something that the product specification says it shoulddo.*
2. *The software does something that the product specification says it shouldn't do.*
3. *The software does something that the product specification doesn't mention.*
4. *The software doesn't do something that the product specification doesn't mention but should.*
5. *The software is difficult to understand, hard to use, slow, or—in the software tester's eyes—will be viewed by the end user as just plain not right.*

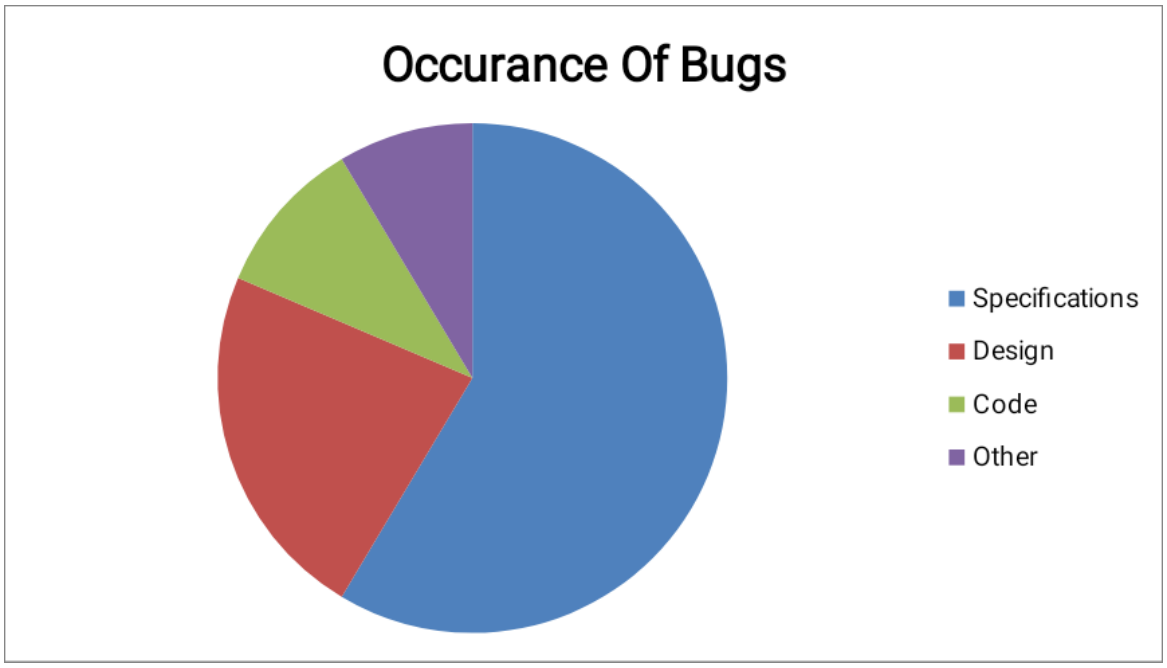
Q4.Why do bug occur in a project.

Ans.

Bugs are a mismatch between expected result and actual result. Bugs play animportant role in software testing.

In large projects bugs occur due to various reasons.

1. *One of the extreme causes is the specification.*
2. *Specifications are the largest producer of bugs.*
3. *Either specifications are not written, specifications are not thorough enough, constantly changing or not communicated well to the development team.*
4. *Another bigger reason is that software is always created by human beings.*
5. *They know numerous things but are not expert and might make mistakes.*
6. *Further, there are deadlines to deliver the project on time. So increasing pressure and workload conduct in no time to check, compromise on quality and incomplete systems. So this leads to occurrence of Bugs.*



Q5. What is cost of Bug in software testing?

Ans.

1. If the error is made and the consequent defect is detected in the requirements phase then it is relatively cheap to fix it.

2. If an error is made and the consequent defect is found in the design phase or in construction phase then the design can be corrected and reissued with relatively little expense.

