



IEEE Standard for Measures of the Software Aspects of Dependability

IEEE Computer Society

Developed by the
Software & Systems Engineering Standards Committee

IEEE Std 982™-2024
(Revision of IEEE Std 982-2005)

STANDARDS

IEEE Standard for Measures of the Software Aspects of Dependability

Developed by the

Software & Systems Engineering Standards Committee
of the
IEEE Computer Society

Approved 26 September 2024

IEEE SA Standards Board

Abstract: Definitions, sample requirements, equations, and guidance for data collection and analysis necessary to make consistent and actionable measurements of a software system's reliability, availability, supportability, and recoverability are provided in this document. The dependability of software systems is collectively defined by these characteristics.

Keywords: availability, dependability, IEEE 982™, measurement, recoverability, reliability, supportability

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2024 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1 November 2024. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 979-8-8557-1326-8 STD27387
Print: ISBN 979-8-8557-1327-5 STDPD27387

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all IEEE standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers involved in technical working groups are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning all standards, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. IEEE Standards documents do not guarantee safety, security, health, or environmental protection, or compliance with law, or guarantee against interference with or from other devices or networks. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document should rely upon their own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus balloting process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English language version published by IEEE is the approved IEEE standard.

Use by artificial intelligence systems

In no event shall material in any IEEE Standards documents be used for the purpose of creating, training, enhancing, developing, maintaining, or contributing to any artificial intelligence systems without the express, written consent of IEEE SA in advance. “Artificial intelligence” refers to any software, application, or other system that uses artificial intelligence, machine learning, or similar technologies, to analyze, train, process, or generate content. Requests for consent can be submitted using the Contact Us form.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual is not, and shall not be considered or inferred to be, the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE or IEEE SA. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter’s views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s). News releases about IEEE standards issued by entities other than IEEE SA should be considered the view of the entity issuing the release rather than the formal position of IEEE or IEEE SA.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments or questions, except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE SA working group. You can indicate interest in a working group using the Interests tab in the Manage Profile and Interests area of the [IEEE SA myProject system](#).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.²

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory

¹Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

²Available at: <https://standards.ieee.org/about/contact/>.

requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

³Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴Available at: <https://standards.ieee.org/standard/index.html>.

Patents

IEEE standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

Technologies, application of technologies, and recommended procedures in various industries evolve over time. The IEEE standards development process allows participants to review developments in industries, technologies, and practices, and to determine what, if any, updates should be made to the IEEE standard. During this evolution, the technologies and recommendations in IEEE standards may be implemented in ways not foreseen during the standard's development. IEEE standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, data privacy, and interference protection practices and all applicable laws and regulations.

⁵Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

Participants

At the time this draft standard was completed, the Standard for Measures of Software Aspects of Dependability Working Group had the following membership:

Robert Binder, *Chair*

Pieter Botman	Louis Gullo	Ann Marie Neufelder
Lynn Robert Carter	Jon Hagar	Joanna Olszewska
Sigrid Eldh	Pratap Lakshman	Annette Reilly
Michael Grottke	Phillip Laplante	Rob Schaaf

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Boon Chong Ang	Louis Gullo	Thomas Owens
Butch Anton	Jon Hagar	Howard Penrose
Robert Binder	Sheri Harshberger	Rajaraman
Juris Borzovs	Werner Hoelzl	Ramachandran
Pieter Botman	Ronald Jarrett	R. K. Rannow
Lawrence Catchpole	Piotr Karocki	Annette Reilly
Zhiman Chen	Madhav Krishna	Robert Schaaf
Diego Chiozzi	Xiao Liang	Jhony Sembiring
Scott Crawford	Murray Little	Thomas Starai
Ronald Dean	Johnny Marques	Walter Struppler
Kurniawan Diharja	Roberto Moreno	John Vergis
David Fuschi	Rajesh Murthy	Oren Yuen
Emanuel Grant	Andrew Nack	Janusz Zalewski
	Joanna Olszewska	

When the IEEE SA Standards Board approved this standard on 26 September 2024, it had the following membership:

David J. Law, *Chair* **Jon Walter Rosdahl, *Vice Chair*** **Gary Hoffman, *Past Chair*** **Alpesh Shah, *Secretary***

Sara R. Biyabani	Hao Hu	Paul Nikolich
Ted Burse	Yousef Kimiagar	Robby Robson
Stephen Dukes	Joseph L. Koepfinger*	Lei Wang
Doug Edwards	Howard Li	F. Keith Waters
J. Travis Griffith	Xiaohui Liu	Sha Wei
Guido R. Hiertz	John Haiying Lu	Philip B. Winston
Ronald W Hotchkiss	Kevin W. Lu	Don Wright
	Hiroshi Mano	

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 982-2024, IEEE Standard for Measures of the Software Aspects of Dependability.

