WHITE PAPER

Improving the Defect Life Cycle Management Process

Defect State Transition Analysis

Abstract
This document discusses the various metrics used in analyzing the Defect Life Cycle by using the defect statuses to describe the defect state transitions

Mr. Oded A. Tankus, M.Sc.
ODED.TANKUS@ASSURE.NET
# White Paper – The Quality Index

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>3</td>
</tr>
<tr>
<td>DEFECT LIFE CYCLE ANALYTICS</td>
<td>4</td>
</tr>
<tr>
<td>MAPPING THE DEFECT LIFE CYCLE</td>
<td>5</td>
</tr>
<tr>
<td>STANDARDIZING THE DEFECT LIFE CYCLE</td>
<td>8</td>
</tr>
<tr>
<td>DEFECT TRANSITIONS</td>
<td>10</td>
</tr>
<tr>
<td>CYCLES OF INEFFICIENCIES</td>
<td>12</td>
</tr>
<tr>
<td>START TO END PATHS</td>
<td>16</td>
</tr>
<tr>
<td>APPENDIX A – DEFINITIONS</td>
<td>19</td>
</tr>
</tbody>
</table>

© 2019 Assure, Mr. Oded A. Tankus  
www.assure.net
1 Introduction
This document will discuss the analytics process as it supports Evidence Based Management of the Defect Life Cycle.

Analysis of the Defect Life Cycle Management Process will allow to focus on:
1. Highlighting weak links
2. Improving the process
3. Ultimately producing a higher quality SW product

Added benefits of the analysis may include:
1. Reducing costs and resource allocations
2. Reducing inefficiencies resulting in efficient streamlining of defect treatments
3. Increasing overall SW quality
4. Decreasing release delays and time to market
5. Decreasing defect treatment time.

In order to be successful in the efficient analysis of the Defect Life Cycle, there needs to be two major activities:
1. Data Management
   a. Mapping the Defect Life Cycle
   b. Standardizing the Defect Life Cycle across multiple projects
2. Data Analysis
   a. Using agreed upon metrics (both In progress and End Of Release) to visualize and analyze the Defect Life Cycle
   b. Identifying Defect Life Cycle hotspots and bottlenecks, thereby understanding the implications and suggesting plausible solutions
   c. Creating a Defect Life Cycle knowledge base for the learning organization

The basic assumptions in this type of analysis are as follows:
1. **Defects identification and characterization** must be defined, standardized and communicated. For the purposes of this analysis, a most important characteristic of a defect is its current status and current severity level

2. **Defect treatment** is recognized to be time sensitive, so a defined timeline must be defined and adhered to for various types of defects

3. The **Defect Life Cycle** is well defined by two main characteristics: Status and Assignment Teams (test, dev). This must be clearly communicated to all relevant staff so they may understand their role in treating defects.

In meeting the above three assumptions and using the proposed metrics and algorithms, it will be very simple to identify various issues, propose immediate solution and ultimately reap the above stated benefits.
2 Defect Life Cycle Analytics

There are four major activities when analyzing the Defect Life Cycle:

![Figure 1: Analytics Process of the Defect Life Cycle](image)

<table>
<thead>
<tr>
<th>Activities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map &amp; Standardize</td>
<td>The purpose of this activity is to define the Defect Life Cycle if not defined. If defined, the Defect Life Cycle should be reviewed, simplified and standardized so it may be applied across the release landscape.</td>
</tr>
</tbody>
</table>
| Identify & Define           | The purpose of this activity is to identify three types of transitions:  
   1. All status transitions undergone by a defect (legal and illegal alike) throughout its travels within the Defect Life Cycle  
   2. Cycles of inefficiencies – loops within a defect is trapped and may escape after one or more rounds  
   3. Start to End Paths – all possible combinations of start and end points Also in this activity, thresholds are defined to reflect corporate policy and as to acceptable process behaviour. |
| Analyses                    | For each of the transitions, the following are calculated:  
   1. Defect Distributions (counts)  
   2. Median | Mean treatment times  
   3. Gaps between treatment times and thresholds |
| Improve                     | The purpose of this activity is to:  
   1. Identify the characteristics of each of the transition types and ultimately the overall process  
   2. Recognize root causes of extreme treatment time values in order to identify potential improvements within the process  
   3. Plan and implement improvements |
3 Mapping the Defect Life Cycle

Let’s assume a naïve Defect Life Cycle.

