Looking Beyond Quantitative Defect Management

Anil Midha
BAE Systems, NES
Wayne, NJ
Presentation Agenda

• Background
• Issues involved
• Where do we need to look and why
• Recommendation – “The Three-Prong Approach“
• Defect insertion vs. detection analysis
• An integrated approach to defect analysis
• A robust causal system
• Summary
• In closing
Background

- Organizations aspiring to be (or operating at) high CMMI maturity levels generally focus on defects

- Typically they collect defects from work product inspections or reviews in development phases

- For CMMI Maturity Levels 4 and 5, it is required to quantitatively manage and statistically analyze the defects to:
  - Understand the impact of common as well as special causes of variations
  - Perform root cause analysis of high impact defects
Issue is ...

Quantitative defect analysis most often focuses on “quantity” of defects, not other aspects of defect analysis, such as:

- Defect prevention throughout the development lifecycle
- How early are the defects detected after getting inserted?
- What is the “chasm” between defect insertion and defect detection?
- How do we reduce this “chasm” and thus cause left-shift in phase detection of defects?
- What is the cost-effective approach to impact both quantity of defects and early detection of defects?
Need to:

• Have a defect management approach that complements quantitative analysis
• Analyze defect insertion, detection, and correction process
• Consider other significant defects beyond the software and systems defects detected through peer reviews and inspections
• Review defects from more of an integrated engineering view rather than a single functional discipline view
• Examine the effectiveness of the causal system and apply it to a broader range of anomalies and opportunities than just defects
Why is it important?

• Quantitative management and application of statistical control techniques for defect analysis is “necessary” but not “sufficient”

• Defects getting detected at later development lifecycle phases do not receive the scrutiny of the quantitative management

• A non-integrated approach of defect management misses out on some of the key opportunities of addressing defect prevention and early detection in the most cost-effective manner

• Lacking a robust causal system leads to treating the symptoms rather than addressing the causes of deeper issues
The Answer is …

A Three-Prong Approach:

- Reduce the gap between phase of defect insertion and defect detection
- Adopt an engineering integrated approach to defect analysis
- Apply a robust causal system
Engineering Development Lifecycle

- Engineering development lifecycle can be considered as a series of 1 to n Design-Development (DD) activities followed by a corresponding Verification and/or Validation (V & V) activities

- Assumptions:
  - One or more functional disciplines (Software, Systems, Electrical Hardware, Mechanical, etc.) are working in parallel in various phases
  - Every functional discipline may or may not be performing a corresponding V & V activity after its DD activity
  - Parts of the development lifecycle may be repeated incrementally and/or iteratively
Engineering Development – Defects Injection & Removal

Injected Defects in Various DD Activities

Requirements Development & Corresponding V & V Activities

Design & Corresponding V & V activities

Build & Corresponding V & V activities

Integration & Corresponding V & V activities

Escaping Defects

Defects Removed Through V & V Activities
Conceptual Defect Insertion/Detection Model

Defects Inserted in Requirements Phase
\( \alpha_R \)

Total Remaining Defects
\( \gamma_R \)

Defects Detected in Requirements Phase
\( \beta_R \)

Defects Inserted in Design Phase
\( \alpha_D \)

Total Remaining Defects
\( \gamma_D \)

Defects Detected in Design Phase
\( \beta_D \)

Defects Inserted in Build Phase
\( \alpha_C \)

Total Remaining Defects
\( \gamma_C \)

Defects Detected in Coding Phase
\( \beta_C \)

\( \alpha_R \geq \beta_R \)

Total Defects in Design Phase, \( T = \gamma_R + \alpha_D \) and so on ...

\( \beta_D < T \)
Defect Insertion vs. Detection (1)

- A defect detected in k\textsuperscript{th} V & V activity might not have been inserted in the corresponding preceding kth DD activity
- Usually it is an earlier DD activity of the same or a different functional discipline in which the defect got inserted
- In some extreme cases, the defect might have been inserted in an earlier DD activity of a different spiral/iteration
Defect Insertion vs. Detection (2)

- **Ideal Case:** A defect inserted in the $k^{th}$ DD activity gets detected in the $k^{th}$ V & V activity.
- **Typical Case:** A defect inserted in the $k^{th}$ DD activity gets detected in the $i^{th}$ V & V activity, where $i^{th}$ activity is an earlier DD activity in the time sequence.
- The gap, $G_{ik}$ is the number of intervening V & V activities between the $i^{th}$ and the $k^{th}$ V & V activities.
- One would like the gap, $G_{ik}$ to be zero.
Defect Insertion vs. Detection (3)

- Practically, it may not be possible to get this ideal state because it may be one of the intervening V & V activities (like a simulation) that might have been the only practical first V & V activity to detect the defect.
- Analyze which V & V activity “should have” detected the defect; let us assume it is jth V & V activity.