Reliance on software in all manner of critical systems has increased tremendously since the previous version of this standard was issued, as have concerns regarding growing cyber threats. The expanding dialog on software dependability and the increasing body of work in software assurance call for a broader perspective and a revision of this standard to ensure its continuing relevance. Therefore, the revised standard updates and augments the set of measurable attributes of the software aspects of dependability while acknowledging the significance of prior work, preserving its rigor, and retaining essential measures and concepts.

This standard provides equations and definitions to characterize in-service dependability along with guidance to develop these measurements. It presents a unified framework for measurement of quality characteristics evidenced by system behavior, including functionality, performance, safety, and security; and by the time consumed by certain kinds of support actions. The intended audience for this standard is primarily software and systems engineers; persons operating or supporting software systems; sponsors, managers, and users of software systems; researchers and other parties interested in software systems.

Dependability measurements, produced at any life cycle stage, can be used for many purposes: evaluation of operational effectiveness, evidence for key performance indicators, service level management, service monetization, system health monitoring, troubleshooting, identification of product gaps and opportunities, failure identification and mitigation, dependability prediction, allocating resources for support, and regulatory compliance.

Although this standard focuses on in-service measurement, its definitions can also be applied during development. Such pre-release measurements are essential to guide system engineering, manage development activities, improve development processes, predict operational dependability, and assess post-release risks.

As part of a quality management system, development and operational dependability measurements can be used to manage risks and as an input for preventative and correction actions for process improvement.

This standard supersedes and completely replaces inactive standard IEEE Std 982.1™-2005, IEEE Standard Dictionary of Measures of the Software Aspects of Dependability, which replaced IEEE Std 982.1™:1988, IEEE Standard Dictionary of Measures to Produce Reliable Software.

Contents

1. Overview	10
1.1 Scope	10
1.2 Purpose	10
1.3 Applicability	10
1.4 Word usage	12
2. Normative references	13
3. Definitions, acronyms, and abbreviations	13
3.1 Definitions	13
3.2 Acronyms and abbreviations	19
4. Dependability measurement model	21
4.1 Dependability is a composite quality characteristic	21
4.2 System of interest (SOI)	21
4.3 Content and role of dependability requirements	22
4.4 Dependability requirements	24
4.5 In-service reference model (ISRM)	26
4.6 Timing	31
4.7 Abstract data model	35
5. Dependability measures	36
5.1 Reliability	36
5.2 Availability	48
5.3 Supportability	60
5.4 Recoverability	62
Annex A (informative) Bibliography	67

IEEE Standard for Measures of the Software Aspects of Dependability

1. Overview

1.1 Scope

This standard identifies, classifies, and defines measures of the software aspects of in-service dependability. As such, it provides a framework for stakeholders to understand the software aspects of dependability and to make dependability measurements throughout the system life cycle.

It provides definitions, requirements, and guidance for the data collection and analysis necessary to make consistent and actionable measurements of a software system's reliability, availability, supportability, and recoverability.

This standard is applicable to any software system or service. It is particularly useful for critical systems where the reliability of their safety, security, functionality, performance, and utilization, as well as recoverability, supportability, and availability are foremost. Examples of critical systems include those supplying communications infrastructure, first responder services, energy production and distribution, real time financial transactions, medical devices, health care administrative services, defense capabilities, personal and mass transportation, environmental monitoring, and water and waste management.

1.2 Purpose

The purpose of this standard is to establish guidance for measuring the software aspects of dependability as applicable to a system's operational context. It is intended for in-service measurement. It may be used during requirements, architecture, and design development; software construction and integration; and verification and validation of the software products of these activities. The overall objective is to enable stakeholders to measure dependability in software intensive systems throughout software operations and maintenance.

1.3 Applicability

This standard is intended for use in any domain of software application, whether user-facing or embedded, including personal convenience, real-time control, medical devices, high-volume transaction processing, high-performance numerical computing, data communications, industrial control, or artificial intelligence, to name a few. It is intended for use with any kind of computer hardware, runtime stack (operating system), or programming language; or in any computing infrastructure including personal computers, smartphones, on-board embedded devices, industrial controllers, edge devices, network operations centers, or data centers.