![Naïve Defect Life Cycle Diagram]

**Figure 2:** Naïve Defect Life Cycle

The process is as follows:

1. On the top left hand side in the above figure, we can see the testing process producing potential defects.

2. These potential defects are tagged with a status of ‘Detected | New’.

3. After a triage of the potential defects, they will be tagged with one of the following statuses:
   - ‘Enhancement’ – this is not a defect but an enhancement to be delivered in the future
   - ‘Deferred’ – the defect fix is postponed to a future release and not within the scope of the current release
   - ‘Open’ – a recognized actual defect

4. Once identified as an actual defect, the defect will be tagged with one of the following statuses:
   - Withdrawn – the testing team has withdrawn the defect for one reason or another – End of Treatment
   - Rejected – the defects has been analyzed by the development team and has been rejected as a defect – End of Treatment
   - In Fix – The development team assigned to treat the defect is in the process of fixing the defect

5. After the development team assigned to treat the defect has completed the fix, the defects is tagged with a status of Fixed.

6. When fixed, the defect is required to undergo testing by the Testing Team. The defect is tagged with the status ‘Re-Test’.

7. Having completed the re-test, the defect is tagged with the following statuses:
   - Closed - The defects has either been fixed and is tagged with a status of ‘Closed’ – End of Treatment
White Paper – The Quality Index

- Failed Re Test - The defect has failed test and is tagged with a status of ‘Failed Re-Test’ and is transferred back to the Development Team to be fixed again, where it will be tagged with a status of ‘In Fix’ as work is initiated.

8. Having been closed, a defect may be encountered again. If so, the defect is tagged with a status of ‘Re-Open’ and is transferred to the Development Team to be fixed again, where it will be tagged with the status ‘In Fix’.

Not many organizations define their Defect Life Cycle at the level of simplicity as described above. Figure 3 shows a more realistic Defect Life Cycle:

Figure 3: Defined Defect Life Cycle

As we can see, there are two levels of statuses: Primary and Secondary statuses. Moreover, there are many more legal transitions of defects from one status to another.

We assume that as the level of complexity increases, so does the management of the Defect Life Cycle, ultimately leading to illegal transitions, e.g., From ‘New’ to ‘Closed’ without showing the paths the defect has traversed. This is mostly due to over mapping and undue process administration, causing employees to be creative in finding ‘shortcuts’ to the documentation bureaucracy.
Although Figure 3 above represents the defined legal transitions of defects, there may be many more paths that are used illegally (see Figure 4 below). In comparing the two, we can see how much more complicated is the Illegal Defect Life Cycle.

Figure 4: Actual Defect Life Cycle including illegal transitions (taken from one of the largest banks in the USA)

The major ‘take aways’ are:

1. Do not define/plan your Defect Life Cycle to cover all possibilities.
2. Do plan the Defect Life Cycle to meet your monitoring and reporting requirements.  
   For example: There is no need for a ‘Closed’ status for each of the ‘Rejected’ statuses (see Figure 3 above).
3. Make sure that the Defect Life Cycle is communicated and adhered to
4. Make sure that the tools used for managing defects, e.g.: MF ALM, JIRA, etc…, do not allow for illegal transitions between statuses
4 Standardizing the Defect Life Cycle
There are many issues to consider in standardizing the Defect Life Cycle. The major ones are:

