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

**Measurement Framework**

**Part 2: Measurement Specifications: Mean Time to Restore**

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 Armycheryl.l.jones128.civ@mail.mil |  | **Geoff Draper**L3Harris Technologiesgeoff.draper@l3harris.com |
| **Bill Golaz**Lockheed Martinwillliam.h.golaz@lmco.com |  | **Paul Janusz**US Armypaul.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, 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](#_Toc70233787)

[8.8 Mean Time to Restore (MTTR) / Mean Time to Detect (MTTD) (Product or Enterprise Measure) 1](#_Toc70233788)

List of Figures

[Figure 1: Operations Outage Summary 2](#_Toc70233789)

List of Tables

[Table 1: MTTR Statistics 2](#_Toc70233790)

# Measurement Specifications

## Mean Time to Restore (MTTR) / Mean Time to Detect (MTTD) (Product or Enterprise Measure)

|  |
| --- |
| **Measure Introduction** |
| **Description** | In an operational environment, continuity of deployed services is fundamental to the delivery of user value. MTTR is essential for systems in which operational availability is critical. This includes both critical embedded systems as well as those systems focused on the delivery of software services.Operations can be impacted by planned or unplanned outages. Operational service incidents are typically recorded in a trouble ticket which is used to track the incident to closure and restoration of service. Each trouble ticket has an associated restoration time. Sometimes there may be an alternative or workaround that enables the service to continue in the field, such as redundant paths or resources, even if in a degraded mode. Some repairs must be returned to the factory for correction and redeployment. The enterprise may collect the average time to detect a service-impacting issue (Mean Time to Detect) and the average restoration time (Mean Time to Restore). This provides measures of operational effectiveness for maintaining service continuity, across all tickets, or classes of tickets. A summary of these concepts is depicted visually in Figure 2, Measurement Context Diagram.MTTR, MTTD and other operational measures of service continuity can be applied in each of many potential stakeholder environments including the development/integration environment(s), production representative environment, or operationally relevant environment, or the operational environment. The enterprise generally focuses on actual measures from the operational environment. The product team may also focus on ensuring MTTR/MTTD objectives will be met as the system is developed and sustained. |
| **Relevant Terminology** |

|  |  |
| --- | --- |
| Mean Time to Detect (MTTD) | Time required to identify an interruption to service delivery. MTTD measures how long it takes the operations team to detect that an incident has occurred which affects delivery of operational services. |
| Mean Time to Restore (MTTR) | Time required to restore service after an outage occurs. MTTR measures how long it takes the operations team to restore the system to an operational state, either through a rollback, restart, fix in operations, return to the factory for repair, or another action. Sometimes synonymous with Mean Time to Recover, but with a focus on restoration of operations. |

 |

|  |
| --- |
| **Information Need and Measure Description** |
| **Information Need** | What is the reliability and availability of operational capabilities? How long does it generally take to restore service when a service incident occurs? How quickly can we recover from failures that impact the system in operations (e.g., impacts service reliability or availability), or the software in development or test?*(time to restore the build or the service to a previous, known good state.)* |
| **Base Measure 1** | Failure Occurrence Time *(timestamp)*  |
| **Base Measure 2** | Failure Detection Time *(timestamp)* |
| **Base Measure 3** | Service Restoration Time *(timestamp)* |
| **Derived Measure 1** | Time to Detect = (Failure Detection Time) – (Failure Occurrence Time)*(units for elapsed time may vary; seconds, minutes, hours, days)* |
| **Derived Measure 2** | MTTD = ∑ (Time to Detect) / N *(rolling average Time to Detect, based on N previous failures)* |
| **Derived Measure 3** | Time to Restore = (Service Restoration Time) – (Failure Occurrence Time)*(units for elapsed time may vary; seconds, minutes, hours, days)* |
| **Derived Measure 4** | MTTR = ∑(Time to Restore) / N *(rolling average Time to Restore, based on N previous failures)* |

|  |
| --- |
| **Indicator Specification** |
| **Indicator Description and Sample** | When practicing CID, a key concern is speed: to deliver software rapidly and frequently. However, quality should be maintained. In particular, when practicing Continuous Deployment into operations it is important to be able to quickly recover when a new release/deployment introduces a failure in this live environment.MTTD and MTTR indicators can be represented in multiple ways (e.g., graphical, tabular). In Figure 1, three measures are plotted for each operational outage: Time to Detect, Time to Repair, and Time to Restore (sum of detection + repair). A comparison of data across outages indicates general trends, severity, and operational impacts. A summary of statistical measures (mean, median, standard deviation) for each of detection time, repair time, and total restoration time is provided in the table below the chart. A rolling average of Mean Time to Restore (MTTR) is also plotted for the 10 most recent outages.Figure 1: Operations Outage Summary