The Gap, $G_{jk}$

- It is this real gap, $G_{jk}$ – let us call “Opportunity Gap”, that must be reduced.
Defect Insertion vs. Detection Analysis (1)

- For each significant defect in each V & V activity, we need to collect and analyze:
  a) Sequence number of the DD activity where the defect got inserted
  b) Sequence number of the V &V activity where the defect got detected
  c) Sequence number of the V &V activity where the defect should have been detected (by default, it should be same as for defect inserted)
- Calculate Opportunity Gap (OG) = difference of sequence numbers between (b) and (c)
- Best Case: OG being zero, i.e., no V &V activity missed detecting the defect
- Typical Case: OG being greater than zero; i.e., 1 or more V &V activities missed detecting the defect
Defect Insertion vs. Detection Analysis (2)

• Goal: Reduce OG for all significant defects

• Further analyze the data for:
  – Which V & V activities are able to detect more defects?
  – Which DD activities are more error prone in inserting defects
  – Which VV activities are more prone to missing defect detection?

• This analysis should lead to strengthening:
  – Those V & V activities that are missing defect detection
  – Those DD activities that are prone to defect insertion

• Over time – with appropriate process adjustments – OG should be reducing
Benefits of The Approach

• By paying close attention to defect insertion and detection, the process changes will be applied where it would be most needed

• Over time, most of the defects will tend to be detected in the earliest possible detection opportunity

• Over time, most of the defect detection will have a “left shift” effect

• This will lead to the most cost effective DD and V&V activities, also impacting product development cycle time and quality
Need for an Integrated Approach to Defect Analysis

• Defects detected in the later development lifecycle phases are usually more complex – they impact most disciplines, are the most expensive to fix, and require broader & deeper scrutiny

• Often superficial analysis leads to categorizing the detected defect in one or the other discipline, while it may have been best addressed in a multi-discipline approach for the most optimal solution

• Addressing only a subset of root causes of the problem may lead to a partial or sub-optimal solution
An Integrated Approach to Defect Analysis

- Record all integration and systems test defects in a cross-discipline engineering defect tracking system
- Review the detected defects in a multi-discipline team to:
  - Analyze all possible causes of the problem
  - Assess the impact in all subsystems/components in all functional disciplines
  - Identify an optimal near-term solution while simultaneously analyzing if there is a better longer-term solution for later implementation
- Implement the identified solution using a systems approach
- Address root causes of the defect to implement preventative actions for the future
Need for a Robust Causal System

• Inadequate root cause analysis may lead to:
  – Treating the symptoms rather than the problem
  – Addressing the wrong problem

• A robust causal system would help uncover real causes of the problems so that actions could be taken to avoid similar problems in the future

• Addressing root causes of the problems is one of the most effective defect prevention mechanisms
A Robust Causal System (1)

Observations

Cause → Problem → Symptoms

Actions

Preventive

Corrective

Mitigating

Improvements

Control

Management

Objectives

Reference: Card, David N. “Understanding Causal Systems” CrossTalk, October 2004
A Robust Causal System (2)

• For each significant defect:
  – Isolate and Identify symptoms and problem
  – Use Ishikawa diagram approach to identify all the root causes of the problem
  – Understand that it may produce multiple symptoms
  – Identify all possible causes that may have contributed to the problem

• Identify appropriate preventive, corrective, and mitigating actions to address causes, problem, and the symptoms
Summary

- The Three-Prong Approach needs to complement, not replace, quantitative defect management and statistical control
- Defects need to be detected in the earliest possible V & V activity
- Adopt an integrated systems approach to address the significant problems identified in the later development lifecycle phases
- Use a robust causal system to analyze significant defects and their root causes
In closing …

Reducing the number of defects is as important as preventing them and detecting them at the earliest opportunity.
Thank You!