<table>
<thead>
<tr>
<th>#</th>
<th>Issue</th>
<th>Considerations</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Large number of defined statuses across the organization (projects)</td>
<td>Define minimal reporting requirements that are standard across the board</td>
<td>Regulatory requirements</td>
</tr>
<tr>
<td></td>
<td>Reporting is complex</td>
<td></td>
<td>Quality requirements</td>
</tr>
<tr>
<td>2</td>
<td>Collecting different defect statuses from multiple projects</td>
<td>Are all these “flavors” really necessary? Why?</td>
<td>Organizational requirements</td>
</tr>
<tr>
<td></td>
<td>Dealing with various flavors of the Defect Life Cycle process across projects</td>
<td>How can we standardize?</td>
<td>Management requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>What is the information required to manage DLC process more effectively and efficiently?</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Organizational inefficiencies across projects</td>
<td>What improvements are necessary</td>
<td>Quality requirements</td>
</tr>
<tr>
<td></td>
<td>Wasted effort and resource to manage flavors of the Defect Life Cycle process</td>
<td>How to better engineer the process so costs and resources are kept to a minimum while maintaining a high level of quality product?</td>
<td></td>
</tr>
</tbody>
</table>

The following table suggests ways to efficiently standardize the Defect Life Cycle:

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1 | Analyze the current set of statuses                                  | Collect all the defect statuses from all projects in scope  
Analyze list and create unique values |
| 2 | Analyze defect status usage level                                    | Make sure that all the statuses are relevant and used  
Delete obsolete values |
| 3 | Create and Analyze the Defect Network Diagram (State Transition Diagram) | Map the defect statuses onto the Defect Life Cycle process, thereby defining legal transitions from status to status |
| 4 | Verify that defects statuses meet “good enough” criteria            | 1. Statuses are exclusive and do not overlap  
2. Status timing changes are correct, i.e., defects should clearly always belong to just ONE status at a time  
3. Statuses are defined and easily understood  
4. Statuses DO NOT try to cover other “fields of purpose” i.e., priority, severity |
| 5 | Identify minimum reporting requirements                              | What is relevant for all projects?  
How to support individual project needs, e.g.: projects dealing only with production defect might use a set of statuses that are NOT visible for other projects. |
| 6 | Standardize                                                           | Standardization is technically easy – the hard part is to decide on:  
1. A new set of standardized statuses that is agreed upon by everyone  
2. Which old statuses should be mapped onto the new standardized statuses |
The last step in this section is to consolidate the various statuses and the legal transitions. In doing so, we get a well-defined Defect Life Cycle to be communicated to all relevant staff.

**Figure 5:** Well Defined Defect Life Cycle

The major take-away for this section is standardization. Without it, there can be no basis for identifying transitions and scaling up to enterprise analysis of the Defect Life Cycle.

The Defect Life Cycle in Figure 5 will be the one used in this document going forward.
5 Defect Transitions

Defect Transitions

The first action that is required is to map the ‘From’-‘To’ Statuses within the Defect Life Cycle map, i.e., identify the unique ‘From’-‘To’ status couples that form the basis of the transitions.

Once identified, we count the number of defects for each status couple.

Mapping the defect counts on the ‘From’-‘To’ status couples will produce a 3-D frequency map and two marginal distribution charts (Figures 6a-c below):

The marginal distributions of the ‘From’ statuses and of the ‘To’ statuses (Figures 6b and 6c respectively) show the frequency of defects as they exit a status and as they enter the subsequent status respectively.

This metric is most useful in identifying bottle necks in the transition over the network (see 1 and 2 within Figure 6a), specifically when focusing on Critical and High severity level defects. By doing so, we can identify inefficiencies at specific points in the Defect Life Cycle process. Implications of this analysis is that we have the ability to identify inefficiencies at specific points in the process.
Alternatively, instead of counts, we can calculate the mean treatment time within each ‘From’ - ‘To’ status couple, where treatment time is defined as the time entering the ‘From’ status to the time exiting the ‘To’ status. This will indicate which part of the process are congested, and may require improvement.
6 Cycles of Inefficiencies

Cycles of Inefficiencies are ‘loops’ within the Defect Life Cycle in which defects get ‘trapped. A defect may go around many times until it escapes the loop, so identifying the loops, counting the number of defects within each loop and analyzing the time taken to escape the cycle is of paramount importance.