As a catalog of measures, this standard:

- Defines variables and equations to quantitatively characterize the software aspects of dependability: reliability, availability, supportability, and recoverability.
- Presents the in-service reference model (ISRM), a framework for uniform identification of system behaviors and support actions to be measured.
- Presents the reliability class model to support consistent and extensible differential analysis of the effects of anomalies and failures on the reliability of safety, security, functionality, performance, and utilization.
- Defines downtime subsets for certain types of activity (slices), providing insight into factors contributing to availability and supportability.
- Calls for explicit and bounded dependability requirements as a prerequisite for measurement and therefore provides normative criteria and guidance for their development and use.

It does not address other pre-release development practices including 1) software reliability prediction, 2) software engineering techniques for improving dependability, or 3) software maintenance.

IEEE Std 1633™-2016, Recommended Practice on Software Reliability [B15] provides a thorough resource for topics 1 and 2.⁶ ISO/IEC/IEEE Std 14764:2022, Software engineering—Software life cycle processes—Maintenance [B21] likewise addresses topic 3.

Although this standard calls for well-formed and complete expressions of capability and dependability requirements, it does not address how to set specific levels of reliability, availability, supportability, or recoverability for any system or class of systems. It does not address how to develop software artifacts that achieve acceptable levels of functionality, performance, utilization, fault-tolerance, security, or safety.

This standard contains notional examples of requirement statements for reliability, availability, supportability, and recoverability. These are informative and not requirements of this standard.

Certain software engineering and operational processes, practices, and artifacts tend to result in higher dependability; their absence tends to result in lower dependability (Bourque and Fairley [B3]). Although general principles of software engineering for dependability are well-established, specific practices evolve and become irrelevant, often rapidly. This standard therefore focuses on defining useful measures of in-service dependability, regardless of the developmental and operational practices applied to a system of interest (SOI).

NOTE—IEEE 1012™-2016 [B13] provides guidance for requirements-based verification and validation of software/hardware systems.⁷

Often, identical copies of a software system are installed on numerous host computers. A population of deployed instances often has substantial variation in versions, configuration, operating conditions, and usage patterns. This variation typically has material effects on the dependability of any given instance. When this standard was revised, no generally accepted and validated techniques or models for measuring dependability of a software population were published and proven in practice. Therefore, this standard's dependability measurements are limited to a single instance of a system in a particular context; it does not address obtaining or characterizing population dependability measurements.

⁶The numbers in brackets correspond to those of the bibliography in Annex A.

⁷Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

Consistent and accurate measurements of in-service dependability and other quality attributes are essential for effective corrective and preventative actions (CAPA) in all software life cycle stages. Software product and process improvement through CAPA is a well-established and well-documented best practice. This standard therefore does not address CAPA.

NOTE—ISO/IEC/IEEE 90003:2018 [B18] provides comprehensive guidance for establishing a quality management system for software products and identifying software-related CAPAs.

This standard does not attempt to provide a mathematically formal and complete model for a measurement system, its abstract functions, or its binding to any data management system. It does not identify techniques to obtain logs or records of events from any specific software SOI or runtime environment.

To characterize dependability, this standard is focused on the classification and quantitative characterization (analysis) of certain operational activities and system behaviors from an external observer's perspective. Except to the extent necessary to make these measurements, it does not address other engineering or operational topics that may affect or depend on the nature or degree of these behaviors. It does not address characterizations of source-code, software architecture, or development practices, as they do not produce in-service behavioral data. Although some implementation measurements have been shown to be useful leading indicators of in-service dependability, many application-specific factors can confound these predictions (Hall, Beecham, Bowes, Gray, and Counsell [B9]).

Most extant definitions of software system trustworthiness include some or all of the dependability measurements defined in this standard (Newman [B30]). They also call for evidence to support claims that a system is sufficiently unlikely to behave in an anomalous, insecure, or harmful manner, and that its future behavior is likely to be consistent with socio-technical norms. These claims may be supported by evidence or analysis of any phenomena deemed relevant (Cho, Chan, and Adali [B3]). The observational measurements of this standard can provide some of this evidence, but do not address the broad scope of artifacts and activities or the subjective characterizations necessary to judge system trustworthiness.

