



**V2V EDTECH LLP**  
DIPLOMA | DEGREE | BSCIT/CS

# Chapter 4 : Defect Management





**V2V EDTECH LLP**

DIPLOMA | DEGREE | BSCIT/CS



[Free Study Material](#) [Buy Ty Diploma](#) [Buy Sy Diploma](#) [Whatsapp Group for Study Material](#)

Sr. No	Question	Marks
1	Give defect classification in detail	S-15, 2 marks
2	Explain defect classification	S-15, 4 marks
3	Give the defect classification and its meaning	S-16, W-17, 4 marks
4	List all defect classification. Also describe any one defect in brief	S-17, 4 marks

5	Explain defect classification with an example	S-18, 4 marks
---	---	---------------

- **Defect Classification**

- Defect in a software project reflects its inability or inefficiency to meet the specified requirements
- purpose of testing is to trace out defects so a tester needs to be aware about different forms of defects
- Types of Defects: defects are mainly classified as:
  - Severity Basis
  - Probability Basis

– Priority Basis

1. Severity Basis:



## Types of severity

Severity	Description
<b>Critical</b>	This defect indicates complete shut-down of the process, nothing can proceed further.
<b>Major</b>	It is a highly severe defect and collapses the system. However, certain parts of the system remain functional.
<b>Medium</b>	It causes some undesirable behavior, but the system is still functional.
<b>Low</b>	It won't cause any major break-down of the system.

## Types of Defects in Software Testing



Functional Defects

Usability Defects

Performance Defects

Compatibility Defects

Security Defects

**P1**

• Critical

**S1**

• Critical

**P2**

• High

**S2**

• Major

**P3**

• Medium

**S3**

• Moderate

**P4**

• Low

**S4**

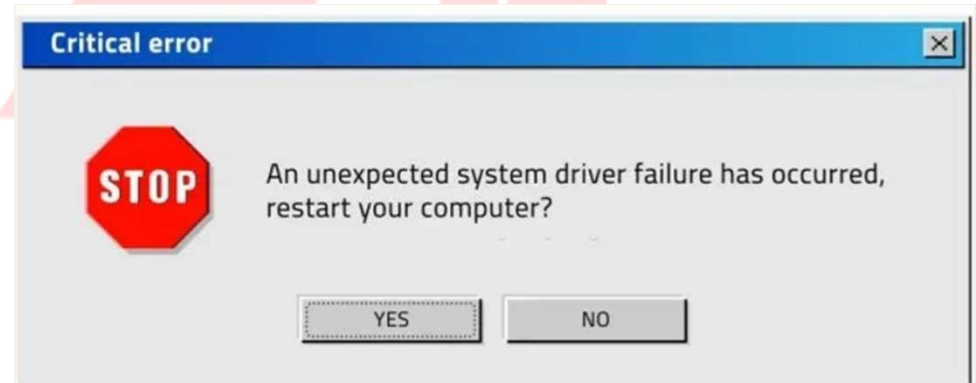
• Low

Defect Priority Levels

Defect Severity Levels



- **Severity:** it is the degree of impact.
- **Defect severity:** - degree of particular defect to impact on a software product or its working adversely.
- **Types of defects based on severity metric:**
- **Critical:**
  - Needs immediate attention and treatment.
  - Affects critical and essential functionalities.
  - Affects s/w or product on large scale (failure or system crash).







- Major:
  - Affect core or major functionalities.
  - Does not result into complete failure bring several major functions to rest.
- Minor:
- Produces minor impact and does not have significant influence.
- Results may be seen in product's working but it may not stop users to execute the task, which may be carried out using some other alternative.



–Trivial:

- They have no impact on working of product.
- Can be ignored or skipped such as spelling or grammatical mistake.
- **Illustrating with an Example:**
  - A **spelling mistake on a website's homepage** serves as a perfect illustration. From a functionality standpoint, the software operates flawlessly. Hence, the severity of this defect is low.



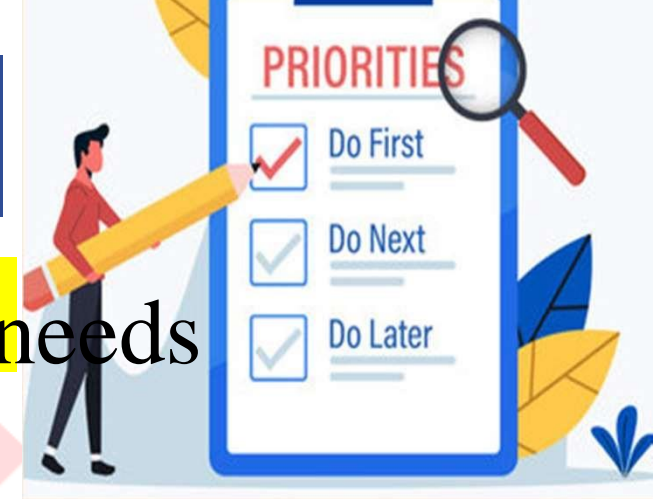
- However, the **homepage represents the brand's image**. A spelling mistake there can **tarnish the brand's reputation**. the severity of this defect is major.
- The same is true when you create a **logo**. Due to its **high visibility and potential impact on brand perception**, the priority to fix this defect becomes high.



## 2. Probability Basis:



- **Probability = Likelihood/chances**
    - High: high probability of getting traced out by almost all users of the application.
    - Medium: half of the users are able to trace out the defect.
    - Low: detected by few users or not detected by any user.
3. Priority Basis:
- Defects are seen from business perspective.



