* Terminologies related to Bug:

1 Bug
- A software bug may be defined as a coding error that causes unexpected defect, fault in your computer program.
- It introduces by the coder or developer inside the code.

2 Error
- Human action that produces an incorrect result.
- Error can be gramatical or logical error, in carrying out one or more client requirement.

3 Failure
- Failure is a inability of system or component to perform required function according ists specification.

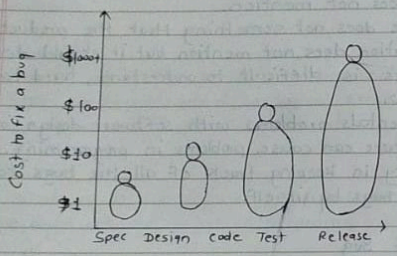
4 Fault
- It is incorrect steps, process or data defination in computer program which causes the program to perform in unintended manner (not excepted)

5 Defect
- Defect is an error in coding and logical that causes program to fail or to produces incorrect result.

* Why do bug occurs?
- Software does not do something that the product specification says it should do.
- Software does something that the product specification says it should not do.
- Software does something that the product specification does not mention.
- Software does not something that the product specification does not mention but it should do.
- Software is difficult to understand, hard to use or slow.
- Fundamentals problems with software design and architecture can cause problems in programming.
- Complexity in keeping track of all the bugs can again cause bugs by itself.

* Cost of Bug
- Cost of bug are logarithmic, they increase in size tenfold as the time increases.
- A bug found and fixed during the early stages - requirement or product specification stage can be fixed by brief interaction with concerned and might cost next to nothing.
- During coding, a swiftly spotted mistake may take only very loss effort to fix.
- During integration testing, it costs the paperwork of bug report and formally documented fix as well as the delay and expense of re-test.

- During system testing, it costs even more time and may delay delivery.
- During operation it may cause anything from a nuisance to system failure, possibly with catastrophic consequences in safety critical system such as an aircraft or an emergency service.



Time when Bug is found

x Objectives of Software Testing:
- The purpose of testing can be quality assurance, verification and validation or reliability estimation.
- To point out the error or defects that was made during development phases.
- The goal of testing is to ensure that software performs as intended and to improve software quality, reliability and maintainability.
- To found out issues occuring in the application before it is encountered by user.
- To make sure that the end result meet the business and

- Gain the confidence of the customer by providing them a quality product.
- To evaluate the overall performance of the application.
- To deliver, complete, correct and quality product to the user.

x Software Project Staff:

1 Project managers
- Producers or project managers drive the project from beginning to end.
- They are usually responsible for writing the product spec, managing the schedule, and making critical decisions and trade-offs.

2 Developer
- Programmer or coder design and write software and fix the bugs that are found.
- They work closely with the architects and project manager to create the software.
- Then they work closely with project managers and bugs testers to get the bugs fixed.

3 Tester or Quality Assurance Staff
- They are responsible for finding and reporting problems in software product.
- They work closely with all members of the team as they develop and run their tests and report the problems they find.
- SQA thoroughly covers the difference between software testing & software quality assurance task.

4 Technical writers
- It creates paper and online documentation that comes with software product.

5 Builder or configuration management
- It handles the process of pulling togethier all the software written by the programmer and all the documentation created by the writer and pulling it togetter into a single packages.

* Qualities of software tester.
1 Explorer
- Software tester are not afraid to venture into unknown situations.
- They love to get new piece of software, install it on their PC and see what happens.

2 Troubleshooter
- Software tester are good at figuring out why something doesn't work.
- They love puzzles.

3 Relentless
- Software tester keep trying. They may see bug that quickly vanishes or is difficult to re-create.
- Rather than dismiss it as a fluke, they will try every way possible to find it.

4 Poses people skills and tenacity
- Tester can face a lot of resistance from programmers.

5 Perfectionists.
- They strive for prefection, but they know when it becomes unattainable and they are okay with getting as close as they can.

6 They exercise good judgment
- Software tester need to make decision about what they will test, how long it will take and if the problem they are looking at is really a bug.

* Role of Software tester.
1 To speed up development process by identifying bugs at an early stage (Ex. specification stage)
2 To reduce the organizations risk of legal liability.
3 Maximize the value of software.
4 Assure successful launch of the project, save money, time and reputation of the company by discovering bugs and design flaws at an early stage before Failures occurs in production or in the field
5 To promote continual improvement.

Verification addresses the concern: "Are you building it right?"

Validation addresses the concern "Are you building the right thing?"

Software Testing

Verification	Validation
1. It includes checking documents, design, codes and programs.	It includes testing and validating the actual products.
2. Verification is the static testing.	Validation is the dynamic testing.
3. It does not include the execution of the code.	It includes the execution of the code.
4. Methods used in verification are reviews, walkthroughs, inspections and desk-checking.	Methods used in validation are black box testing, white box testing and non-functional testing.
5. It can find the bugs in the early stage of the development.	It can only find the bugs that could not found by the verification process.
6. The goal of verification is application and software architecture and specification.	The goal of validation is an actual product. (user requirement)
7. Quality Assurance team does verification.	Validation is executed on software code with the help of testing them.
8. It comes before validation.	It comes after verification.
9. It consists of checking of documents/files and is performed by humans.	It consists of execution of programs and is performed by computer.

SIT: It includes the activities ... in testing software. It is focus on actual testing. It is product-oriented activities, as well as preventive process. Testing is the subset of Quality control.

