**Create Defect Report:**

1. In defect report, “Submitted by”, “Defect Logged date” along with “Release version” are key information that will help during **Defect tracking phase.**
2. Dynamic defect can be identified during **Test Execution phase** only. When the result of a test case does not matches with the expected one, then it is a **dynamic defect**.
3. **Static defects** can be identified **during Requirement analysis phase, Test design phase and Test Closure phase.**
4. Types of defects that can be found in static testing phase – **Variables that is not used, uncalled functions/procedures and programming standard violations.**
5. **Defect Severity Chart** – Track and recognize the number of defects of greatest importance and to ensure defect resolution is progressing in line.
6. In a defect report, **Reviewer name** is not really required when a defect report is sent daily. But Defect count across severity, Defect ID, Description, Severity, Status & comments are some of the key variables to be mentioned in a defect report.
7. **Priority chart** in a defect report helps to verify the most important defects are resolved first.
8. **Severity chart** in a defect report helps to verify the defects are resolved based on software impact.
9. **QC, ALM** – Tools acts as a repository for tracking requirements, test sets, test runs & defects.
10. **Defect rejection ratio** can increase with increase in invalid defects.
11. Customer escalations on defect report quality can be avoided by **reviewing the report** before submitting and to ensure the quality of the defect report.
12. **Defect Fix Rejection Ratio** – Metric to generate report status for multiple time re-opened defects.
13. **Defect report is useful in process improvement.**
14. Defect Prevention – **Analyze defects or errors to trace the root causes**

**Suggest preventive actions to eliminate the defect root causes**

**Implement the preventive actions**

1. **Defect status –** Component in defect report to understand the progress of defect

**Defect Life Cycle:**

1. During defect retesting, tester A retested and closed a defect as it is working fine. But tester B noticed that fix made to defect was not working in his machine. It was then identified as an environmental issue. In this situation, tester B should reopen a defect and add comment as environmental issue.
2. In case if the tester B is facing the same issue in the next phase of testing, **a new defect needs to be raised.**
3. **Masked defect** – An existing problem that hasn’t yet caused a failure just because a defect has prevented that part of the code from being executed.
4. **Latent defect** – An existing problem in the system over a longer period of time even after the release of the software and just waiting to reveal themselves.
5. During system testing, a defect is raised with high severity. This is analyzed by the development team and is rejected as an invalid defect. But if the **tester has all the supporting documents with valid explanations and can reopen the defect**. During triage meetings, it would be validated and decided whether it needs to be fixed or not.
6. During system testing, if a defect is rejected by the developer, **the tester should look for the loopholes in the original defect report that could be supported with further investigative data and provide additional details in the defect log with supporting screenshots that will help the developer.**
7. While logging a defect, the tester **should not provide** **high level description** of the defect assuming that developer would understand by reading steps to replicate but should give **detailed description of the defect so that the developer can understand and follow the steps to replicate the defect.**
8. Test Lead is reviewing the list of defects rejected by the development team. Follow up action could be – **Analyze the reason for rejection and try to reproduce the defect and provide sufficient details if the defect is a valid one**.
9. Generally a **tester** assigns the **severity** and **Product manager/ triage team** decides the **priority**.
10. **Critical defect** – severe defect which causes termination of a complete system or one or more components of the system and causes extensive corruption of data.
11. **High priority, high severity:** Any defects due to which the testing cannot continue at any cost or causes a severe system failure fall under this category.

**Example:** site maintaining the student details, on saving record if it, doesn’t allow to save the record then this is high priority and high severity bug.

Or 20 rupees transactional charges are being made to the customers for every withdrawal transaction when it is not intended to do.

1. **High priority, low severity:** Defects which have to be fixed but do not affect the application come under this category.

**Example:** A logo error for any shipment website (or) the spelling mistakes on cover page or in the title of the application, can be of low severity as it not going to affect the functionality of the website but can be of high priority as you don't want any further shipment to proceed with wrong logo.

1. **High Severity, low priority:** An error which occurs on the functionality of the application (for which there is no workaround) and will not allow the user to use the system but on click of link which is rarely used by the end user.

