

# Import Sourcing of Defect Life Cycle and Defect Management Process

Dr V.Goutham

Department of Computer Science and Engineering  
TKR Engineering College, JNTU Hyderabad  
Telangana - India

## ABSTRACT

The Defect Management Process provides a single reference document that defines the methods for tracking defects in the test environment. This paper details the information necessary to understand the flow of the defect process and how to document defects properly. The purpose of this paper will be to communicate and assure understanding of the Defect Process Flow, explain the different stages of the Process Flow and the associated SLA's at each stage, demonstrate the appropriate access rights for each stage and explain how software tools are utilized in the Defect management process.

**Keywords:-** Defect, Defect Management, Information Technology Infrastructure Library, Defect Life Cycle, Service level Agreement.

## I. INTRODUCTION

Distinct organizations are using different defect management process model presented by the IT Infrastructure Library (ITIL) [1]. The major fault of ITIL model is that it does not consider the customer as a related participant of the defect management process. Defect life cycle is a cycle which a defect goes through during its lifetime. It starts when defect is found and ends when a defect is closed, after ensuring it's not reproduced. Defect life cycle is related to the bug found during testing. Defect Management tracks and manages the discovery, resolution and retest of system defects identified during test execution. This process involves: Recording, reviewing and prioritizing defects, Assigning defects to developers for fixing, Assigning Test Analysts to retest fixes. This paper provides an overview of Import Sourcing Defect Life Cycle and Defect Management process..

## II. APPROACH

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

### A. Roles and Responsibilities

Defect tracking can be tedious, yet comparing tracked defects can also help testers improve their work. Each team will select a primary and secondary point of contact (POC) to facilitate the testing for their particular area. Each POC, and their respective contact information, shall be made available to all domain teams and the triage.

The standard test management tool used is Mercury Interactive Test Director 8.0. Test Director provides a single, web-based application for all essential aspects of test management — Requirements Management, Test Plan, Test Lab, and Defects Management

Role	Responsibilities
Tester	Identifying and logging defects Retesting defect fixes
Test Lead	Managing defects opened by test team Reviewing new and returned defects Identifying duplicate defects Participating in Tier 2 Triage
Development Lead	Assigning new defects to developers Participating in Tier 2 Triage
Developer	Updates code to fix defects Updates defects with relevant information
Domain Lead	Takes care of Change Requests
Domain Analyst	Final Review

**Table 1: Roles and Responsibility Matrix**

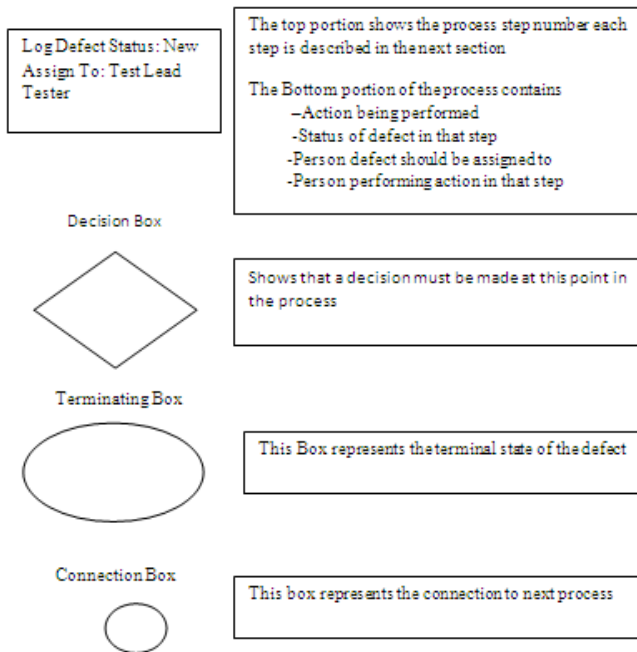
### B. Defect Management Process Flow:

The level of defect prevention in this process has specific techniques for example orthogonal defect classification (ODC) [2], [3] is used to identify the defects and their root causes. The ultimate goal of this defect prevention technique is to

prevent the defects from recurring in the future. The defect found in the first two levels can also be used in defect prevention to eliminate the root causes of defects [4]. The following figures provide a graphical representation of the defect management lifecycle.

