

# Defect Migration

Sankara Narayanan R  
Oracle India Pvt. Ltd.  
4<sup>th</sup> Floor, Cyber Towers  
HITECH City, Madhapur  
Hyderabad

Tel: +91-40-3110222/3/4  
Fax: +91-40-3110225  
e-Mail: narayananss@yahoo.co.uk

## Abstract

Testing Science states that it is impossible to achieve 100% test coverage in testing software. Based on the risks in a project, the desired amount of test coverage is arrived at and established. Few areas of the software are left unexercised with defects uncovered and practically it happens. It is equally impossible to uncover all the defects early in the Software Development Life Cycle. For example, Performance defects remains uncovered until the software is integrated and moved to the Performance Testing Lab. This article puts forth a metrics that depicts the status of Defect Migration.

## **Objective of Testing**

Integration Testing uncovers some defects that ideally should have been uncovered in Unit Testing itself. System Testing uncovers some defects that ideally should have been uncovered in Integration Testing. Extrapolated, Customers unearth defects that ideally should have been uncovered by the Software Development Organization. Testing aims at finding defects in the product before the customer identifies it and make it defect-free. A Migrated Defect is defined as a defect that ideally should have been found in any of the previous testing phases. Stopping the Defect Migration is one of the key objectives of any Testing Activity

The product is excellent. There is always a scope for improvement. The product undergoes changes and accommodates enhancements. This makes Regression Testing inevitable. Foreseeing the future becomes inevitable to stop defect injection and it is one of the key objectives of any Testing Activity.

## **Why Defect Migration happens**

Testers are human. The errors that human psychology permits apply here as well. This natural phenomenon needs to be overseen. Scientific principles indeed teach the relevant testing methodologies to minimize the risk of this human psychology inhibiting the defect migration but not to mitigate it.

It is practically impossible to achieve 100% test coverage. This leaves the tester with the fact that some customer in some corner of this world is going to execute a test path that is not covered in (his/her) testing.

Software Developers write code that works adequately, considering the additional work needed there as an 'enhancement'. They defer doing the same attributing it to other work pressure. They often provide justification to the top management in unusual ways and this eventually results in the migration of the defects.

Reviews are the best way to uncover defects early in SDLC, though it is costly. Project Manager failing to justify the review that needs to be scheduled allows the defects to get migrated.

## Defect Migration Accounting

Measuring a Quality Attribute and taking necessary steps to enhance their quality ensures Continuous Improvement in the quality of the Attribute. Defect Migration Index (DMI) measures Defect Migration from one phase to another.

$$\text{DMI of Testing Phase Y to Phase X} = \frac{\text{Number of Migrated Defects from Phase X}}{\text{Number of Defects found in Testing Phase Y}}$$