**Example:** Delete functionality of a webpage is not working which is rarely used by the customer.

1. **Low severity, low priority:** Any cosmetic or spelling issues which is within a paragraph or in the report (Not on cover page, heading, title).
2. While reviewing a defect raised by the testing team, the development team was able to find that a defect is repeated twice or the defect corresponds to the same concept of another defect. Status of the defect should be **Duplicate**.
3. Steps to replicate field in the defect report **is to assist the development team** but **not the testing team to simulate the defect in development environment.**
4. A defect was rejected by the developer stating “**Missing requirement**”. Later the missing requirement was implemented as change request in Phase 2 and the defect raised was still found during phase 2 execution. In this case, a new defect should be raised in phase 2 and the same should be linked to the defect in phase 1.
5. In a project, tester logged a defect stating that the actual result does not meet the expected result and the developer has rejected the defect. Possible reasons could be – **No document available in the requirement as per the expected result mentioned by the tester or Defects logged were out of scope requirements.**

Some other possible reasons are missing requirement, incorrect requirement understanding and incorrect test data, different configuration or browser version is being tested.

1. **Priority and severity** of the defect helps developers to prioritize the defects logged by the testing team and work accordingly to provide fixes.
2. During triage meeting, team will **not just discuss** about **the “new” defects** raised for the day but also about all the open defects.
3. In defect status, **Fixed, Deferred and Rejected** are the status that a developer can give.
4. If a defect has no feasibility to be fixed in this release, the status would be **deferred**.
5. **Severity** of a defect **does not depend** on its priority.
6. **Severity** of the defect is assessed on the impact of the defect on software.
7. **Severity** helps to decide the stability of the application and analyze the risk in the project.
8. Providing walkthrough and inspection of the requirements will help in identifying the defects by static testing to determine issues earlier in project life cycle.
9. Cost of defect fixes in different phases arranged in ascending order below:

* **Requirement gathering(least expensive)**
* **Design**
* **Coding**
* **Testing(most expensive)**

1. **Project name & Module name** – Defect components help to identify which project the defect belongs to and which module gets impacted.
2. **Defects under functional requirements** – Requirements that are not testable and documentation errors.
3. STLC phases:
4. Requirement Development
5. High level design
6. Detail design
7. Coding
8. Unit testing
9. Integration testing
10. System testing
11. Acceptance testing

**Defect Age – Phases:**

Defect age in phases is defined as the difference between defect injection phase and defect detection phase. **Defect Age in Phase = Defect Detection Phase - Defect Injection Phase**

If a defect is found in Requirement development phase and fixed in System testing phase, then defect age is 7 – 1 = 6.

**Defect Age – Days:**

**Defect age is defined as the time difference between defects detected date and the current date,** provided the defect is still said to be open.

If a defect was detected on 05/05/2013 11:30:00 AM and closed on 23/05/2013 12:00:00 PM,

Defect Age in Days = 05/05/2013 11:30:00 AM - 23/05/2013 12:00:00 PM

Defect Age in Days = 19 days

1. A shopping website application is being tested. There is a release to production planned for tomorrow and a high critical defect is found. Dev team fixes the defect immediately and moves it to resolve. Being a last minute fix, the tester should verify the defect fix is working and do a risk based testing to test the critical scenarios.
2. **Test Lead** – Responsible to change a defect from New to open state when it is found to be valid.
3. In business sensitive projects like banking, stock exchange etc., all defects need not be treated as priority as it will have its testing environment where the actual testing goes live.
4. **Inspection** – Test technique that can detect defects in the requirements specification at the earliest possible time.
5. **Inspection** – Formal assessment of a work product conducted by one or more qualified independent review to detect defects.
6. While logging defects, defect screenshots should be provided **and defect summary, defect description, actual result, expected result** should also be provided.
7. During execution phase, most of the defects were rejected due to **documentation errors**. This could have been avoided by identifying the defects at the early stage of SDLC phases.
8. **Defect log** need not necessarily be a word document.
9. If a new defect is reported by the customer in production,
10. **If it is not reported by testing team, do RCA**
11. **If it is already reported by testing team and defect is open, close the defect as duplicate**
12. **If it is a miss from testing team, check the test case corresponding to the defect and add a test case If the scenario is missed.**
13. Common cause of defect is **ambiguous or incomplete requirements**
14. Possible final status of defects – **Resolved, Rejected, Closed and Deferred**
15. For a defect whose fix is progress, the status of the defect should be **In Progress.**