- Priority is decided based on current needs and demand of the business.

- High: defects need to be corrected as soon as possible in the same build.
- Medium: defects next to high priority – may be addressed in next version.
- Low: does not need to be corrected individually /may not be corrected with other defects.



## Defect

Wrong

Missing

Extra

- Other defects:

- Extra

- Defects:

- defects due

- to implementation of requirements other than specified ones.

- Missing Defects: arise due to not fulfilling any requirement specified by user.

–Wrong Defects: defects due to misunderstanding between specified requirements and what we understood. (misunderstood or misinterpreted)

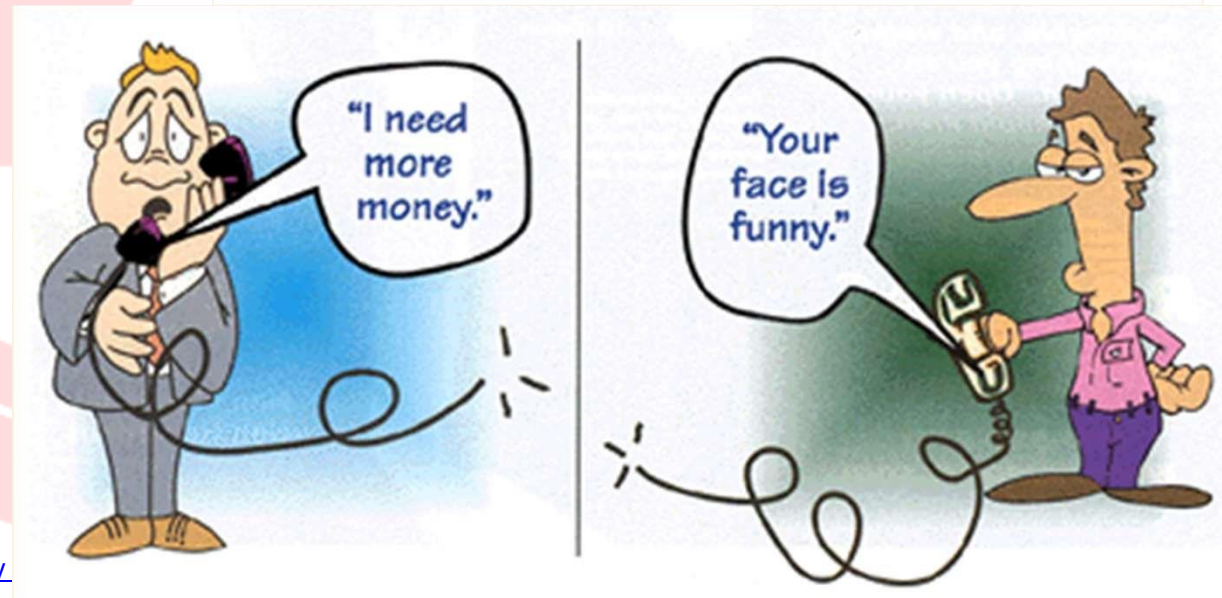
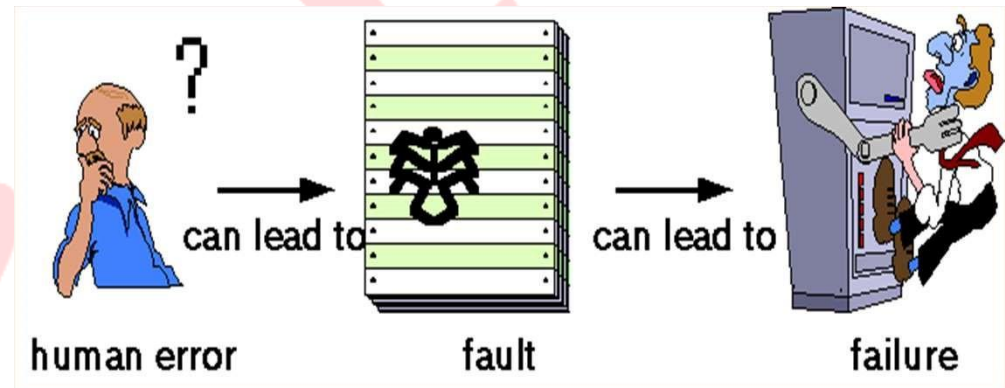
Sr. No	Question	Marks
1	Which are different causes of software defects	W-15, S-18, 4 marks



## • Causes Of Defects

1. Human Factor: It is a human tendency to make mistakes.

2. Communication Failure: miscommunication, lack of communication, incorrect communication arise when requirements are







incomplete or code is modified for the second time.

3. Unreal Development Timeframe: when tester **does not have enough information about schedule** which could lead to bad quality or defective service.



4. Poor Coding Practices: lack of validation and missing errors lead to defects in code which becomes difficult to debug.



5. Buggy third-party tools: third party tools may contain defects which can cause bugs in current



software.

6. Lack of Skilled Testing: it is important to have seriousness for testing and enough skills for it.





Sr. No	Question	Marks
1	Describe steps in defect management process	S-15, 4 marks
2	What is defect management	S-16, 2 marks



3	Explain defect management process with proper diagram	W-15, S-16, W17, S-18, W-18 4 marks
---	---	---

- **Defect Management Process:**
- **Defect Prevention:** it is efficient and effective in reducing the number of defects.