The following shows the varying Cycles of Inefficiencies

![Diagram of 2nd Order Cycles](image1)

**Example:**
- New(1)–Assigned to Resolver(2)–New(1)
- Assessment(5)–Rejected(11)–Assessment(5)

![Diagram of 3rd Order Cycles](image2)

**Example:**
- New(1)–Assessment(5)–Assigned to Resolver(2)–New(1)
- Ready to Deploy(4)–Test Delayed(7)–Ready for Re Test(6)–Ready to Deploy(4)

![Diagram of 4th Order Cycles](image3)

**Example:**
- Assessment(5)–Fix In Progress(3)–Ready for Re Test(6)–Failed(9)–Assessment(5)

![Diagram of 5th Order Cycles](image4)

**Example:**
- Assigned to Resolver(2)–Fix In Progress(3)–Ready to Deploy(4) – Ready for Re Test(6) – Failed(9)–Assigned to Resolver(2)
There are several analyses that may be done on cycles of inefficiencies:

The graph below displays the defect frequency (bars) by Cycles of Inefficiencies. For each of the cycles, the average time in the cycle is displayed (line).

**Figure 8:** Defect Frequency by Cycles of Inefficiencies

The analysis of the above in Figure 8 helps in identifying what are the most common cycles of inefficiencies. We are then able to identify the defect distribution (average # times that a defect is trapped in the cycle of inefficiency and the average time the defect stays there.

Several key points in the analysis do show:

1. Although the ‘Open’-‘Rejected’-‘Open’ cycle has a high frequency of defects, the average time a defect is in the cycle is low (see #1 within Figure 8)
White Paper – The Quality Index

2. Although the ‘Open’-‘Fixed’-‘Rejected’-‘Open’ cycle, has a low frequency of defects, the average time a defect in in the cycle is very high (see #2 within Figure 8).

These are two examples focusing on the extreme: low counts with long time in cycle, and high counts with small time in cycle.

The implications of this analysis is that it points to:

1. Faulty reasoning in the defect analysis
2. Disagreements between teams (e.g.: ‘ping ponging’ between Test and Dev teams that do not agree if this is a real defect or if it should be rejected).
3. Areas in need of process improvement

This type of analysis will also help to:

1. Estimate treatment time of defects at various severity levels
2. Identify the causes of a trapped defect

When focusing on one specific cycle, we delve deeper into the behaviour of defects within the cycle. In Figure 9 below, we choose one Cycle of Inefficiency and analyse the frequency of defects by the # rounds that a defect executed within the cycle until it escapes.

Figure 9: Defect frequency and length of time by rounds within a cycle
In analyzing the data displayed in Figure 9, we see that:

1. Two defects have gone around the cycle six times. As such, we would like to understand the reason for that…..e.g., a possible blocker between the Test and Development team positionings (see #1 within Figure 9).

2. Defects that were trapped for only one round passed with an average time of 14.3 days (see #2 within Figure 8)

3. Nine defects have cycled four times around this Cycle Of Inefficiency with an average time of 32.8 days (see #3 within Figure 8)….something to review
7 Start to End Paths

Start to End (S2E) transitions are defined by two end points of the defect transition, i.e., the initial status and final status of the defect life cycle history...what happens in between these statuses is not relevant in this analysis.

Figure 10: Start to End Paths

The identified S2E paths in Figure 10 are:

- New – Rejected
- New – Closed
- Assigned to Resolver – Rejected
- Assigned to Resolver – Resolved
- Fix In Progress – Rejected
- Fix In Progress – Resolved
- Ready to Deploy – Rejected
- Ready to Deploy – Resolved

The first question that needs to be asked is: Why is it that defects have an initial status of anything other than New?
As in the other analyses above, we will analyze the defect frequency and the mean treatment time for each of the S2E paths.