The flow can be utilized as an aid in following the defect processes. The defect management lifecycle workflow will present an overview of the entire defect management process.

**Description of Symbols**



**C. Defect Lifecycle:**

This section outlines the process for creating, retesting and closing defects. The main resources involved in this stage are the Tester, Test Lead, Developer, Development Lead, Domain Lead and Domain ANALYST. Defects may be routed to the Development Team, or the Domain ANALYST for resolution. When defects are fixed they are routed back to the Tester for retest and verification. During test execution, team members may encounter application defects. In order to ensure that the development team can recreate the defect, the testing team must document the entire process on how the defect was generated. This includes screen shots, details of data used, and a step-by-step description of what was performed [5].

The Test Lead will review the defects and re-assign severity/priority to the defects and submit the defect to the Development Lead with the status as Open. The Development

Lead will review the defect and assign it to the appropriate Developer, with the status as "In Progress". The Development Lead may reject the defect by changing the status to "Rejected" with documented reasons for rejecting it. The Development Lead may keep the defect On Hold due to a change in the requirements. The Change Request Process will be followed in this scenario.

After the defect is fixed, the Developer informs the Development Lead of the fix by changing the status to Fixed with the details of the fix in the Comments section. The Development Lead will schedule the fix to be part of a release and ensure that the release coordinator includes the revised code in the appropriate release. All defect fixes should be recorded in the appropriate release report/note/mail.

The Development Lead will update the release code and change the state of the defects from Fixed to Retest status. This is the signal to the Tester to retest the script(s). If the error no longer occurs, the defect should be changed to "Verified" with comments of a successful retest. Otherwise, the tester will change the status to ReOpen with additional comments for the failure and the process will begin again, until the defect is fully resolved.

All defects in Rejected or Verified status will be reviewed and closed by the Domain ANALYST. If the Domain ANALYST does not agree with the recommended closure of the defect, he may ReOpen a defect with information for declining.

**D. Test Director Guidelines:**

All Defects will be tracked using Test Director, a single, Web-based application for all essential aspects of test management — Requirements Management, Test Plan, Test Lab, and Defects Management.

The Test Director tool will be used along with the process flow to help guide and maintain order in the defect management process. Everyone will be required to obtain a user ID and password to access the software. This user ID will have associated access rights and permissions within the Test Director software. These permissions will allow the user to access the necessary actions needed to execute their duties within the software and also help to control access to areas that are not necessary for the user to complete their job. This document will touch on the key areas in order to create and manage defects in the Test Director software and not an all-inclusive guide for using the software

**E. Defect Standards:**

When a defect is discovered there should be a defect created in Test Director. The Tester can click the Add Defect button to create a defect. Each new defect must adhere to the following standards:

**Summary**

The defect Owner must provide a concise summary that includes the application name where the defect was found and should use appropriate keywords. The summary format must be as follows:

Application/Program Name – Brief summary of error received

Detected on Date: This field is auto-populated by Test Director, and represents the date on which the defect was first identified.

Severity: Once defects are identified, they are assigned a severity level which is validated by the Test Lead.

<b>1.1 – Log Defect</b>	
<i>Actor:</i>	Tester
<i>Predecessor(s):</i>	None
<i>Status:</i>	New
<i>Assign To:</i>	Test Lead
When a Test Analyst finds a variation between the expected result and the actual result while running a test case, a defect has been identified. All defects must be logged in Test Director. The defect must adhere to the standards defined in this document. The Tester assigns the defect to the Test Lead for review and assignment	

<b>1.1.1– Review for Closure</b>	
<i>Actor:</i>	Tester
<i>Predecessor(s):</i>	1.1 Log Defect
<i>Status:</i>	Closed
<i>Assign To:</i>	None
In some instances, the Test Lead will determine that a logged defect should not have been logged (i.e. production issues, data issues). These defects will be changed to status “Closed” with comments added to justify Closure.	

<b>1.2 – Review Defect</b>	
<i>Actor:</i>	Test Lead
<i>Predecessor(s):</i>	1.1 Log Defect
<i>Status:</i>	Open
<i>Assign To:</i>	Dev Lead
The Test Lead reviews the defects and assigns severity/ priority to the defects and classifies the defects. The Test Lead will submit the defect to the Development Lead with the status as ‘Open’.	