- It is also cost effective to fix defects found during **early stages**.
- Most organizations conduct Defect Discovery, Defect Removal and then Process Improvement which is collectively known as Defect Management Process.
- Goals of defect management
  - Prevent The Defect
  - Early Detection
  - Minimize The Impact



- Resolution of The Defect
- Process Improvement

Fig: Defect Management Process





Sr. No	Question	Marks
1	Illustrate defect prevention process of defect fixing process	S-15, 4 marks
2	Explain defect prevention cycle	W-18, 4 marks

## 1. Defect Prevention

- Best mechanism to remove defects in **early stages**
- This is **cost effective**.





- But it is **difficult to eliminate all defects at early stages** but their impact can be minimized.
- Steps involved in defect prevention:
  - Identify Critical Risk: risks having big impact on the system.
  - Estimate Expected Impact: calculate amount of financial impact.
  - Minimize Expected Impact





: try to minimize or eliminate the risks.

## 2. Deliverable Baseline

When deliverables (system, product or document) reaches its predefined milestones then we can say deliverable is a baseline.



- If deliverable moves from one stage to another then **defect** in the system also carried forward to next milestone or stage.
- As issues are identified after completion of earlier milestones they are called as **defects**.
- Finally deliverables are baselined when changes into deliverables are done and possible defects are identified and fixed.

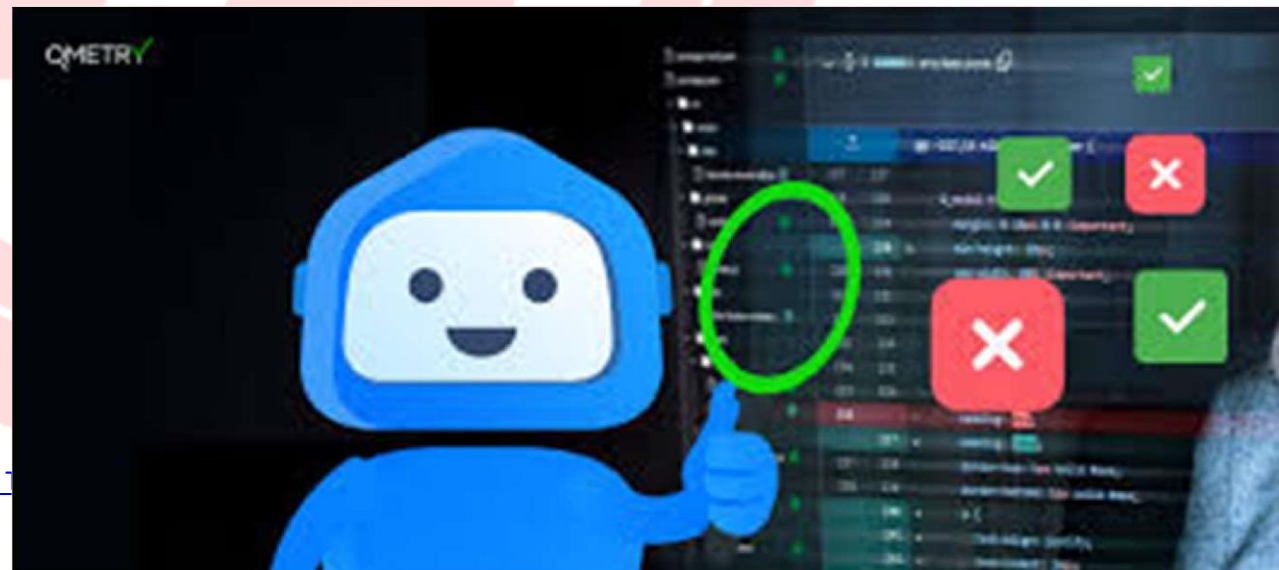


- Defects which are found before achieving the milestones are not called as defects so despite of having defects, code is called as baselined and ready for next milestone .

### 3. Defect Discovery



- It is almost impossible to remove all defects but defects can be identifies at earlier stages before they become costlier.
- Defects are discovered means defects are accepted by development team.





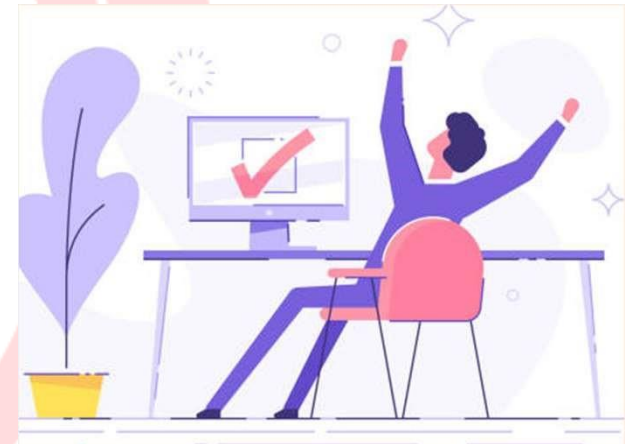
- Steps involved in Defect Discovery:
  - Find a defect: Identify defect before it becomes major problem to the system
  - Report Defect: Testing team makes development team aware about the issue
  - Acknowledge Defect: Development team



acknowledges the issue and fixes it if it is a valid defect

## 4. Defect Resolution

- Development team has to proceed for resolution of the defect.
- Steps involved in defect resolution:
  - Prioritize the risk: Defects having **more impact** on the system are fixed on **highest priority**.