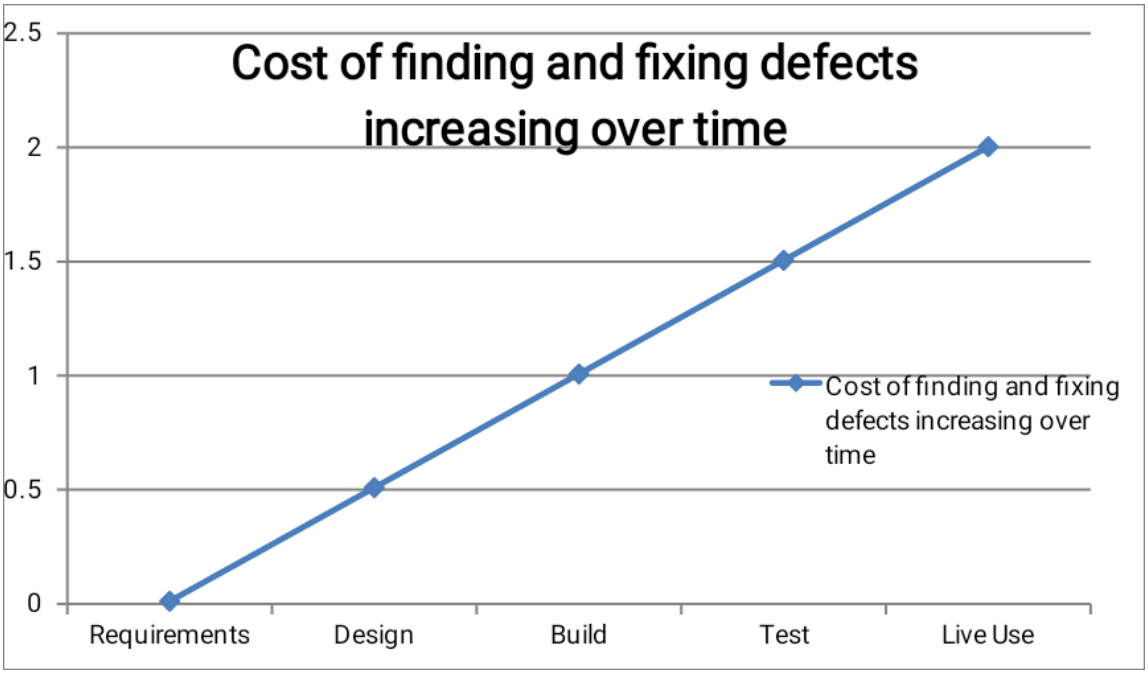


Figure 1.2

3. If a defect is introduced in the requirement specification and it is not detected until acceptance testing or even once the system has been implemented then it will be much more expensive to fix.

4. This is because rework will be needed in the specification and design before changes can be made in construction; because one defect in the requirements may well propagate into several places in the design and code.

5. All the testing work done-to that point will need to be repeated in order to reachthe confidence level in the software that we require.

6. It is quite often the case that defects detected at a very late stage, depending on how serious they are, are not corrected because the cost of doing so is too expensive.

Q6. Explain the traits/qualities of good software tester.

Ans.

List of traits that most software testers should have:

1. *They are explorers: - Software testers aren't afraid to venture into unknownsituations. They love to get a new piece of software, install it on their PC, and see what happens.*
2. *They are troubleshooters: - Software testers are good at figuring out why something doesn't work. As they find the bugs it is their responsibility to givethe solution also.*
3. *They are relentless: - Software testers keep trying. They may see a bug thatquickly vanishes or is difficult to re-create. Rather than dismiss it as a fluke, they will try every way possible to find it.*
4. *They are creative: - Testing the obvious isn't sufficient for software testers. Their job is to think up creative and even off-the-wall approaches to find bugs.Their perception of finding the bug is more than that of a programmer.*
5. *They are perfectionists: - They strive for perfection, but they know when it becomes unattainable and they want to be close as they can to the solution.*
6. *They exercise good judgment: - Software testers need to make decisionsabout what they will test, how long it will take, and if the problem they're looking at is really a bug. The job of a software tester is in time constraints.They have to find the bugs as early as possible.*
7. *They are tactful and diplomatic: - Software testers are always the bearersof bad news. They have to tell the programmers about the various bugs andthe error that exist in their program and it has to be told carefully. Good software testers know how to do so tactfully and professionally and know how to work with programmers who aren't always tactful and diplomatic.*
8. *They are persuasive: - Bugs that testers find won't always be viewed as severe enough to be fixed. Testers need to be good at making their points clear, demonstrating why the bug does indeed need to be fixed, and followingthrough on making it happen.*

Q7. Explain Waterfall and Spiral Model.