<b>1.3 – Analysis</b>	
<i>Actor:</i>	Dev Lead
<i>Predecessor(s):</i>	1.2 Review Defect
<i>Status:</i>	In Progress
<i>Assign To:</i>	Developer
The Development Lead reviews the defect and assigns it to the appropriate developer with the status as “In Progress”.	

1.4 – Fix Implemented	
<i>Actor:</i>	Developer
<i>Predecessor(s):</i>	1.3 Analysis
<i>Status:</i>	Fixed
<i>Assign To:</i>	Development Lead
The Developer will provide a fix (coding change, configuration change or other) and updates the defect with the details of the fix implemented with notes on the unit testing successfully performed to validate the fix. Once it is successfully implemented and unit tested, the developer will reassign the defect to the Development Lead with status as 'Fixed'.	

1.5 – Retest	
<i>Actor:</i>	Development Lead
<i>Predecessor(s):</i>	1.4 Fix Implemented
<i>Status:</i>	Retest
<i>Assign To:</i>	Tester
The defect has been successfully fixed and unit tested. The development lead reviews the defect information and verifies the code deployment prior to reassigning the defect to the test team for validation in functional testing.	

1.6 – Verified Defect	
<i>Actor:</i>	Tester
<i>Predecessor(s):</i>	1.5 Retest
<i>Status:</i>	Verified
<i>Assign To:</i>	Test Lead
The tester will verify the implemented fix and changes the status to 'Verified' and assigns it to Test Lead for further validation.	

1.7– Review	
<i>Actor:</i>	Test Lead
<i>Predecessor(s):</i>	1.6 Verified defect
<i>Status:</i>	Verified
<i>Assign To:</i>	Domain ANALYST
The Test Lead reviews the defect marked as "Verified" by Tester prior to presenting to the Domain ANALYST for recommended closure.	

2.1 – Need More Information	
<i>Actor:</i>	Dev Lead
<i>Predecessor(s):</i>	1.2 Review Defect
<i>Status:</i>	Need More Info
<i>Assign To:</i>	Test Lead
In case the Development Lead is unable to understand the Defect or thinks consultation is required with the Testing Team (Test Lead/Tester) before assigning it to Developer then "Need More Info" status is assigned by the Development Lead.	

2.2 – Reopen the Defect	
<i>Actor:</i>	Test Lead
<i>Predecessor(s):</i>	2.1 Need More Information
<i>Status:</i>	Reopen
<i>Assign To:</i>	Dev Lead
Test Lead provides the necessary information sought by the Development Lead or lends more clarity to the Defect and Reopens the Defect.	

<b>3.1 – Change Request</b>	
<i>Actor:</i>	Dev Lead
<i>Predecessor(s):</i>	1.2 Review Defect
<i>Status:</i>	On hold
<i>Assign To:</i>	Domain Lead
<p>The defect has been identified as a missing or new requirement that requires a Change Request (CR). The CR process will be followed with the ticket being changed to status “On Hold” and assigned to the Domain Lead.</p>	

<b>3.2 – Create CR</b>	
<i>Actor:</i>	Domain Lead
<i>Predecessor(s):</i>	3.1 Change Request
<i>Status:</i>	Rejected
<i>Assign To:</i>	Domain ANALYST
<p>Once a CR is created and approved with signatures from the Domain ANALYST, the defect is assigned to the Domain ANALYST with comments indicating the approved CR#.</p>	

<b>3.3 – Mark as CR</b>	
<i>Actor:</i>	Domain ANALYST
<i>Predecessor(s):</i>	3.2 Create CR
<i>Status:</i>	CR
<i>Assign To:</i>	None
<p>The defect status is changed to CR (Change Request) by Domain ANALYST with approved CR# in the comments. This status indicates that Defect is regarded to be an Enhancement, that can be implemented either in the current Release of the application or in the subsequent Releases.</p> <p>This would also give the number of defects that have received the Status of CR, indicating the minor/major features that could not be envisaged in the Requirements document (BRD, HLD, etc).</p>	

<b>4.1 – Retest Failed</b>	
<i>Actor:</i>	Tester
<i>Predecessor(s):</i>	1.5 Retest
<i>Status:</i>	Reopen
<i>Assign To:</i>	Dev Lead
<p>The fix has failed functional testing and is returned to the development lead for further resolution.</p>	