□ Fix the defect: higher priority defects are **fixed first** and lower priority defects are **fixed at the end**.

□ Report the resolution: it is the development team's responsibility to ensure that testing team is **aware** when the defects are going for a fix and how the defect has been fixed.



## 5. Process Improvement

- Though the defects are resolved on the basis of priority it does not mean that lower priority defects





are not important and not much impacting the system.

- From process point of view all defects are identified are same as a critical defect.
- For process improvement everyone in the project needs to look back and check from where the defect was originated.



- Based on that, **changes** are made in validation process, baselining document, review process.
- Which will help to catch defect early and will be less expensive.

Sr. No	Question	Marks
-----------	----------	-------



1	Draw the diagram of defect/bug life cycle and explain its process.	S-15, 4 marks
2	Explain defect life cycle to identity status of defect with proper labelled diagram	W-15,W-17, 4 marks
3	Explain the defect tracing with defect/bug life cycle diagram and the different defect states	S-16, 8 marks



4	Describe defect life cycle with neat diagram	S-17,S-18, 4 marks
---	--	--------------------

- **Defect Life Cycle:**

- Defect life cycle, also known as **Bug Life cycle**.
- It is the journey of a defect cycle, which a defect goes through during its lifetime.

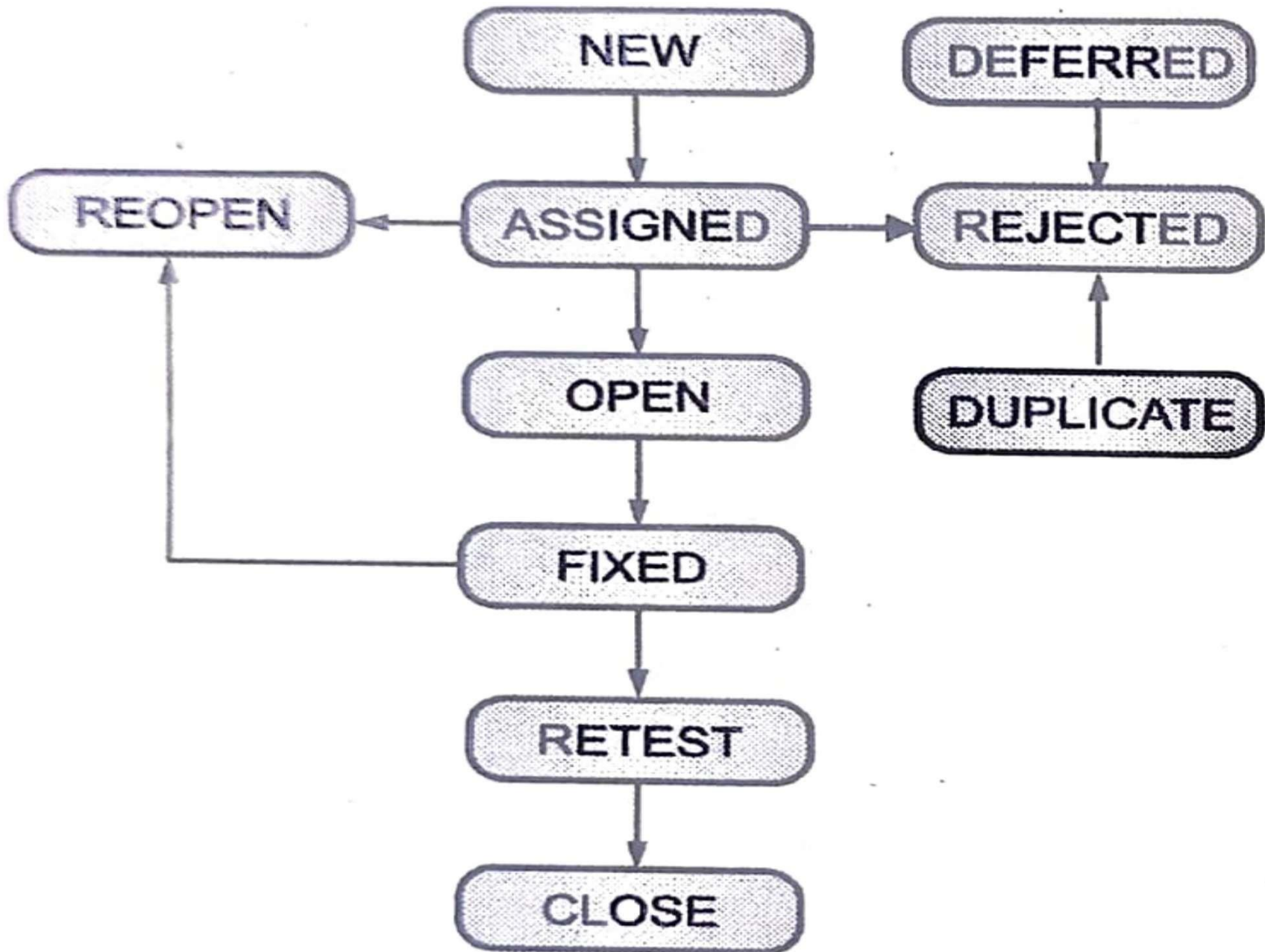


- It varies from organization to organization and project to project.
- governed by the software testing process.
- depends upon the tools used.



**V2V EDTECH LLP**

DIPLOMA | DEGREE | BSCIT/CS





- Defect Life Cycle includes following stages :
  - a. **New** : When a defect is logged and posted for the first time.
  - b. **Assigned** : Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to developer team. Defect can be assigned to developer who owns the functionality or can be assigned to



development lead who will further move the defect to developer.