Ans.

a. **Waterfall Model:**

1. The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall)

2. The phases are Requirement Gathering and Analysis, System Design, Implementation, Testing, Deployment of System and Maintenance.

Diagram of Waterfall-model:

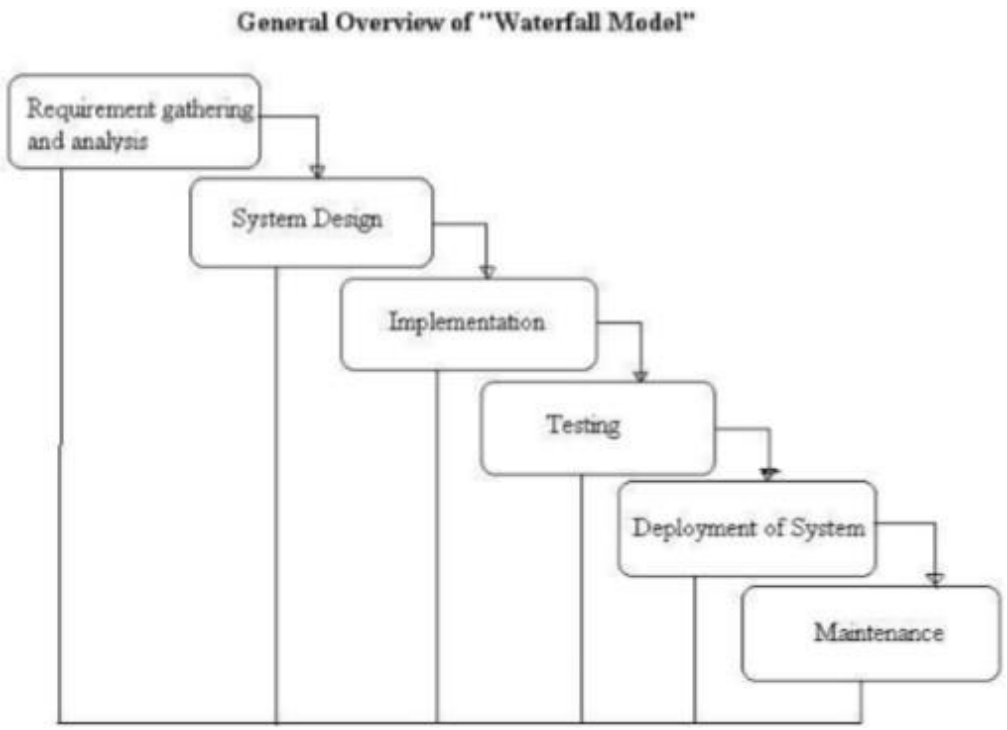


Figure 1.6

Advantages of waterfall model:

1. Simple and easy to understand and use.

2. Easy to manage due to the rigidity of the model, each phase has specific deliverables and a review process.

3. Phases are processed and completed one at a time.

4. Works well for smaller projects where requirements are very well understood.

Disadvantages of waterfall model:

1. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.

2. No working software is produced until late during the life cycle.

3. High amounts of risk and uncertainty.

4. Not a good model for complex and object-oriented projects.

5. Poor model for long and on-going projects.

6. Not suitable for the projects where requirements are at a moderate to high risk of changing.

When to use the waterfall model:

1. Requirements are very well known, clear and fixed.

2. Product definition is stable.

3. Technology is understood.

4. There are no ambiguous requirements

5. Ample resources with required expertise are available freely

6. The project is short.

b. Spiral Model:

1. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation.

2. A software project repeatedly passes through these phases in iterations.

3. The baseline spiral starting in the planning phase, requirements are gathered and risk is assessed.

4. The general idea behind the spiral model is that you don't define everythingin detail at the very beginning.

5. Start small, define your important features, try them out, get feedback from your customers, and then move on to the next level.

