

Unit- II

Basics of Software Testing

- Dr. Shivani Budhkar

Defect in Software Testing

- When the application is not working as per the requirement is known as **defects**.
- It is specified as the irregularity from the **actual and expected result** of the application or software.
- The **Defect** is the difference between the actual outcomes and expected outputs.
- The **Test engineer** can identify the defect, and it was fixed by the developer in the development phase of the **software development life cycle**.
- When a test engineer tests a piece of code, he/she comes across differences in expected output to the existing output, which is known as a **defect**. And the substitute of defect can be further known as **issues, bugs, and incidents** in software testing.

Defect Management Process (DMP)

- The **defect management process** is the core of software testing.
- Once the defects have been identified, the most significant activity for any organization is to manage the flaws, not only for the testing team but also for everyone involved in the software development or project management process.
- **Defect prevention** is an effective and efficient way to decrease the number of defects.
- The defect prevention is a very cost-effective process to fix those defects discovered in the earlier stages of software processes.
- The **Defect Management Process** is process where most of the organizations manage the **Defect Discovery**, **Defect Removal**, and then the **Process Improvement**.
- **Defect Management Process (DMP)** manages defects by purely detecting and resolving or fixing the faults.
- It is impossible to make a software 100% error or defect-free, but several defects can be declined by fixing or resolving them.
- The defect management process primarily focuses on stopping defects, finding defects in the earlier stages, and moderating the effect of defects.

The Objective of Defect Management Process (DMP)

- The primary objective of DMP is to expose the defects at an early stage of the software development process.
- The execution of the defect management process will help us enhance the process and implementation of software.
- The defect management process reduces the impact or effects of defects on software.
- The Defect management process (DMP) helps us to avoid defects.
- The main goal of the Defect management process is to resolve or fixing the defects.

Stages of Defect Management Process

- Defect Prevention
 - Estimate Predictable Impact
 - Minimize expected impact
 - Identify Critical Risk
- Deliverable Baseline
- Defect Discovery
 - Identify a defect
 - Report a defect
 - Acknowledge Defect
- Defect Resolution
 - Prioritize the risk
 - Fix the defect
 - Report the Resolution
- Process Improvement
- Management Reporting

Advantages of Defect Management Process

- Confirm Resolution
- Accessibility of Automation Tools
- Offer Valuable Metrics

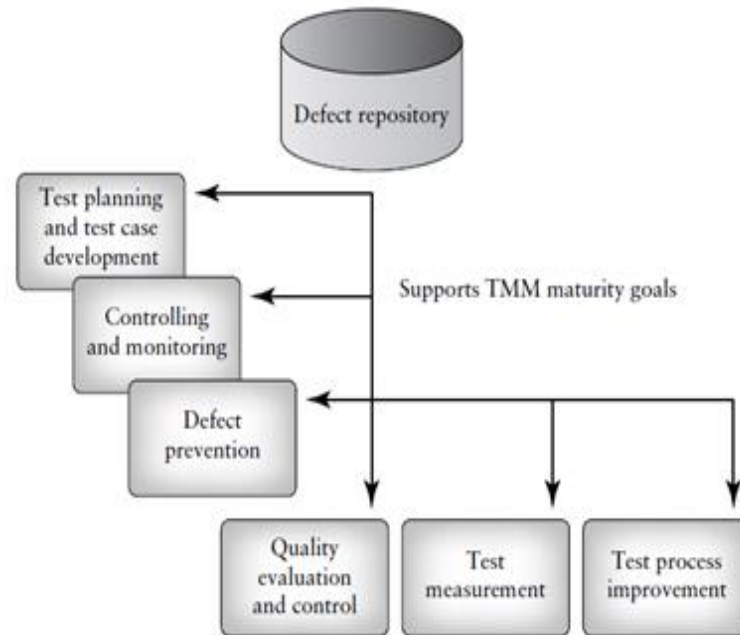
Disadvantages of Defect Management Process

- If the defect management process is not performed appropriately, then we may have a loss of customers, loss of revenue, and damaged brand reputations.
- If the defect management process is not handled properly, then there will be a huge amplified cost in a creeping that is a rise in the price of the product.
- If defects are not accomplished appropriately at an early stage, then afterward, the defect might cause greater damage, and costs to fix the defects will also get enhanced.

Developer/Tester Support for Developing a Defect Repository

- Software engineers and test specialists should follow the examples of engineers in other disciplines who make use of defect data. A requirement for repository development should be a part of testing and/or debugging policy statements.
- Forms and templates should be designed to collect the data. Each defect and frequency of occurrence must be recorded after testing.
- Defect monitoring should be done for each on-going project. The distribution of defects will change when changes are made to the process.
- The defect data is useful for test planning. It is a TMM level 2 maturity goal. It helps a tester to select applicable testing techniques, design the test cases, and allocate the amount of resources needed to detect and remove defects. This allows tester to estimate testing schedules and costs.
- The defect data can support debugging activities also. A defect repository can help in implementing several TMM maturity goals including
 - Controlling and monitoring of test,
 - Software quality evaluation and control,
 - Test measurement,
 - Test process improvement.

The Defect Repository and Support for TMM Maturity Goals



Defect Repository Template

- Defect Name/id:
- Release name:
- Project name:
- Module name:
- Status:
- Severity:
- Priority:
- Test case id/name:
- Brief description :
- Detail description:
- Following are the steps to reproduce the defects
 - 1.
 - 2.
 - 3.
 - 4.
 - 5.
- Expected result :
- Actual result:

Defect Repository Template

- **Defect ID** - Unique identification number for the defect.
- **Defect Description** - Detailed description of the Defect including information about the module in which Defect was found.
- **Version** - Version of the application in which defect was found.
- **Steps** - Detailed steps along with screenshots with which the developer can reproduce the defects.
- **Date Raised** - Date when the defect is raised
- **Reference**- where in you Provide reference to the documents like . requirements, design, architecture or maybe even screenshots of the error to help understand the defect
- **Detected By** - Name/ID of the tester who raised the defect
- **Status** - Status of the defect , more on this later
- **Fixed by** - Name/ID of the developer who fixed it
- **Date Closed** - Date when the defect is closed
- **Severity** which describes the impact of the defect on the application
- **Priority** which is related to defect fixing urgency. Severity Priority could be High/Medium/Low based on the impact urgency at which the defect should be fixed respectively

Defect Life cycle

- **Defect Life Cycle** or Bug Life Cycle in software testing is the specific set of states that defect or bug goes through in its entire life.
- The purpose of Defect life cycle is to easily coordinate and communicate current status of defect which changes to various assignees and make the defect fixing process systematic and efficient.
- **Defect Status** or Bug Status in defect life cycle is the present state from which the defect or a bug is currently undergoing.
- The goal of defect status is to precisely convey the current state or progress of a defect or bug in order to better track and understand the actual progress of the defect life cycle.
- New
- Assigned
- Open
- Fixed
- Pending retest
- Retest
- Verified
- Reopen
- Closed
- Duplicate
- Rejected
- Deferred
- Not a bug

Defect Life cycle