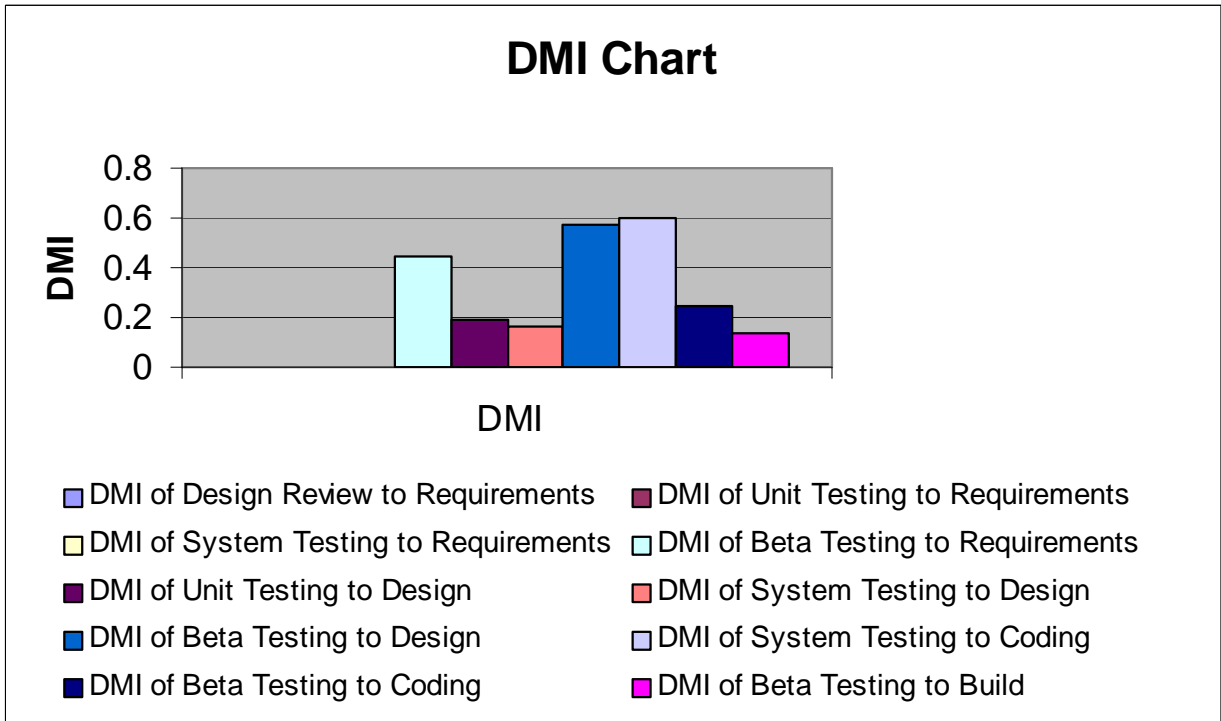
DMI accounts for the migrated defects to a testing phase. In an ideal environment, it should be zero.

Consider the following Defect Distribution – An Example

Testing Phase	Defects Injection Stage			
	Requirements	Design	Coding	Build
<b>Requirements Review</b>	15	-	-	-
<b>Design Review</b>	0	76	-	-
<b>Unit Testing</b>	0	23	98	-
<b>System Testing</b>	0	9	34	14
<b>Beta Testing</b>	2	25	11	6

The below is the DMI Chart for the above distribution.

DMI of	Requirements	Design	Coding	Build
<b>Requirements Review</b>	-	-	-	-
<b>Design Review</b>	0/(0+76) = 0.00 (0%)	-	-	-
<b>Unit Testing</b>	0/(0+23+98) = 0.00 (0%)	23/(0+23+98) = 0.19 (19%)	-	-
<b>System Testing</b>	0/(0+9+34+14) = 0.00 (0%)	9/(0+9+34+14) = 0.16 (16%)	34/(0+9+34+14) =0.60 (60%)	-
<b>Beta Testing</b>	2/(2+25+11+6) = 0.45 (45%)	25/(2+25+11+6) = 0.57 (57%)	11/(2+25+11+6) =0.25 (25%)	6/(2+25+11+6) =0.14 (14%)



### Significance of DMI

DMI serves as a metric that shows the status of effectiveness of the testing in stopping the Defect Migration. Unlike Defect Removal Efficiency, this accounts for defects migrated from any Phase X to Phase Y.

DMI chart shall occupy a section in the Project Closure Analysis Report. The Top Management shall use DMI as the measure of the performance of the testing activity. In the above example, DMI of System Testing to Coding = 0.60. This indicates that Unit Testing is not convincing and needs improvement.

## Acknowledgements

I take this opportunity to thank Sudha Pichumani, Tanushree Parial and Neeraja Thyagarajan for reviewing this article. The time and effort they have taken to review this article and making it more comprehensive is highly appreciated.

## About the Author

Sankara has nearly three years of industrial experience in software testing specializing in the area of Software Testing Tools and Test Automation. This includes the development of a test execution tool and its successful introduction into general use within the Product Team.

Since joining Planetasia.com Ltd., in June 2000, Sankara has defined the Organization-wide Testing Processes and successfully imbibed Exploratory Testing into it. Apart from introducing new dimension to the work in his organization, Sankara has published papers in [www.stickyminds.com](http://www.stickyminds.com).

Since joining Oracle India Pvt. Ltd., in December 2001, Sankara is responsible for the Quality Assurance and Test Automation needs of a Product Testing Team.