**Practical Software and Systems Measurement Continuous Iterative Development**

**Measurement Framework**

**Part 2: Measurement Specifications: Defect Resolution**

Version 2.1

April 15, 2021

|  |  |  |
| --- | --- | --- |
| **Developed and Published by Members of:** | | |
| Practical Software & Systems Measurement | National Defense Industrial Association | International Council on Systems Engineering |
|  | NDIA 100 Year logo |  |
| Product No.  PSM-2021-03-001 |  | Product No.  INCOSE-TP-2020-001-06 |

|  |  |  |
| --- | --- | --- |
| **Editors:** | | |
| **Cheryl L. Jones**  US Army  [cheryl.l.jones128.civ@mail.mil](mailto:cheryl.l.jones128.civ@mail.mil) |  | **Geoff Draper**  L3Harris Technologies  [geoff.draper@l3harris.com](mailto:geoff.draper@l3harris.com) |
| **Bill Golaz**  Lockheed Martin  [willliam.h.golaz@lmco.com](mailto:willliam.h.golaz@lmco.com) |  | **Paul Janusz**  US Army  [paul.e.janusz.civ@mail.mil](mailto:paul.e.janusz.civ@mail.mil) |

Unclassified: Distribution Statement A: Approved for Public Release; Distribution is Unlimited

PSM Product Number: PSM-2020-06-001

INCOSE Product Number: INCOSE-TP-2020-001-06

**Copyright Notice:**

For this document, each of the collaborative organizations listed on the cover page is the sole manager of their products and services and are the only parties authorized to modify them. Since this is a collaborative product, modifications are managed through the participation of all parties.

General Use: Permission to reproduce, use this document or parts thereof, and to prepare derivative works from this document is granted, with attribution to PSM, NDIA, and INCOSE, and the original author(s), provided this copyright notice is included with all reproductions and derivative works.

Supplemental Materials: Additional materials may be added for tailoring or supplemental purposes if the material developed separately is clearly indicated. A courtesy copy of additional materials shall be forwarded to PSM ([psm@psmsc.com](mailto:psm@psmsc.com), attention: Cheryl Jones). The supplemental materials will remain the property of the author(s) and will not be distributed, but will be coordinated with the other collaboration parties.

Author Use: Authors have full rights to use their contributions with credit to the technical source.

Supplemental Notice from INCOSE: This work is an Affiliate Product per INCOSE Policy TEC-107 INCOSE Technical Product Development & Commercialization (26 October 2018). It is a technical product developed outside the INCOSE product development process and was made by INCOSE members in cooperation with PSM and NDIA; then approved by INCOSE to be distributed from INCOSE central channels. The authors own the copyright and take primary responsibility for proper branding, intellectual property, content quality and appropriate citations with INCOSE oversight based on this policy & related procedure.

Contents

[8. Measurement Specifications 1](#_Toc70233510)

[8.7 Defect Resolution (Team or Product Measure) 1](#_Toc70233511)

List of Figures

[Figure 1: Defects Detected versus Resolved 1](#_Toc70233512)

[Figure 2: Cumulative Defects Detected vs. Cumulative Defects Resolved 2](#_Toc70233513)

[Figure 3: Defect Resolution Lag Time 3](#_Toc70233514)

List of Tables

[Table 1: Defect Resolution Lag Time 3](#_Toc70233515)

# Measurement Specifications

## Defect Resolution (Team or Product Measure)

|  |  |
| --- | --- |
| **Measure Introduction** | |
| **Description** | Defect Resolution refers to the process of correcting defects that are detected in the system. It is used in conjunction with the Defect Detection measures to ensure that critical defects are resolved in an efficient manner and do not result in inherent quality problems. |
| **Relevant Terminology** | The terms defects (team errors), iterations, containment, escapes, and releases is defined in Section 3 of Part 1: Defect Terminology. These terms are also used in the measurement specification for Defect Detection. |

|  |  |
| --- | --- |
| **Information Need and Measure Description** | |
| **Information Need** | * When are detected defects resolved? Are high priority defects resolved prior to release? * How many iterations does it take to resolve defects? (aging) * Which defect types have the greatest impact? * Are certain defects taking longer to resolve than others? * How effective was the defect resolution process? |
| **Base Measure 1** | Defects detected, per iteration (integer scale) |
| **Base Measure 2** | Defects resolved, per iteration (integer scale) |
| **Base Measure 3** | Iterations to Resolve (# of iterations between detection and resolution) (integer scale) |
| **Derived Measure 0…n** | Resolved 0...n Iteration = the number of defects that are resolved 0..n iterations after being detected  Note: Defects resolved in iteration 0, are contained defects. |