	Quality Assurance	Quality control.
1	It focuses on providing assurance that the quality requested will be achieved.	1) It focuses on fulfilling the quality requested.
2	It is the technique of managing quality.	2) It is the technique to verify quality.
3	It is involved during the development phase.	3) It is not included during the development phase.
4	It does not include the execution of the program.	4) It always includes the execution of the program.
5	It is a managerial tool.	5) It is corrective tool.
6	It is process oriented	6) It is product oriented.
7	The aim of quality assurance is to prevent defects.	7) The aim of quality control is to identify and improve the defects.
8	It is preventive technique.	8) It is a corrective technique.
9	It is proactive measure	9) It is reactive measure.
10	It is responsible for entire software development life cycles.	10) It is responsible for software testing life cycles.
11	It pays main focus is on the intermediate process.	11) It's primary focus is on final products.
12	All team members of the project are involved.	12) Generally, the testing team of the project are involved.
13	It aims to prevent defects in the system.	13) It aims to identify defects or bugs in the system.
14	It is less-time consuming activity	14) It is more-time consuming activity
15	Statistical Process control (SPC) is applied on QA. (Statistical technique)	15) Statistical quality control (SQC) is applied on QC.

	Static testing	dynamic testing
1	It is performed in the early stage of the software development.	1) It is performed at the later stage of the software development.
2	In static testing, whole code is not executed.	2) In dynamic testing whole code is executed.
3	It prevents defects.	3) It finds and fixes the defects.
4	Static testing is performed before code deployment.	4) Dynamic testing is performed after code deployment.
5	It is less costly	5) It is highly costly.
6	Static testing involves check-list for testing process.	6) Dynamic testing involves test cases for testing process.
7	It includes walkthroughs, code review, inspection, etc.	7) It involves functional and non-functional testing.
8	It generally takes shorter time.	8) It usually takes longer time as it involves running several test cases.
9	It can discover variety of bugs.	9) It expose the bugs that are explorable through execution hence discover only limited type of bugs.
10	It may complete 100% statement coverage in comparably less time.	While dynamic testing only achieves less than 50% statement coverage.
Ex:	Verification	Validation.
	This testing does the verification process	This testing does validation process.

software isn't actually used.

software must actually be compile and run.

ST Unit-2

* Software testing axioms
- It is impossible to test program completely.
- Software testing is a risk-base exercise.
- Testing cannot show the absense of bugs.
- The more bug you find, the more bugs there are.
- Not all bugs found will be fixed.
- It is diffult to say when bug is indeed a bug.
- Software testers are not the most popular members of a project.
- Software testing is disciplined and technical profession.

* Black box testing
- It is a software testing technique that examines the functionality of software without knowing its internal structure or coding.
- Black-box testing is also known as functional testing, data-driven testing and closed box testing.
- The main objective of black-box testing is to specify the business needs or the customer's requirement.
- In black box testing, there is less programming knowledge is required.
- As well as, it is not well suitable for algorithm testing.
- It is done at higher levels, that are system testing and acceptance testing.
- Generally, it is mainly performed by software tester.
- The base of this testing is external expectation as well as it is less exhaustive than white box testing.
- In black box testing, there is no implementation knowledge is required.
- It does not find error related to the code.

It is

- Types: Functional testing
  Non-functional testing
  Regression testing

- Advantages:
1. It is efficient when used on larger systems.
2. Tester can be non-technical
3. Test cases can be designed as soon as the functional specificification are complete.
4. There is no need of having detailed functional knowledge of system to the tester.

- Disadvantages:
1. Test cases are tough and challenging to design without having clear functional specification.
2. Tester has limited knoledge about an application.
3. Writing test cases is slow and difficult.
4. To identify all possible input in limited testing time, it is impossible.

- It is less time consuming in black box testing, time consumption depends upon the availability of the functional specification.

* White box testing
- It is a software testing, that the internal structure of the software is known to the tester.
- White box testing is also known as Structural testing, code-based testing, code-based testing and clear box testing and transparent testing.
- The main objective of white-box testing is to check the code quality.
- In white box testing, there is requirement of programming knowledge.
- As well as, it is well suitable for algorithm testing.
- Generally, it is done at lower levels, that are unit testing and integration testing.
- It is mainly performed by developer or coder.
- The base of White-box testing is coding which is responsible for internal working as well as it
- is more exhaustive than black-box testing.
- In white-box testing, there is a requirement of implementation knowledge.
- In white box testing, there is the detection of hidden errors. It also helps to optimize the code.
- Types: Path testing
  loop testing
  Condition testing

- Advantages
1. It helps in optimizing the code.
2. Extra lines of code can be removed, which can brings in hidden effect.

- Disadvantages
1. The cost is high to need of skilled tester to perform white box testing.
2. It is difficult to maintain
3. It is very impossible to look into every bit of code to find out hidden errors.

- It is more-time consuming. It takes a long time to design test cases due to lengthy of code.

* Give eight low level specification technique:
→ Following are the eight important specification attributes
1 Complete:
- Is anything missing or forgotten?
- Is it through?
- Does it include everything necessary to make it standalone?

2 Accurate:
- Is the proposed solution correct?
- Does it properly define the goal?
- Are there any errors?

3 Precise:
- Unambiguous and clear.
- Is the description exact and not vague?
- Is there a single interpretation?
- Is it easy to read and understand?

4 Consistent:
- Is the description of the feature written so that it doesn't conflict with it-self or other items in the specification?

5 Relevant:
- Is the statement necessary to specify the feature?
- Is the feature traceable to an original customer need?
- Is it extra information that should be left out?

6 Code-free:
- Does the specification stick with defining the product and not underlying software design, architecture and code?

7 Testable:
- Can the feature be tested?
- Is enough information provided that a tester could create tests to verify its operation?

8 Feasible
- Can the feature implemented with the available personal tools and resources within the specified budget & schedule?