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

**Measurement Framework**

**Part 2: Measurement Specifications: Release Frequency**

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](#_Toc70233989)

[8.9 Release (or Deployment) Frequency (Product or Enterprise Measure) 1](#_Toc70233990)

List of Figures

[Figure 1: Iterative Development 1](#_Toc70233991)

[Figure 2: Product Iterative Releases (Conceptual) 2](#_Toc70233992)

[Figure 3: Release Duration for Product Tango 4](#_Toc70233993)

[Figure 4: Product Release Frequency 4](#_Toc70233994)

List of Tables

[Table 1: Product Release Averages 3](#_Toc70233995)

[Table 2: Release Frequency and Labor Hours 3](#_Toc70233996)

# Measurement Specifications

## Release (or Deployment) Frequency (Product or Enterprise Measure)

|  |  |
| --- | --- |
| **Measure Introduction** | |
| **Description** | As described in Overarching Principles, products are typically planned and developed iteratively (e.g., capabilities, features, stories, tasks) into a set of internal releases, candidate releases, and deployed product releases. This is represented conceptually in Figure 1.    Figure 1: Iterative Development  The speed and frequency at which products are released are crucial in providing useful capability to users as rapidly as possible. The scheduling, duration, and frequency of releases can vary widely (e.g., months, weeks, days, or on demand) based on domain or business need. Products may be iteratively released on a predictable fixed cadence, or on demand as needed. The time and effort to develop candidate product releases and transition them to deployed external product releases are primary measures of efficiency in making features/capabilities available to users, as depicted in Figure 2.    Figure 2: Product Iterative Releases (Conceptual) |
| **Relevant Terminology** | |  |  | | --- | --- | | MVP | Minimum Viable Product | | MVCR | Minimum Viable Capability Release | | NVP | Next Viable Product | | Release | Internal Release; Candidate Release (External Release); Operational Release (Deployment Release) |   Refer to glossary for definitions. |

|  |  |
| --- | --- |
| **Information Need and Measure Description** | |
| **Information Need** | How long does it take to deploy an identified feature/capability? *[Product]*  What is the cadence or frequency for product release or deployment? *[Product, Enterprise]*  How long does it take to release a minimum viable product? *[Product, Enterprise]*  How much effort/cost/time is needed to develop new products and transition them to release? *[Product, Enterprise]* |
| **Base Measure 1** | Release Start Date (datestamp) *(release, candidate release, or operational release)* |
| **Base Measure 2** | Release End Date (datestamp) *(release, candidate release, or operational release)* |
| **Base Measure 3** | Effort Hours to generate a release (integer) *(internal release candidate or external deployed release)* |
| **Base Measure 4** | # of Releases (for a specified data range) |
| **Derived Measure 1** | Release Duration = (Release End Date) – (Release Start Date)  Note: release durations may be tracked for features/capabilities at various stages of maturity   * Time to Minimal Viable Product (MVP) * Time to Minimal Viable Capability Release (MVCR) * Time to Next Viable Product (NVPn) |
| **Derived Measure 2** | Release Frequency = (# of Releases) / date range (e.g., days, weeks, months, quarters, years) |
| **Derived Measure 3** | Average Release Duration = ∑ (Release Duration) / (# of Releases)  *Note: weighting can be used to emphasize the most recent releases.* |
| **Derived Measure 4** | Average Release Transition Time = ∑ (Release Transition Time) / (# of Releases) |