Figure 11 displays the defect frequency and mean time for each of the S2E paths.

![Figure 11: Defect frequency and length of time within Start to End Paths](image)

For each of the S2E paths we assign two values – the defect frequency and the average time taken to traverse the path.

This type of analysis helps us identify the most common paths and the average time on the path.

In analyzing the results displays in Figure 11 we see the following:

1. The defect counts in the paths ‘New’-‘Rejected’ and ‘New’-‘Resolved’ are almost the same, and both paths have the same low average time to traverse (see #1 within Figure 11).
2. The path ‘Assigned to Resolver’-‘Rejected’ has a relatively low defect count, but the highest average day treatment time (see #2 within Figure 11).

The implication of this analysis is that:

1. Helps in estimating treatment time of defects at various severity levels
2. Points to areas in need of process improvement
When choosing one S2E path, we can analyze the frequency of defects by treatment time.

The most interesting thing in analyzing the data in Figure 12, is the defect outliers, i.e., analyzing the long tail to the right showing the amount of defects that take an extraordinary amount of time to fix. This may be due to defects that have been forgotten and not closed, defects that are being deferred from release to release, or difficulty in fixing the defect.

In analyzing the results displayed in Figure 12, we see that:

1. As expected, the majority of defects travel fairly quickly within the S2E path (see #1 within Figure 12).
2. There are defects that take an overly long time to traverse the path, e.g.: there are over 100 defects that took over 200 days to complete the selected S2E path (see #2 within Figure 12).
3. The black and red vertical dotted lines show the average (red) and median (black) treatment time. Note that the greater the skew (longer ‘tail to the right’), the importance of the average treatment days will decrease and the median treatment days increase (see #3 within Figure 12).

The implication of this type of analysis is that it focuses on defects that require immediate attention – especially critical and high defects with a long treatment time.
Appendix A – Definitions

Analytics
Analytics is the discovery, interpretation, and communication of meaningful patterns in data.
- **Discover** – use of mathematical and/or statistical techniques
- **Interpretation** – Discovering new meaningful insights and/or confirm existing ones
- **Communications** – visualizations through graphs, reports etc.

Analytics is used in support of management decision making and risk mitigation processes by describing, predicting and prescribing effective alternatives

Evidence Based Management
Making decisions through the conscientious, explicit and judicious use of the best available data (evidence) from multiple sources by...
- **Asking** Translating a practical issue or problem into an answerable question
- **Acquiring** Systematically searching and retrieving the data
- **Appraising** Critically judging the trustworthiness and relevance of the data
- **Aggregating** Weighing and consolidating the data
- **Applying** Incorporating the data in the decision-making process
- **Assessing** Evaluating the outcome of the decision taken

...to increase the likelihood of a favorable outcome.

The 5 principles of Evidence Based Management
- **Face the hard facts**
  Build a culture in which people are encouraged to tell the truth

- **Be committed to “fact based” decision making**
  Being committed to getting the best evidence and using it to guide actions

- **Treat your organization as an unfinished prototype**
  Encourage experimentation and learning by doing

- **Look for the risks and drawbacks in what people recommend**
  Even the best medicine has side effects

- **Avoid basing decisions on:**
  ...untested but strongly held beliefs, what you have done in the past, or uncritical “benchmarking” of what winners do
Life Cycle

There are many definitions of the term “Life Cycle”. The following are two of the most relevant for the purposes of this document:

“The total phases through which an item passes from the time it is initially created until the time it is either consumed in use or disposed of as being excess to all known material requirements.”

Dictionary of Military and Associated Terms. US Department of Defense 2005...modified

The series of changes that happen to an object over the course of its lifetime, e.g.: a defect traversing its various legal statuses: New → Open → In Fix → Fixed → Tested → Closed

YourDictionary definition and usage example...modified