- c. **Open**: It is the state when developer starts analysing and working on the defect fix.
- d. **Fixed** : When developer makes necessary code changes and verifies the changes then he/she can make bug status as “Fixed”.





- e. **Retest** : At this stage the tester do the **retesting of the changed code** which developer has given to him to check whether the defect got fixed or not.
- f. **Reopened** : If the bug still exists even after the bug is fixed by the developer, the **tester changes the status to "reopened"**. The bug goes through the life cycle once again.



- g. Deferred** : The bug, changed to deferred state means the bug is expected to be fixed in next releases. the bug may not have major effect on the software.
- h. Rejected** : If the developer feels that the bug is not genuine, developer rejects the bug. Then the state of the bug is changed to “rejected”.



- i. **Duplicate** : If the bug is repeated twice or the two bugs mention the same concept of the bug, then the recent/latest bug status is changed to "duplicate".
- j. **Closed** : Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, tester changes the status of the bug to "closed". This state means that the bug is fixed, tested and approved.



Sr. No	Question	Marks
1	State how to minimize risk impact while estimating defects	S-15, 4 marks
2	What are the points considered while estimating impact of a defect.	W-15,W-17, 4 marks



3	What do you mean by ‘Defect Impact’? Explain how to estimate defect impact	S-16, 4 marks
---	--	---------------

- **Estimate Expected Impact of a Defect**
- Defect impact means problems which may arise when an application with defect is executed.



- Once the critical risks are identified, the financial impact of each risk should be estimated.

The expected impact of a risk (E) is calculated as :

$$E = P * I$$

Where

P = Probability of the risk becoming a problem and

I = Impact in dollars if the risk becomes a problem.





- The risks should be prioritized by the expected impact and the degree to which the expected impact can be reduced.
- important is to identify the risk, and determine the risk's order of magnitude.
- Large, complex systems will have many critical risks.



- Reduction of risk leads to successful project.
- One of the more effective methods for estimating the expected impact of a risk is the **annual loss expectation (ALE) formula**:
- The occurrence of a risk can be called an event."
- Loss per event can be defined as the average loss for a sample of events.
- The formula states that the **ALE** equals the loss per event multiplied by the number of events.





- For example, if the risk is that the software system will abnormally terminate, then the average cost of correcting an abnormal termination is calculated and multiplied by the expected number of abnormal terminations associated with this risk
- For the annual calculation, the number of events should be the number of events per year.

Sr. No	Question	Marks
-----------	----------	-------



1	Explain techniques to find defect in short	W-15,W-17, 2 marks
2	List the different techniques to find defects. Describe any two of them	W-16, 4 marks
3	Describe techniques for finding defects.	S-17, 4 marks
4	List and explain techniques for finding bugs.	W-17, 4 marks