6. Repeat this until you have your final product.

Diagram of Spiral Model:

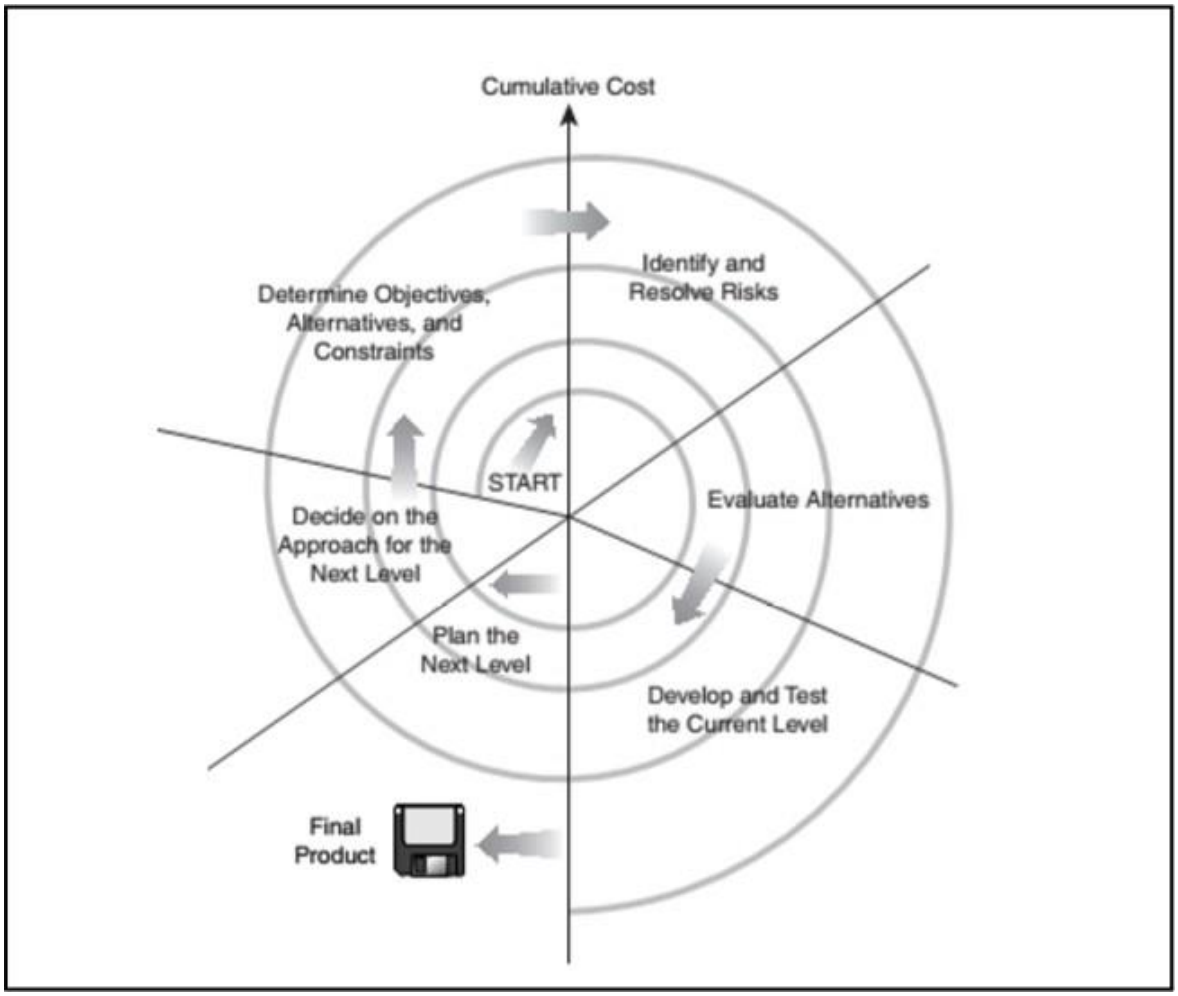


Figure 1.7

Advantages of Spiral model:

1. *High amount of risk analysis hence, avoidance of Risk is enhanced.*
2. *Good for large and mission-critical projects.*
3. *Strong approval and documentation control.*
4. *Additional Functionality can be added at a later date.*
5. *Software is produced early in the software life cycle.*

Disadvantages of Spiral model:

1. *Can be a costly model to use.*
2. *Risk analysis requires highly specific expertise.*
3. *Project's success is highly dependent on the risk analysis phase.*

4. *Doesn't work well for smaller projects.*

When to use Spiral model:

1. *When costs and risk evaluation is important*
2. *For medium to high-risk projects*
3. *Long-term project commitment unwise because of potential changes toeconomic priorities*
4. *Users are unsure of their needs*
5. *Requirements are complex*
6. *New product line*
7. *Significant changes are expected (research and exploration)*

Q8. What is Software Testing.

Ans.

Software testing *is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.*