|  |  |
| --- | --- |
| **Indicator Specification** | |
| **Indicator Description and Sample** | In this example, (Table 2) a commercial software company deployed a new product (Tango) to the market in October 2018 (MVP release), with a business objective to release iterations twice monthly to support quarterly product capabilities releases. Ten product releases were completed between October 2018 and March 2019. The table below summarizes, for each release, the start and end dates for each release (from which duration is calculated), the type of release, and the total labor spent in hours.   |  | | --- | | Table 1: Product Release Averages |   Following the higher effort for the initial MVP R2018.01 release, durations of iterations have averaged 18 days. The initial MVP did not meet market needs, however, a Minimum Marketable Product (MMP) was available two months later in December 2018. After the MMP, the NVP release occurred 90 days later, in line with the business objective of quarterly releases.  A longer duration for the R2019.01 iteration (25 days) at the end of 2018 is attributed to staffing reductions due to holiday vacations. Overall averages for release time and labor across releases is shown in the Table 1, by calendar year.  Table 2: Release Frequency and Labor Hours    The product team plots each release in the Figure 3 below for a visual comparison of durations (vertical bars aligned with the left axis) and labor hours invested (teal line aligned with the right axis). A rolling average of the durations for the most recent 3 product releases is calculated and displayed in the dashed red line.    Figure 3: Release Duration for Product Tango  In Figure 4 the marketing department tracks the release frequency for all three of the company’s products at the enterprise level against the business plan for twice monthly iterative releases.    Figure 4: Product Release Frequency |
| **Analysis**  **Model** | Can we consistently release product baselines at a rate needed to meet demand?  Is the process performance (time and labor) for generating and deploying product releases improving? Does it take more/less/same amount of time to transition release products to candidate release products or operational release products?  Not all development organizations are in control of when their internal baselines may be deployed to live operations. For instance, deployments for military platforms must be certified and coordinated with the user community. As shown in Figure 1, additional effort may be needed to prepare and transition candidate releases to operationally representative environments. This may require a separate set of release measures to manage and optimize the rapid delivery of capabilities to end users. This preparation and transition effort may increase significantly as the baseline grows. Not only must the new capability be verified, so must prior functionality be verified through regression testing. If done manually, the additional effort for testing and release can scale at a rate incompatible with maintaining product release timelines.  Automation can help improve build, testing, and release efficiency to maintain a consistent release transition cadence.  The time to build and create product releases is directly related to the quantity and size of design and implementation features. Smaller batch sizes enable releasing products more quickly. Efficiency of the deployment and release process further accelerates the speed at which products can be released to the customer.  Releases are typically built by automated build/test automation frequently on the baseline. Releases are typically built every day or upon every merge or end of a sprint. The frequency of failures for releases impacts confidence in the software baseline. Ideally over time, releases can be produced more efficiently by replacing manual steps with automation. |
| **Decision Criteria** | * If the effort to transition release products to candidate releases or operational releases is increasing steadily beyond performance goals, consider approaches such as automation or reducing batch size to increase release frequency and speed the delivery of capability to users. * Once stabilized, action may be needed if the quality of deployed products declines or if the team is unable to sustain release timelines. * Does adding features/capabilities result in increased cost to create a candidate release or operational release? |

|  |  |
| --- | --- |
| **Additional Information** | |
| **Additional Analysis Guidance** | Release frequency (how often?) have close dependencies with Lead Time and Cycle Time (how fast?) measures. All these measures rely on the batch size of the capability or stories being released, and the efficiency of the pipeline in generating and provisioning products. Automation of the build/test elements has a profound impact on all these measures. Consistency of staffing and team composition can also impact the team’s ability to release their capabilities as needed. Generally faster release cycles on a predictable cadence are desirable to quickly deliver value to users and obtain feedback.  There can be a tension between speed and quality tradeoffs. An over-emphasis on speed can be at the expense of product quality. There is often a ‘sweet spot’ tradeoff between speed and quality that delivers a best value solution based on project objectives. Quality needs to be monitored, in addition to speed, to ensure that these measures are appropriately balanced.  Additional statistical measures can be generated (e.g., mean, median, standard deviation, quartiles) to determine the aggregate performance, repeatability, and consistency of product release timelines. |
| **Implementation Considerations** | Applying Build/Test Automation to generate releases as early in the program as possible is recommended. Successfully generating releases as early in the release cycle should be a team priority.  Integrity of the product baseline can be ensured by enforcing quality criteria for baseline merges to proceed successfully through the build/test automation pipeline. |

|  |  |
| --- | --- |
| **Additional Specification Information** | |
| **Information Category** | Process Performance |
| **Measurable Concept** | Process Efficiency – Speed |
| **Relevant Entities** | Releases, Effort |
| **Attributes** | Quantity, Labor Hours |
| **Data Collection Procedure** | Date/time is collected at the start and end of each iteration or release (iteration or deliverable, internal or external), typically obtained directly from automated tools. Each release must meet entry and exit criteria to be considered complete. Cycle time is calculated as the difference between release start time and release end time. Release frequency is calculated as the number of releases completed per unit time (e.g., day, week, month, year). |
| **Data Analysis Procedure** | Measures of the release process are analyzed at end of each release for performance within acceptable bounds, with corrective actions or improvements taken as necessary. |