<b>5.1.1 – Not a Defect</b>	
<i>Actor:</i>	Dev lead
<i>Predecessor(s):</i>	5.1 Additional Information
<i>Status:</i>	Rejected
<i>Assign To:</i>	Domain ANALYST
<p>The development lead has identified that the issue is not a defect and submits it to be closed without change. There should be a mandatory “Rejection Cause” field or some “Comment” field for the Development Lead to explain the cause for Rejection. Rejection could be because it is a “Duplicate” defect, “Invalid” defect, “Cannot be Fixed”, due to Browser problems/ OS problems/ Software problems, etc, “Unable to Reproduce” due to insufficient data or incorrect steps . The status would be changed to Rejected and reassigned to the Domain ANALYST for closure.</p>	

5.1 – Additional Information	
Actor:	Developer
Predecessor(s):	1.3 Analysis
Status:	Reopen
Assign To:	Dev Lead
If the Developer needs additional information to fix the defect, the defect status is changed to Reopen and assigned to the Development lead to seek clarifications.	

5.1.2 – ReOpen	
Actor:	Domain ANALYST
Predecessor(s):	5.1 Additional Information
Status:	Reopen
Assign To:	Dev Lead
If the Domain ANALYST does not agree with the assesAnalystnt that a defect is invalid, it may be changed to status “Reopen” and reassigned to the Development Lead with clear reason for the return.	

5.2 – Update Status	
Actor:	Test Lead
Predecessor(s):	5.1 Additional Information
Status:	Verified
Assign To:	Dev Lead
A defect that required additional information, requested by the Developer, will be set to Reopen status. The Development Lead will verify the clarity of the queries or the additional information sought by the Developer and then return the defect to the Test Lead with an updated status of Verified.	

6.1 – Deferred For Next Release	
Actor:	Developer
Predecessor(s):	1.3 Analysis
Status:	Deferred
Assign To:	Dev lead
The Developer in consultation with Development Lead, Domain Lead, Domain ANALYST feels that a Defect cannot be fixed in the current version of the Software then the Defect status is changed to “Deferred”(Deferred for Next Release) and assigned to Development Lead; Here the Development Lead, Domain ANALYST, Domain Lead will deliberate on the views expressed by the Developer and make a judicious decision before changing the status to “Deferred”(Deferred for Next Release).	

6.2 – Progress	
Actor:	Dev lead
Predecessor(s):	6.1 Deferred For Next Release
Status:	In Progress
Assign To:	Developer
Development Lead will reopen the defect in the next Version/Release of the application for fixation and assign it to the Developer by changing the status to “In Progress”	

7.1 – Review Failed	
Actor:	TestLead
Predecessor(s):	1.6 Verified Defect
Status:	Reopen
Assign To:	Dev Lead

On review of the defect if a gap has been identified then the Test Lead will Reopen the defect by providing all the necessary input/information that gives the cause for Reopening the Defect.

8.1 – Update Information	
Actor:	Domain ANALYST
Predecessor(s):	1.7 Final Review
Status:	Reopen
Assign To:	ALL
Domain ANALYST should have the permission to Reopen the Defect that has been marked as Verified by the Tester.	

A typical defect progresses through several statuses which are described below. The defect process flows and descriptions provided in this document explain when a defect can be moved from one status to another.[6]

Status	Description
Closed	Defect fix has been successfully tested or cancelled.
Fixed	Developer has fixed the defect which is now ready for testers to verify.
New	New defect was entered and assigned, but has not yet been accepted by the developers.
In Progress	The defect has been assigned to a developer and is being worked on to provide a resolution.
On Hold	The defect is determined to be a missing or new requirement and a Change Request is pending
Open	The development team has accepted the defect and it is being worked.
Rejected	The Domain Lead or Development Lead feels that this is not a defect.
Reopen	Defect failed retest or a Test Analyst disagrees with developer’s initial analysis.
Retest	Code has been dropped into the test environment and is now ready for retest.
Verified	Defect has been successfully retested and is ready for closure.
Need More Info	Test Lead provides the necessary information sought by the Development Lead or lends more clarity to the Defect logged by the Tester and Reopens the Defect.
Deferred	Development Lead will reopen the defect in the next Version/Release of the application in case it cannot be fixed in the current Version/Release
CR	The defect status is changed to CR(Change Request) by Domain ANALYST indicating that defect is regarded as an Enhancement, that can be implemented either in the current Release of the application or in the subsequent Releases.

