Defect Management Process

In this section, we are going to understand the working of the **Defect management process.**

Also, see the defect in software testing**, defect management process's objective**, **defect management process, advantages and disadvantages of the defect management process.**

But, firstly, we will understand the process of defect management, and we will understand the **defect in software testing**.

Defect in Software Testing

* The bug announced by the **programmer**and inside the code is called a
* In other words, we can say that when the application is not working as per the requirement is knows 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.

What is Defect Management Process?

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.

As we know, **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**.

As the name recommends**, the 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 main objective of the defect management process is as discussed below:

* 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.

## Various Stages of Defect Management Process

The defect management process includes several stages, which are as follows:

1. **Defect Prevention**
2. **Deliverable Baseline**
3. **Defect Discovery**
4. **Defect Resolution**
5. **Process Improvement**
6. **Management Reporting**

Let's discuss them one by one:



### **1. Defect Prevention**

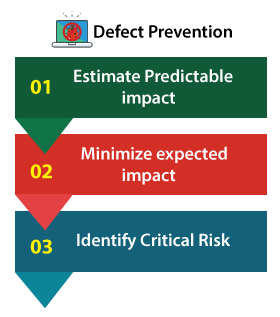
The first stage of the **defect management process** is **defect prevention**. In this stage, the execution of procedures, methodology, and standard approaches decreases the risk of defects. Defect removal at the initial phase is the best approach in order to reduction its impact.

Because in the initial phase of fixing or resolving defects is less expensive, and the impact can also be diminished.

But for the future phases, identifying faults and then fixing it is an expensive process, and the effects of defect can also be amplified.

The defect prevention stage includes the following significant steps:

* **Estimate Predictable Impact**
* **Minimize expected impact**
* **Identify Critical Risk**



**Step1: Estimate Predictable Impact**

In this step, if the risk is encountered, then we can calculate the estimated financial impact for every critical occasion.

**Step2: Minimize expected impact**

When all the critical risk has been discovered, we can take the topmost risks that may be dangerous to the system if encountered and try to diminish or eliminate it.

Those risks that cannot be removed will decrease the possibility of existence and its financial impact.

AD

**Step3: Identify Critical Risk**

In defect prevention, we can quickly identify the system's critical risks that will affect more if they happened throughout the testing or in the future stage.

### **2. Deliverable Baseline**

The second stage of the defect management process is the **Deliverable baseline.** Here, the deliverable defines the **system, documents, or product**.

We can say that the **deliverable is a baseline** as soon as a deliverable reaches its pre-defined milestone.

#### **Note: The Pre-defined milestone describes what the software is supposed to accomplish.**

In this stage, the deliverable is carried from one step to another; the system's existing defects also move forward to the next step or a milestone.

In other words, we can say that once a deliverable is baselined, any additional changes are controlled.

### **3. Defect Discovery**

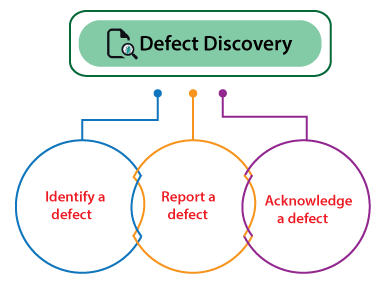
The next stage of the defect management process is **defect discovery**. At the early stage of the defect management process, defect discovery is very significant. And later, it might cause greater damage.

If developers have approved or documented the defect as a valid one, then only a defect is considered as **discovered**.

As we understood that, it is practically impossible to eliminate each defect from the system and make a system defect-free. But we can detect the defects early before they become expensive to the project.

The following phases have been including in the defect discovery stage; let's understand them in details:

* **Identify a defect**
* **Report a defect**
* **Acknowledge Defect**



**Phase1: Identify a Defect**

In the first phase of defect discovery where we need to find the defects before becoming a critical problem.

**Phase2: Report a Defect**

The moment testing team identifies a defect, they need to assign known issues to the development team for further evaluation and fixing process.

**Phase3: Acknowledge Defect**

Once the test engineers' hand over the defect to the assigned developers, now it is the responsibility of development teams to acknowledge the fault and remain further to fix it if the defect is a valid one.

### **4. Defect Resolution**

Once the **defect discovery** stage has been completed successfully, we move to the next step of the defect management process, **Defect Resolution**.

The **Defect Resolution** is a step-by-step procedure of fixing the defects, or we can say that this process is beneficial in order to specified and track the defects.