5

What are different techniques for finding defects? Explain in detail

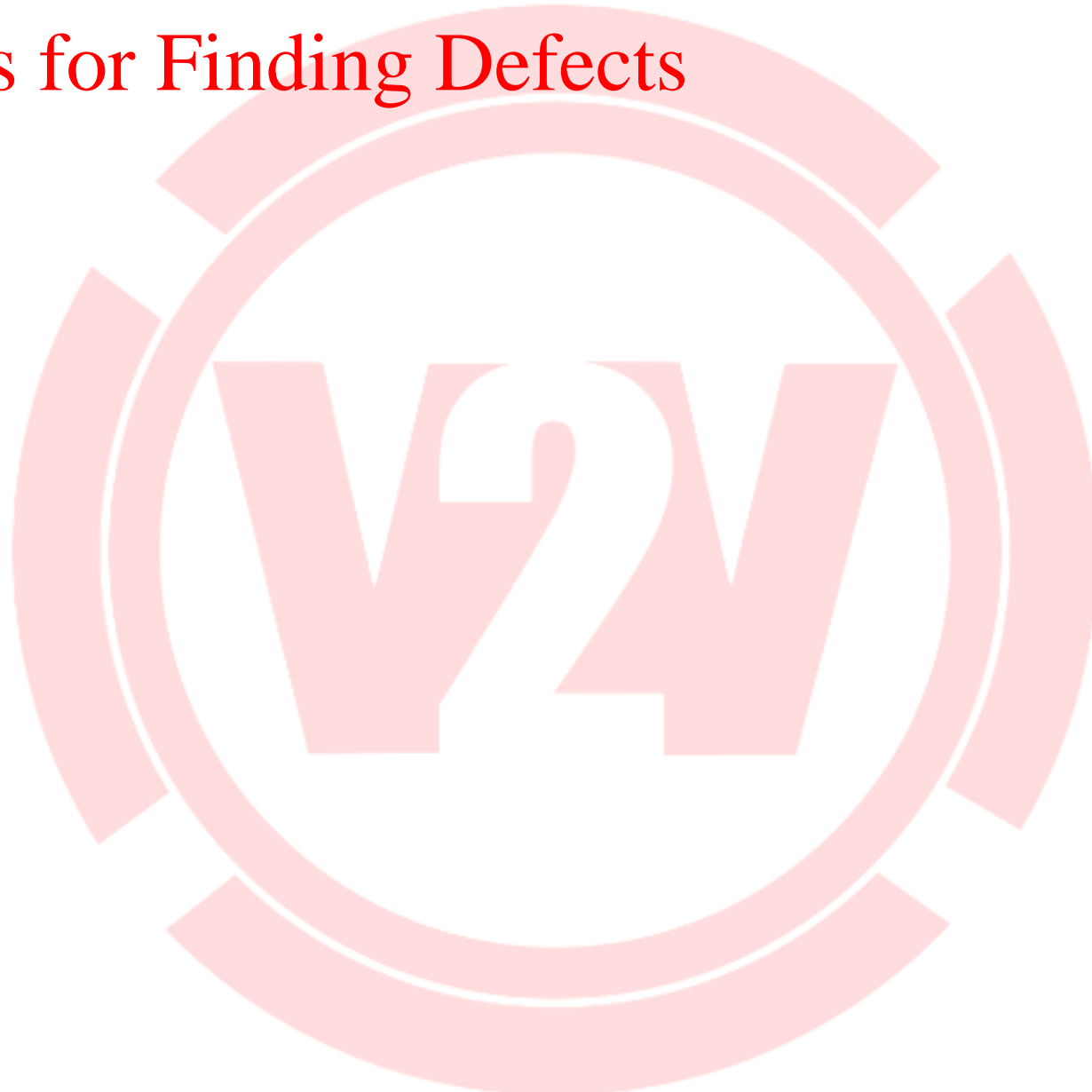
S-18, W-18, 4 marks





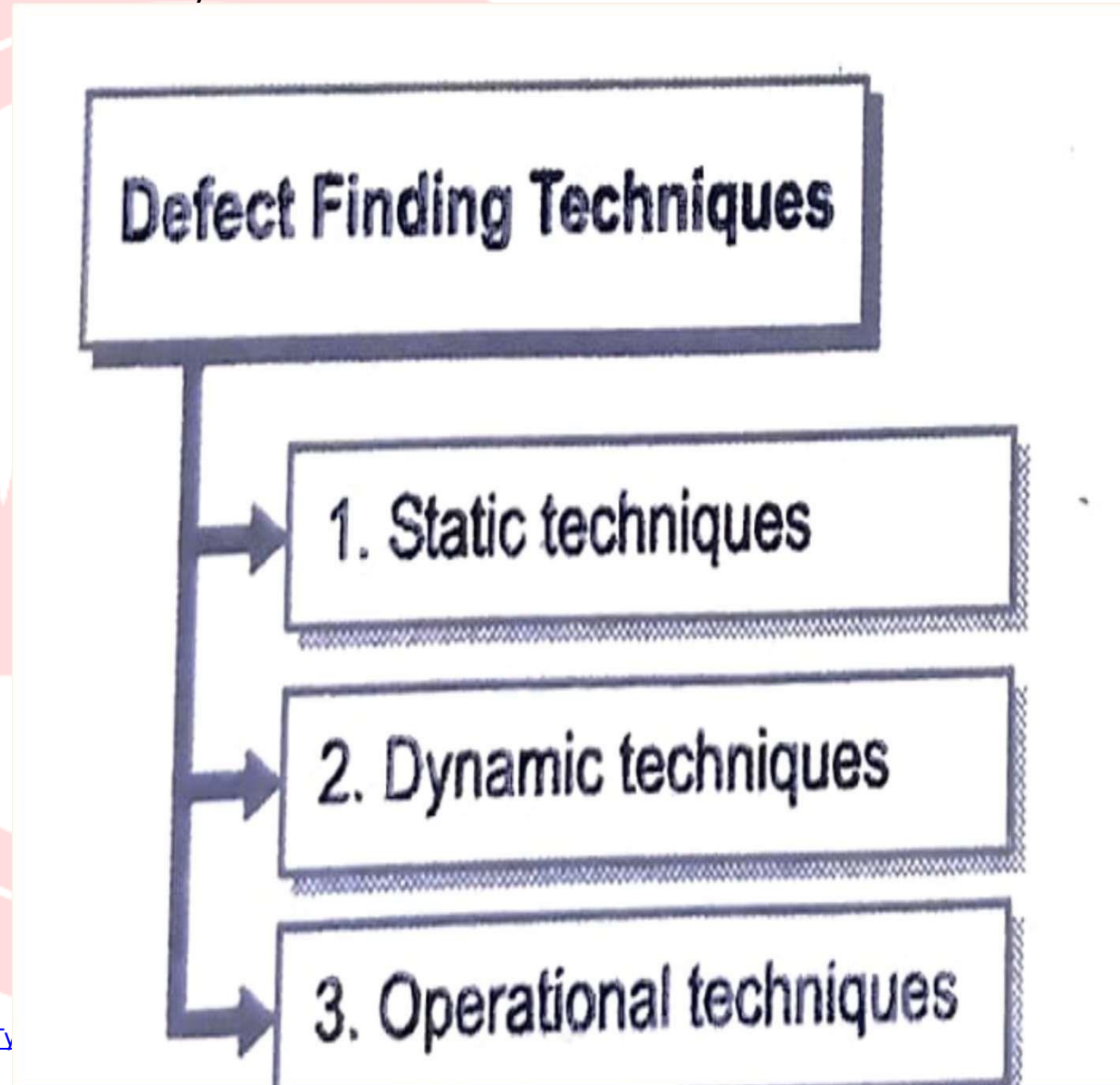
**V2V EDTECH LLP**  
DIPLOMA | DEGREE | BSCIT/CS

# Techniques for Finding Defects





- Defects are found either by:





1. preplanned activities or
  2. by accidents.
- Techniques to find defects can be divided into three categories :
    1. Static Technique:
      - Testing that is done without physically executing a program or system.