### III. DEFECT SERVICE AGREEMENT

Both Priority and Severity should be made as mandatory fields. [6]

Priority:- It defines how quickly the Defect should be fixed . Timeline is the determining factor for Priority.

Severity:- It defines the complexity in fixation of the Defect and its impact on the code.

Priority 1	Target 12 – 24 hours for resolution
	Escalate after 60 minutes in defect has not been moved to Open status
Priority 2	Target 24 – 48 hours for resolution
	Escalate after 4 hours if defect has not been moved to Open status
Priority 3	Target 48-72 hours for resolution
	Escalate after 24 hours if defect has not been moved to Open status
Priority 4	Resolution target is negotiable.

### IV. TRANSITION RULES

The transition rules are defined as the rules by which the various roles must adhere to when using the fields that are provided in the system. There are certain procedural steps that must occur in sequence and these shall be delineated below.

<b>Test Director Transition Rules – Tester</b>		
<i>From Status</i>	<i>To Status</i>	<i>Explanation</i>
New	Closed	The tester has realized that entered defect is not a not valid defect.
Retest	Verified	The tester has verified that the fix has been implemented successfully.
Retest	Reopen	The fix has failed functional testing and is returned to the development lead for further resolution.

<b>Test Director Transition Rules – Test Lead</b>		
<i>From Status</i>	<i>To Status</i>	<i>Explanation</i>
New	Open	The defect has been submitted by the tester and reviewed by the Test Lead. It can be assigned to the Dev Lead for resolution.
Verified	Reopen	On review of the defect if a gap has been identified then the Test Lead will Reopen the defect by providing all the necessary input/information that gives the cause for Reopening the Defect
Need More Info	Reopen	Test Lead provides the necessary information sought by the Development Lead or lends more clarity to the Defect and Reopens the Defect.

<b>Test Director Transition Rules – Developer</b>		
<i>From Status</i>	<i>To Status</i>	<i>Explanation</i>
In Progress	Fixed	A change has been implemented and unit tested successfully and is ready for the test team to retest pending a review by the development lead and release of the code back to the test team
In Progress	Deferred	The Developer in consultation with Development Lead, Domain Lead, Domain ANALYST feels that a Defect cannot be fixed in the current version of the Software then the Defect status is changed to “Deferred”(Deferred for Next Release) and assigned to Development Lead; Here the Development Lead, Domain ANALYST, Domain Lead will deliberate on the views expressed by the Developer and make a judicious decision before changing the status to “Deferred”(Deferred for Next Release)
In Progress	Reopen	If the Developer needs additional information to fix the defect, the defect status is changed to Reopen and assigned to the Development lead to seek clarifications.
Reopen	Fixed	When a ticket is assigned to the developer for clarifying information regarding the fix implemented, once the ticket is updated, it will be reassigned to the development lead and status changed to Fixed.

<b>Test Director Transition Rules – Development Lead</b>		
<i>From Status</i>	<i>To Status</i>	<i>Explanation</i>
Open	In Progress	The defect has been accepted and assigned to a developer for technical resolution.
Open	Need More Info	In case the Development Lead is unable to understand the Defect or thinks consultation is required with the Testing Team(Test Lead/Tester) before assigning it to Developer then “Need More Info” status is assigned by the Development Lead.
Open	On Hold	The defect has been identified as a missing or new requirement that requires a Change Request (CR). The CR process will be followed with the status being changed to “On Hold” and assigned to the Domain Lead.
Reopen	In Progress	The defect has been accepted and assigned to a developer for resolution