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 1: MTTR Statistics

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Detect** | **Repair** | **Restore** |
| **Mean** | 11.56 | 51.06 | 62.61 |
| **Median** | 11 | 51 | 59 |
| **Std Dev** | 6.31 | 34.09 | 33.28 |

 |

In this example, although there are significant variations in individual outage samples (some anomalies are more complicated to fix than others), in aggregate the MTTR rolling average is holding fairly steady (around 1 hour to restore service). Similarly, the mean and median times for Time to Detect, Time to Repair, and Time to Restore are consistent despite a large standard deviation. (Table 1)In the sample indicator, the four short MTTRs are cases where the system was rolled back to a previous version. The longest cases are indicative of complex issues that required additional repair time. The lengthy MTTR in Outage 16 involved an update to a critical component. The fix/corrective action was not implemented correctly, which resulted in Outage 17. An alternative solution was implemented, and the software was shown to work in the next iteration.In this example, feedback from the user community indicates outages of greater than 30 minutes can have a significant impact on Operations, due to reports that are due twice hourly. Missing two consecutive reports impacts decision making. This example program is considering ways to shorten restore times, such as implementing automated roll-back capabilities where any new deployment/release that introduces a failure can be rolled back and the previous release rapidly restored. Program personnel are also conducting a Pareto analysis of outage times by defect type to determine which outage types are most costly, so that resources can be prioritized on targeted improvement actions. |
| **Analysis** **Model** | Data is gathered from service incident tickets and classified or filtered into affinity groupings of interest (e.g., priority, type, component, severity, impact, duration, detection method). Trends and root causes are evaluated. Improvement plans may be defined and implemented with corrective/preventive actions to mitigate the frequency or impact of future occurrences, as appropriate, relative to business objectives. The effectiveness of improvement actions should also be measured.Both MTTD and MTTR need to be evaluated as to whether they meet the business/mission needs in terms of reliability and availability. Projections and actuals are evaluated against objectives, and trends are analyzed to project whether required objectives will be met. A good pipeline should include significant automated testing such that any failure-inducing defects or issues are detected before deploying into the operational environment.MTTD and MTTR are measures of failure trends for a set of issues across a range of time, and they characterize the capability to maintain and rapidly restore operations and operational service. Analysis and improvement actions can vary based on the situation and trends of performance measures and whether these are reliable predictors of future performance so improvement actions can be effective. Examples of potential areas for investigation are summarized in the table below:

|  |  |  |
| --- | --- | --- |
| Trend | MTTD | MTTR |
| Increasing | * Ineffective monitoring, detection processes, tools, training
* Incomplete knowledge of failure modes
 | * Increasing complexity of system, software, or architecture
* Lack of rollback capability or strategy
* Lack of effective redundancy
* Developer changes / inexperience
 |
| Steady | * Established MTTD met and satisfied - no further improvement needed
* Predictable capability; does it meet the business need (Voice of the Customer)?
* Lack of continuous improvement
 | * Established MTTR met and satisfied - no further improvement needed
* Predictable capability; does it meet the business need (Voice of the Customer)?
* Lack of continuous improvement
 |
| Declining | * Improved monitoring effectiveness
* Defect prevention initiatives
 | * Improvements through automation, tools
* Added capability or capacity (redundancy, etc.)
 |
| Erratic | * Inconsistent monitoring or reporting processes
 | * Unstable processes
* Immature system
* Ineffective process improvement
 |

 |
| **Decision Criteria** | After deployment, when MTTR or MTTD is above mission or business objectives, a decision as to whether the system should be rolled back to a previous version may be considered. If the decision is not to roll-back, the user may create a high priority change request to resolve the issue causing the high MTTR. Increasing trends in MTTR or MTTD measures, may also lead to the creation of new defects or stories to improve performance, or the need to evaluate and improve the development/test processes. This is especially important when a safety critical or mission critical failure occurs.When additional defects are introduced after improvements are made, special attention should be applied to the resolution process. During development and test, for any MTTD or MTTR that is more than 10% above the objective or mean, investigate the root cause(s) and decide if additional improvements or testing is required. Trends over time should be improving (getting smaller) as additional functionality is added and as the system nears deployment. Regular occurrences above the objective may mean that the system is not mature enough for operations, and deployment may need to be delayed. For trends that are increasing above the objective or mean, additional focus or process improvements may be required. |

|  |
| --- |
| **Additional Information** |
| **Additional Analysis Guidance** | MTTR is an essential measure for systems in which operational availability (Ao) is critical, with a focus on safety-critical and mission-critical failures.MTTR is also paramount when practicing full continuous deployment into Operations: in this case Operations is an operational environment supporting live operations/missions and thus the system must maintain high reliability and availability. However, even in testing environment, a failure means that integration or test activities are impacted (and possibly deployment which may lead to cost/schedule overruns).Additional analyses of MTTR / MTTD measures can be utilized to determine appropriate actions to improve availability and rapid recovery from operational issues. Examples include statistical analysis methods, profiles of defect distribution or characteristics, Pareto charts, root cause analysis, or other quality management tools. |
| **Implementation Considerations** | Measuring individual failures and restorations should be automated as much as possible, based on timestamps in logs or other automated data collection mechanism. |

|  |
| --- |
| **Additional Specification Information** |
| **Information Category** | Process Performance |
| **Measurable Concept** | Process Efficiency – SpeedSupportability – Maintainability – Dependability – Reliability |
| **Relevant Entities** | Service incidents |
| **Attributes**  | Time of outage, detection, and restoration; defect priority and reason code; affected elements |
| **Data Collection Procedure** | Date/time is collected at the start of each failure or service outage, and at the time of operations or service restoration. The delta between these is the individual outage TTR. These are collected to calculate a historical mean MTTR. |
| **Data Analysis Procedure** | Data is analyzed periodically during development and test, and trends are evaluated. During operations, data is analyzed when safety or mission critical failures occur, as well as periodically.  |