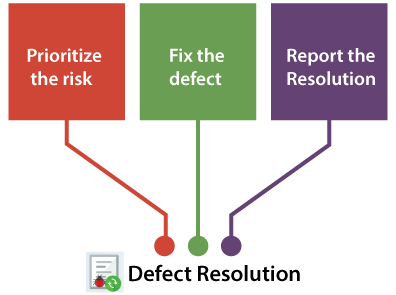
This process begins with handing over the defects to the development team. The developers need to proceed with the resolution of the defect and fixed them based on the **priority.**

Once the defect has been selected, the developer sends a defect report of resolution to the test manager's testing team.

The defect resolution process also involves the notification back to the test engineer to confirm that the resolution is verified.

We need to follow the below steps in order to accomplish the defect resolution stage.

* **Prioritize the risk**
* **Fix the defect**
* **Report the Resolution**



**Step1: Prioritize the risk**

In the first step of defect resolution, the development team evaluates the defects

and arranges the fault's fixing. If a defect is more impactful on the system, then developers need to fix those defects on a high priority.

**Step2: Fix the defect**

In the second step, the developer will fix the defects as per the priority, which implies that the higher priority defects are resolved first. Then the developer will fix the lower priority defects.

**Step3: Report the Resolution**

In the last step of defect resolution, the developer needs to send the fixed defects report. As it is the responsibility of development teams to make sure that the testing team is well aware of when the defects are going to be fixed and how the fault has been fixed.

This step will be beneficial for the testing team's perspective to understand the root of the defect.

### **5. Process Improvement**

In the above stage (defect resolution), the defects have been arranged and fixed.

Now, in the **process improvement** phase, we will look into the lower priority defects because these defects are also essential as well as impact the system.

All the acknowledged defects are equal to a critical defect from the process improvement phase perspective and need to be fixed.

The people involved in this particular stage need to recall and check from where the defect was initiated.

Depending on that, we can make modifications in the **validation process, base-lining document, review process** that may find the flaws early in the process, and make the process less costly.

These minor defects allow us to learn how we can enhance the process and avoid the existence of any kind of defects that may affect the system or the product failure in the future.

### **6. Management Reporting**

**Management reporting** is the last stage of the **defect management process**. It is a significant and essential part of the defect management process. The management reporting is required to make sure that the generated reports have an objective and increase the defect management process.

In simple words, we can say that the evaluation and reporting of defect information support organization and risk management, process improvement, and project management.

The information collected on specific defects by the project teams is the root of the management reporting. Therefore, every organization needs to consider the information gathered throughout the defect management process and the grouping of individual defects.

## Defect Workflow and States

Various organizations that achieve the **software testing** with the help of a tool that keeps track of the defects during the **bug/defect lifecycle** and also contains defect reports.

Generally, one owner of the defects reports at each state of **defect lifecycle**, responsible for finishing a task that would move defect report to the successive state.

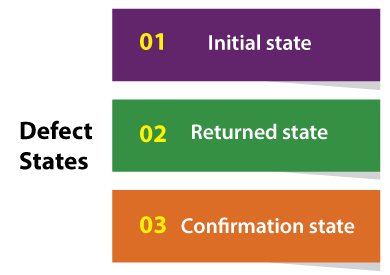
Sometimes, **defect report may not have an owner** in the last phases of the defect lifecycle if we may face the following situation:

* If the defect is invalid, then the Defect report is **cancelled**.
* The defect report is considered **deferred** if the defect won't be fixed as part of the project.
* If the fault cannot be detected anymore, hence defect report is regarded as not **reproducible**.
* The defect report is considered **closed** if the defect has been fixed and tested.

### **Defect States**

If defects are identified throughout the testing, the testing team must manage them in the following three states:

* **Initial state**
* **Returned state**
* **Confirmation state**



### **1. Initial State**

* It is the first state of defect, which is also known as open state.
* One or several test engineers are responsible for collecting all required data to fix the defects in this state.

### **2. Returned state**

* The second state of defect is **returned state**. In this, the person receiving the test report rejects and asks the report creator to provide further information.
* In a returned state, the test engineers can provide more information or accept the rejection of the report.
* If various reports are rejected, the test manager should look out for faults in the initial information collection process itself.
* The returned state is also referred as the **clarification state or rejected state**.

### **3. Confirmation state**

* The last state of defect is **the confirmation state**, where the test engineer performed a **confirmation testing**to make sure that the defect has been fixed.
* It is achieved by repeating the steps, which found the defect at the time of testing.
* If the defect is resolved, then the report is closed.
* And if the defect was not resolved, then the report is considered as **re-opened** and reported back to the owner who formerly preserved the defect report for fixing.
* A confirmation state is also known as **a verified or resolved state**.

Following topic Taken from book Software Quality Assurance Integrating Testing, Security, and Audit, Abu Sayed Mahfuz, CRC Press Taylor & Francis Group, 2016.