Reopen	Verified	A defect that required additional information, requested by the Developer, will be set to Reopen status. The Development Lead will verify the clarity of the queries or the additional information sought by the Developer and then return the defect to the Test Lead with an updated status of Verified.
Reopen	Rejected	The development lead has identified that the issue is not a defect and submits it to be closed without change. There should be a mandatory “Rejection Cause” field or some “Comment” field for the Development Lead to explain the cause for Rejection. Rejection could be because it is a “Duplicate” defect, “Invalid” defect, “Cannot be Fixed”, due to Browser problems/ OS problems/ Software problems, etc, “Unable to Reproduce” due to insufficient data or incorrect steps . The status would be changed to Rejected and reassigned to the Domain ANALYST for closure.
Fixed	Retest	The defect has been successfully fixed and unit tested. The development lead reviews the defect information and verifies the code deployment prior to reassigning the defect to the test team for validation in functional testing.
Fixed	Reopen	A defect is reviewed by the Development Lead and information is determined to be missing from the ticket or the Fix has not been done, it will be changed to status Reopen with clarifying questions and reassigned to the developer
Reopen	On Hold	If a defect in Reopen status is determined to be a change request, it may be placed on hold with the CR process initiated
Deferred	In Progress	Development Lead will reopen the Defect in the next Version/Release of the application for fixation and assign it to the Developer by changing the status to “In Progress”

<b>Test Director Transition Rules – Domain Lead</b>		
<i>From Status</i>	<i>To Status</i>	<i>Explanation</i>
On Hold	Rejected	Once a CR is created and approved with signatures from the Domain ANALYST, the defect is assigned to the Domain ANALYST with comments indicating the approved CR#.
On Hold	Reopen	A defect that is changed to On Hold status for a CR may be determined to be a valid defect and returned to the Development Lead with a status “reopen” for resolution

<b>Test Director Transition Rules – Domain ANALYST</b>		
<i>From Status</i>	<i>To Status</i>	<i>Explanation</i>
Verified	Closed	After a review for completeness by the Test Lead, the defect has been reviewed by the Domain ANALYST and agreed that it can be closed.
Verified	Reopen	Domain ANALYST should also have the permission to Reopen the Defect that has been marked as Verified by the Tester.

Rejected	Reopen	If the Domain ANALYST does not agree with the assessment that a defect is invalid, it may be changed to status “Reopen” and reassigned to the Development Lead with clear reason for the return.
Rejected	Closed	The defect was identified as out of scope or data related. Upon agreement from the Domain ANALYST, the defect would be closed with documentation.
Rejected	CR	The defect status is changed to CR (Change Request) by Domain ANALYST with approved CR# in the comments. This status indicates that Defect is regarded to be an Enhancement, that can be implemented either in the current Release of the application or in the subsequent Releases. This would also give the number of Defects that have received the Status of CR, indicating the minor/major features that could not be envisaged in the Requirements document (BRD, HLD, etc).

## V. LIMITATIONS

The scope of the defect management process is limited to the creation and management of defects by resources involved in the testing process. This paper identifies the roles and responsibilities of these resources as related to the defect management process. This process is not intended to describe the testing workflow. It is focused specifically on the management of testing defects logged in Test Director. The defects identified during the testing process are documented, assigned, tracked, and resolved on a regular basis in Test Director [5].

## VI. CONCLUSION

A software project may include thousand of defects that are found by different people at different stages of the project. Often the person who fixes a defect is different than the people who finds or report the defect. In such scenario, defect reporting and closing cannot be done informally. The use of informal mechanism may lead to defect not getting removed or extra effort in finding the defect again. Hence, defect found must be properly logged in a system and their closure tracked. Defect logging and tracking is considered one of the best practices for managing a project [7].

## ACKNOWLEDGMENT

The authors would like to thanks to the college management for the research support.

## REFERENCES

- [1] J. Marko and M. Aki, “Implementing a software problem management model, a case study,” 2006, Springer Link
- [2] A. A. Shenvi, “Defect prevention with orthogonal defect classification,” ISEC’09, February 23-26, ACM, India, 2009.
- [3] B. Robinson, P. Francis, and F. Ekdahl “A defect-driven process for software quality improvement,” USA: New York, ACM, 2008.
- [4] A. Andrews, P. Runeson, C. Andersson, T. Thelin, and T. Berling, “What do we know about defect detection methods?” *IEEE Software*, May/June 2006.
- [5] J. Marko and M. Aki, “Implementing a software problem management model, a case study,” 2006, Springer Link.
- [6] R. S. Pressman, *Software Engineering: A Practitioner’s Approach* (7th ed.). New York: McGraw-Hill International, 2009.
- [7] Pankaj Jalote , “A Concise Introduction to Software Engineering” Springer, 2008.