|  |  |
| --- | --- |
| **Indicator Specification** | |
| **Indicator Description and Sample** | Figure 1: Defects Detected versus Resolved  Figure 1 shows that for Iteration 1, not all the defects discovered in Iteration 1 were resolved. These defects were deferred, put on the product backlog, prioritized, and planned to be resolved in upcoming iterations. For Iterations 2 and 3, more defects were resolved than detected, meaning that defects discovered from previous iterations were resolved, thus reducing the product backlog. The gap between Cumulative Defects Detected and Cumulative Defects Resolved creates mission or technical debt, which is added to the product backlog. |
|  | Figure 2 shows the cumulative number of defects detected and resolved. In Figure 1 and Figure 2, Iteration 6 was planned to address defects vs. adding new features and capabilities.    Figure 2: Cumulative Defects Detected vs. Cumulative Defects Resolved |
|  | An issue that is often evaluated is how long it takes to resolve discovered defects. In a simplistic case, one can look at how many iterations it takes to resolve the defect. This is shown as a simple bar chart in Figure 3 as Defect Resolution Lag Time. In this example, the defects that took 4 and 5 iterations to fix were lower priority defects dealing with minor changes to screen displays and software documentation.    Figure 3: Defect Resolution Lag Time |
|  | Preferably, a defect would be resolved in the same iteration as it was discovered (the teal series of diagonal cells in Table 1 below). All cells to the right of this diagonal represent escaped defects across iterations. Filtering can be applied for the most critical or highest priority defects. Defects that are not resolved after multiple iterations may represent a risk to the inherent quality of the product, may represent an issue with the defect resolution process, or may indicate lower priority defects that have not been prioritized for implementation. Analysis of the Defect Resolution Lag Time measure should focus on the high priority defects and ensure they are being resolved in a timely matter.  Table 1: Defect Resolution Lag Time |
| **Analysis**  **Model** | Figure 1, Defects Detected vs. Defects Resolved, shows the difference/delta between defects discovered and defects resolved, by iteration.  The Cumulative Defects Detected vs. Resolved indicator can be used in conjunction with the Feature or Capability Backlog measure. When checked cumulatively, if the number of defects discovered is greater than the number of defects resolved, the backlog is growing. If the number of defects discovered is less than the number of defects resolved, the backlog is getting smaller. |
| **Decision Criteria** | In Figure 1, for each defect that does not get resolved in the same iteration as it is discovered, the defect and its priority shall be considered during the planning session for the follow-on iteration.  In Figure 2, when the difference/gap between cumulative defects discovered and cumulative defects resolved exceeds 20% of the cumulative defects discovered, the team shall consider having an iteration specifically designed to resolve the outstanding defects.  In Figure 3 and Table 1, defects with Priority 1 and 2 should have a defect resolution lag time not greater than 1 iteration. If not, the defect shall be considered for resolution in the next iteration, with customer approval of this action. Priority 3 through 5 defects may be deferred until later iterations, based on customer priorities.  In Figure 3 and Table 1, most Priority 1 and 2 defects should be resolved prior to release (e.g., a condition of release). Some may be deferred to a later release, with customer agreement. Priority 3 through 5 defects not resolved may be released with customer approval and have a customer approved work around.  Defect detection and resolution data is often presented and used as a criteria for phase completeness at phase gates and associated reviews. |

|  |  |
| --- | --- |
| **Additional Information** | |
| **Additional Analysis Guidance** | Considering the nature of agile development, a defect lower in severity and priority in the product backlog may not be resolved immediately but, be deferred to be resolved in a later iteration. To account for this planned delay, the Defect Resolution Lag Time could be derived from the Iteration the defect was resolved to the Iteration the defect was planned to be resolved (instead of Iteration the defect was detected).  The derived measure for Defect Resolution Lag Time listed above is measured for defects that were resolved. The lag time for open, unresolved defects would be calculated by the Current Iteration less the Iteration the defect was detected.  Digital Engineering/Model Based Engineering should result in early verification and product specification completeness in earlier lifecycle phases accomplished via models and digital system views. A particular emphasis is determining if defects are both detected and resolved in earlier phases than previous performance, and that defects are resolved as early as possible.  More advanced analysis may evaluate (new) defect insertions during defect resolution, or defects resolutions that failed. Recurring rates may be an important customer concern. |
| **Implementation Considerations** | Counting methods need to be defined to determine:   * What constitutes/does not constitute a defect * E.g., peer review findings may be considered errors and not considered internal defects * E.g., an internal error that is sent back to the originating team and results in rework, may be considered a defect * When defects will/will not be counted (e.g., upon hand-off to another team/3rd party) * Internal defects vs. external defects (e.g., defects discovered by the developer, by the customer in an operationally representative environment, or by the customer in operations)   Determining a value for the Iteration the defect was detected and the Iteration the defect was resolved may be tool dependent.  As an alternative view, these measures and indicators may be constructed using only Priority 1-3 defects that affect functional performance.  Some iterations may consist of only defects resolutions. Keep this contextual information in mind when it comes to analyzing the data. |

|  |  |
| --- | --- |
| **Additional Specification Information** | |
| **Information Category** | Product Quality |
| **Measurable Concept** | Functional Correctness |
| **Relevant Entities** | Defects |
| **Attributes** | Iteration Defect was Detected  Iteration Defect was Resolved  Defect Priority |
| **Data Collection Procedure** | Data is collected at the end of each iteration by the team lead from the team tracking tool. |
| **Data Analysis Procedure** | Iteration the defect was detected and Iteration the defect was resolved are discussed during the defect tracking and defect resolution meetings. Data is analyzed at the end of each iteration by the team during the iteration retrospective meeting and considered during the planning session for the follow-on iteration. |