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

**Measurement Framework**

**Part 2: Measurement Specifications: Defect Detection**

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](#_Toc70233357)

[8.6 Defect Detection (Team, Product, or Enterprise Measure) 1](#_Toc70233358)

List of Figures

[Figure 1: Defect Terminology 2](#_Toc70233359)

List of Tables

[Table 1: Defect Detection by Release 2](#_Toc70233360)

[Table 2: Defect Resolution Lag Time 3](#_Toc70233361)

# Measurement Specifications

## Defect Detection (Team, Product, or Enterprise Measure)

|  |  |
| --- | --- |
| **Measure Introduction** | |
| **Description** | Programs strive to deliver products of acceptable quality for use by internal or external customers, and to manage the extent of defects and rework that could inhibit the effective use of these products in operations. Acceptable quality can often be a tradeoff against other attributes, such as speed, cost, and time to market. Quality objectives may vary by application domain and the business goals of the enterprise, but the objective is generally to minimize the quantity of defects detected after release (escaped) or conversely, to maximize the defects detected during development prior to product release (contained). This may be accomplished through defect detection processes such as effective peer reviews, automated testing throughout development, and other verification and testing approaches. |
| **Relevant Terminology** | Defect terminology is defined in Section 2.3 and Section 3 of Part 1.. |

|  |  |
| --- | --- |
| **Information Need and Measure Description** | |
| **Information Need** | How many defects were contained (discovered) prior to internal release?  How many defects were released (escaped) to an internal customer (e.g., Integration and Test, Formal Test) or released (escaped) to an external customer (e.g., end users)?  For each major release, how many defects were detected in internal development (contained, saves)?  What is the ratio of escaped defects (internal and external) to all defects?  Does committed work (stories, features, capabilities) work as expected? |
| **Base Measure 1** | Contained Defects (integer scale) |
| **Base Measure 2** | Internally Escaped Defects (integer scale) |
| **Base Measure 3** | Externally Escaped Defects (integer scale) |
| **Derived Measure 1** | Total Defects = Contained Defects + Internally Escaped Defects + Externally Escaped Defects |
| **Derived Measure 2** | Internal Defect Escape Ratio = Internally Escaped Defects / Total Defects |
| **Derived Measure 3** | External Defect Escape Ratio = Externally Escaped Defects / Total Defects |
| **Derived Measure 2** | Total Defect Escape Ratio = (Internally Escaped Defects + Externally Escaped Defects) / Total Defects |
| **Indicator Description and Sample** | The concept of categorizing defects as either contained or escaped is key to this measure and others (e.g., Defect Containment). As shown in Part 1 Section 2.3 Figure 3 and repeated below in Figure 1, all defects detected before the release (during development, noted in the blue box) are Contained Defects. All defects detected after release in internal or external operations (noted in the beige and orange boxes) are Escaped Defects.    Figure 1: Defect Terminology  The Defect Escapes table (Table 1) is used to show Contained and Escaped Defects for each release along with the Defect Escape ratio. This measures the quality of the completed product based on the number of defects detected before release (Contained Defects) and after release (Escaped Defects). It also monitors the effectiveness of defect detection processes and verification activities performed during development to detected defects prior to release. Note: while only major releases (e.g., 1.0, 2.0, 3.0) are external releases, it is possible to detect external escapes attributed to minor releases after investigation and assignment of iteration introduced.  Table 1: Defect Detection by Release |
|  | In the example above, Release 1.0 had a ratio of 20% of total escaped defects, with 5% of recorded defects detected after release to the customer. This gradually improved over time to a ratio of 5% on Release 3.0. This was due to a more stable set of requirements, improved test coverage and a more mature product. The Defect Escape Ratio was higher for Release 1.0 because the team decided to implement the more difficult functionality in the first release. Sixty-four defects were discovered in Release 2.0 due to a significant product update. Only 2% of defects were detected externally by the customer.  An alternative way to apply the concept of contained and escaped is to implement the Defect Containment measure. Instead of identifying defects as contained or escaped in relation to the release to an internal or external customer, they would be identified in relationship to iterations. Defects detected in the iteration in which they were inserted (originated) are contained and those detected in later iterations are escaped. Defect counts could be shown in a table as in Table 2 below, identifying which iteration the defects were originated and which iteration the defects were discovered. If this information is unknown, those defects could be tracked separately as Unknown. If legacy defects are detected that were inherited (not originated) by the development team, those could be tracked as Legacy. In a manner similar to the Defect Escape Ratio, various ratios could be determined (e.g., ratio of defects discovered one iteration after they were inserted). See the PSM core framework for more information on Defect Containment.  Table 2: Defect Resolution Lag Time    For this data, 38% of the defects were resolved in the same iteration they were detected. This is less than the organizational goal of 80%. Another 21% were detected in the next release. 41% of defects took at least two iterations to detect, which indicates that the assessment of the iterations needs to be improved, possibly with increased automated test. Some of these escaped defects were not found until after internal release, once an end-to-end test was performed. |
| **Analysis**  **Model** | The Defect Escape Ratio is analyzed to determine the quality of a given release and whether the team is improving over time. The Defect Escape Ratio should be getting smaller over time. The defect containment indicator can be used to evaluate the adequacy and completeness of the testing process and the sufficiency of the automated test.  The enterprise may analyze defect escape ratio across multiple programs, especially external escapes, to evaluate those programs that are successfully handling defects. |
| **Decision Criteria** | Is the Defect Escape ratio acceptable? Is the ratio getting better over time?  Are at least 80% of defects detected in the iteration where they were originated?  Are at least 98% of defects detected before external release? |

|  |  |
| --- | --- |
| **Additional Information** | |
| **Additional Analysis Guidance** | These tables could be separated by priority (e.g., priorities 1-3 and priority 1) or other attributes. This measure may be used in conjunction with other quality measures including the Defect Density, Defect Resolution, and Rework measures. By looking at both internal and external escapes, the team can determine where improvement actions are needed.  A project may intentionally decide to defer defects and add them to the backlog for consideration for resolution in a later iteration or release. These deferred defects may be tagged and tracked separately. |
| **Implementation Considerations** | Defects in the problem reporting tool must be discernable whether they were detected before (contained) or after (escaped) the release to an internal or external customer. A parameter or a review of the dates could be used to determine if defects are contained or escaped. |

|  |  |
| --- | --- |
| **Additional Specification Information** | |
| **Information Category** | Product Quality |
| **Measurable Concept** | Functional Correctness |
| **Relevant Entities** | Defects |
| **Attributes** | Project activity or iteration where defects are detected (e.g., development, internal release, external release). |
| **Data Collection Procedure** | Defect data is recorded in the problem reporting tool as defects are detected.  Each defect must be categorized as contained or escaped by assigning a parameter in the tool or by the iteration or date detected. |
| **Data Analysis Procedure** | Defect counts and ratios are analyzed at the end of each major release to determine status and progress over time. |