–Eg. A code review, walkthroughs, inspection, and audits.

## 2. Dynamic Technique

–Testing in which system components are **physically executed** to identify defects.

–Eg. Execution of test cases.

–The product is evaluated and mark it as pass or fail.

## 3. Operational Technique



- An operational system produces deliverable containing a defect found by users, customers, or control personnel.
- i.e., the defect is found as a result of a failure.
- Smoke testing (also called build verification testing or confidence testing) is considered as common operational technique.





**V2V EDTECH LLP**

DIPLOMA | DEGREE | BSCIT/CS



## Reporting a Defect

- Aim of software engineers is to produce error free end product that fulfils all demands of client.



- The defects in the software applications are detected and resolved through various testing methods, techniques and tools.

Any serious bug can harm the reputation of developer as well as the organization.

- to make sure all defects are properly monitored and tackled during the software development process, software engineers prepare a defect report.



- which provides the detailed information of the defects and errors found during the Software Development Life Cycle (SDLC).

### Defining Defect Reports:

- Defects reports are tremendously essential to software testing.
- They communicate issues and errors found in the software to the developers and other stakeholders, to



ensure transparency and clear communication between them.

- When initially discovered, one can confuse defects with an undocumented requirements change, a user error, or a misunderstanding.
- it is **important** that the defect is quickly identified and brought to the developers' attention.
- testers generate a formal defect report to state the problem as clearly as possible.



- A defect report should:
  - Be well written.
  - Should have sufficient and high quality information to reproduce and fix defects in the software.



- The report should also enable stakeholders to make wise decisions about the defects that need to be fixed.
- Importance of Defect Reports:
  - It documents an irregularity discovered during testing.



- It includes all the information needed to reproduce the problem, including the author, release/build number, open/close dates, problem area, problem description, test environment, defect type, how it was detected, who detected it, priority, severity, status, and more.
- Defect reports contain details of actions in the software application that don't provide expected results.





- Other reasons that make defect reporting important are:
- For Developers: Information regarding the issues and defects is provided to the developers through these defect reports, which further helps them in understanding what they need to do' to reproduce the problem. It also makes them aware of what needs to be done to fix defects in the software.





- For Testers : If a developer or another tester needs to have more information about a bug, a tester can give it to them and provide experienced assistance.
- For Other Testers: With the assistance of defect reports, other testers are also able to



handle and fix similar problems and defects if they arise again in the future.

Sr. No	Question	Marks
1	Which parameters are considered while writing good defect report?	W-15,W-17, 4 marks



2	What are different points to be noted in reporting defects	W-16, 4 marks
---	--	---------------





- **Things to include while Creating Defect Report**
  - To ensure that the defect report is perfect and conveys all the necessary information, it is essential to include some important details in it.
  - The elements that should be included in a defect report are :
    - Testing Condition :



- Summary of the condition in which the testing was executed
- **Eg.** Tester testing a webpage should know the browser to be used
- Steps Taken to Resolve issues**
- Report should contain 
  - a clear breakdown of exactly what was done to detect and resolve the issues.



- information regarding the used methods, techniques and tools should be included.
  - the outcome of each step was, and whether that outcome was expected or not.
- Steps Indicating Successful Defect Duplication :
- the details of how many times did the issue reoccurred during the whole testing process



–Any additional details :

- Information that can improve the quality of the report should be included, such as charts, screenshots and more
- Reporting Defects Effectively:

- To assure that the defects are reported effectively.
- It is important for testers to follow a set guideline while preparing defect report.



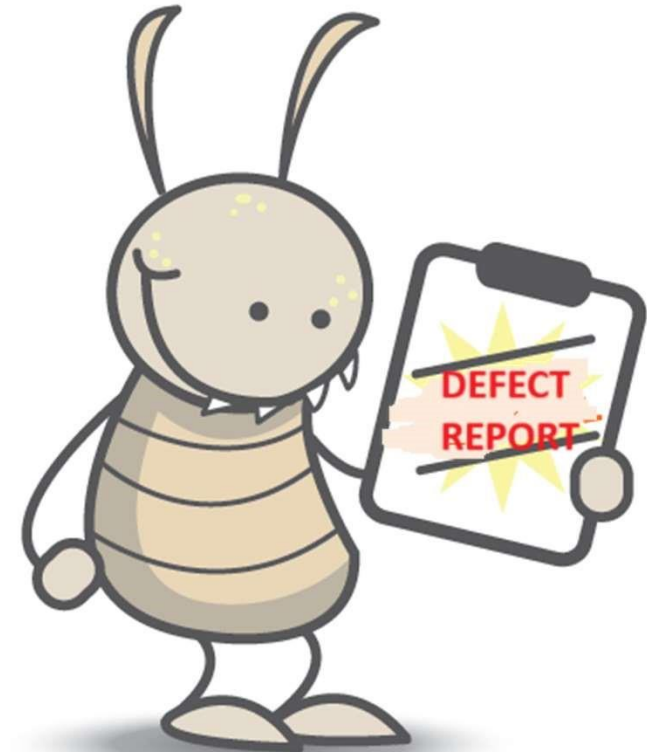
## Be Specific :

- The defect description should be on point.
- the exact action should be specified in the report.

## Provide Information in Detail :

- It is extremely important for the testers and developers to provide detailed information regarding the defects found in the software.

- Be Objective:







- Do not make subjective statements and provide as many facts as possible, without including random information.
- Steps to Reproduce the Defects:
  - While filing the defect one should add the proper steps to reproduce the defect and should not rush the process.
- Defect Tagging:
  - Add the references to the specifications or other related defect number while adding the defect.

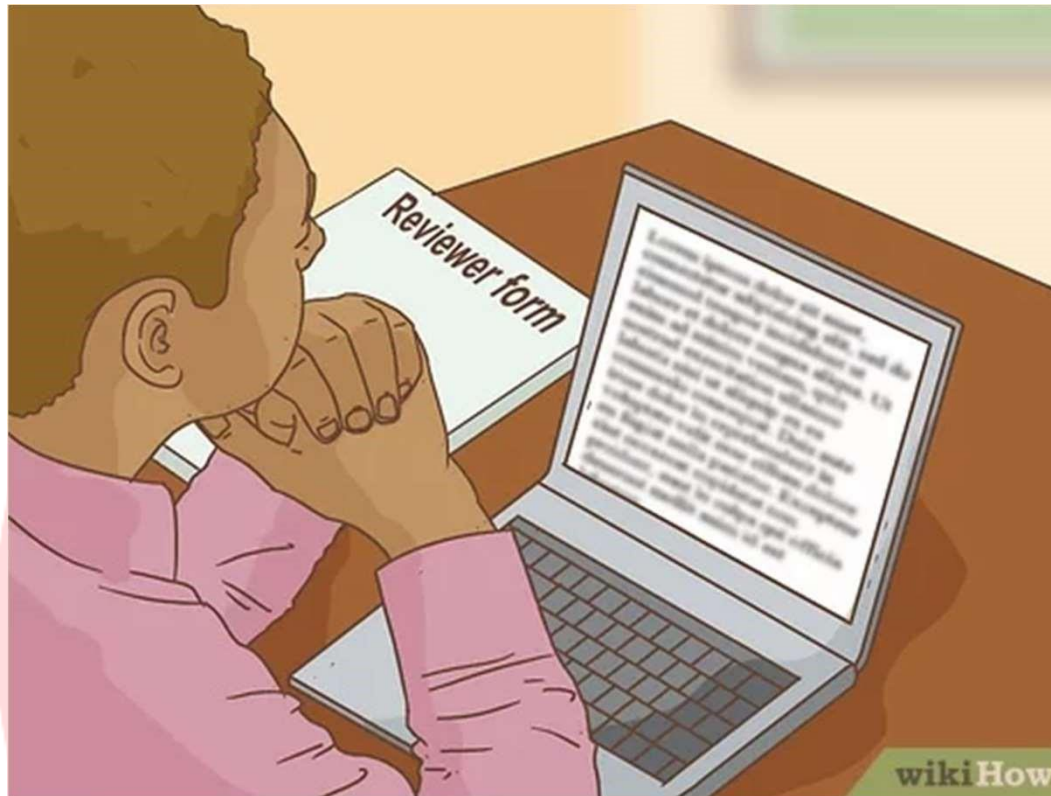


- Review the Report :
  - Before submitting the report developer or stakeholder should review the report for any errors and discrepancies.
  - This validates the quality as well as the information of the report.



**V2V EDTECH LLP**

DIPLOMA | DEGREE | BSCIT/CS



Sr.  
No

Question

Marks



1	Write contents of defect template	W-15,W-17, 4 marks
2	Explain the of defect template with its attributes.	W-16, S-18, 4 marks
3	Enlist any six attributes of defect, describe with suitable example	S-17, 4 marks
4	Explain the of defect report template with its attributes.	W-18, 4 marks



- **Defect Template**
- DEFECT REPORT is a document that identifies and describes a defect detected by a tester.
- The purpose of a defect report is to state the problem clearly.
- Defect Report Template
  - In most companies, a defect reporting tool is used and the elements of a report can vary.
  - A defect report can consist of the following elements



# V2V EDTECH LLP

DIPLOMA | DEGREE | BSCIT/CS

<b>ID</b>	Unique identifier given to the defect. (Usually, automated)
<b>Project</b>	Project name.
<b>Product</b>	Product name.
<b>Release Version</b>	Release version of the product. (e.g. 1.2.3)
<b>Module</b>	Specific module of the product where the defect was detected.
<b>Detected Build Version</b>	Build version of the product where the defect was detected (e.g. 1.2.3.5)
<b>Summary</b>	Summary of the defect. Keep this clear and concise.
<b>Description</b>	Detailed description of the defect. Describe as much as possible but without repeating anything or using complex words. Keep it simple but comprehensive.
<b>Steps to Replicate</b>	Step by step description of the way to reproduce the defect. Number the steps.
<b>Actual Result</b>	The actual result you received when you followed the steps.
<b>Expected Results</b>	The expected results.
<b>Attachments</b>	Attach any additional information like screenshots and logs.
<b>Remarks</b>	Any additional comments on the defect.
<b>Defect Severity</b>	Severity of the Defect. (See Defect Severity)
<b>Defect Priority</b>	Priority of the Defect. (See Defect Priority)
<b>Reported By</b>	The name of the person who reported the defect.
<b>Assigned To</b>	The name of the person that is assigned to analyze/fix the defect.
<b>Status</b>	The status of the defect. (See Defect Life Cycle)
<b>Fixed Build Version</b>	Build version of the product where the defect was fixed (e.g. 1.2.3.9)