1.4 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{8,9}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

⁸The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

⁹The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

ISO/IEC/IEEE 15939, Systems and software engineering—Measurement process.^{10,11,12,13}

ISO/IEC/IEEE 24765, Systems and software engineering—Vocabulary.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.¹⁴ Definitions for other system and software engineering terms can be found in ISO/IEC/IEEE 24765, available at <www.computer.org/sevocab>.¹⁵

abnormal termination: Uncommanded cessation of software execution, typically resulting in a terminal failure. *Syn:* abend, blue screen, crash, dump, panic.

absolute time: A tick in a generally agreed chronological frame of reference. *Contrast:* **relative time**, **natural unit**.

EXAMPLE—a calendar date and time of day in a given reference time zone, such as Coordinated Universal Time (UTC).

abstract behavior: A description, model, or other general characterization of similar behaviors.

NOTE—There are many representations that express abstract behavior, for example, use cases, protocol message types, sequence diagram messages, and interface control documents.

adaptive support: Modification of an in-service SOI performed after installation to keep it usable in a changed or changing runtime environment. (Adapted from IESO/IEC/IEEE 14764:2022 [B21])

additive support: Modification of an in-service system performed after installation to add functionality or features to a system instance. (Adapted from IESO/IEC/IEEE 14764:2022 [B21])

NOTE—In contrast to perfective support, additive support results in new functions or features.

anomaly: A judgment that a system's behavior or artifact deviates from expectations based on 1) the system's high-level goals for outcomes, similar requirements, development or operational documentation, 2) applicable standards, regulations, or laws, or 3) an observer's perceptions or experiences.

NOTE—An anomaly can result in a revealed requirement.

¹⁰ISO publications are available from the International Organization for Standardization (<https://www.iso.org/>) and the American National Standards Institute (<https://www.ansi.org/>).

¹¹IEC publications are available from the International Electrotechnical Commission (<https://www.iec.ch>) and the American National Standards Institute (<https://www.ansi.org/>).

¹²IEEE standards and products are trademarks owned by The Institute of Electrical and Electronics Engineers, Incorporated.

¹³IEEE publications are available from The Institute of Electrical and Electronics Engineers (<https://standards.ieee.org/>).

¹⁴*IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

¹⁵Information on references can be found in [Clause 2](#).

availability: The ratio of uptime divided by the sum of uptime plus downtime, that is, $uptime/(uptime + downtime)$.

NOTE 1—In the ISRM, the proportion of an in-service span during which an SOI is in READY or RUNNING mode.

NOTE 2—Availability is a quantitative component of dependability, expressed as a requirement parameter (threshold), as a probability (prediction or estimate), or as a measurement obtained during test or in-service operation (observation).

behavior: An observable sequence of inputs and outputs for a system, sufficient to evaluate compliance with its requirements, or for an observer to identify an anomaly. (Adapted from ISO/IEC/IEEE 24765)

blocking bug: A software defect that obstructs further work on a particular task or feature until it is resolved.

capability requirement: A requirement that specifies a condition or behavior that a system is to achieve under stated conditions. *Contrast:* **dependability requirement**.

configuration item (CI): Unit or aggregation of hardware, software, or both that is designated for configuration management and treated as a single entity in the configuration management process.

NOTE—Although a flat file is the most commonly used software configuration item, items can vary widely in complexity, size, and type, ranging from an entire system including all hardware, software, and documentation, to a single module or a minor hardware component.

contingent threshold: A threshold value in a dependability requirement that is desirable but optional owing to uncertainty about the trade-offs related to its achievement.

NOTE—A contingent threshold typically quantifies a capability level that is required only if it can be achieved without incurring excessive cost, unfavorable risk, or an unfavorable operational effect.

corrective support: Modification of an in-service system to remove faults causing failures or anomalies. (Adapted from IESO/IEC/IEEE 14764:2022 [B21])

NOTE—Corrective support changes an in-service system so that it satisfies defined system requirements.

countable failure: Anomalous behavior that 1) does not satisfy one or more of a system's explicit or revealed capability requirements, 2) meets or exceeds the severity criteria of its reliability class, and 3) the tick at which it occurred is imputed or recorded.

data corruption: Incorrect and unrecovered alteration to a data item, including malicious encryption.

data erasure: Irrevocable loss of a data item.

data exposure: Unpermitted access to a data item, without actual unpermitted usage or transfer.

data failure: An occurrence of data corruption, erasure, exposure, or leakage occurring during any IN-SERVICE or REMOVED mode of an SOI instance.

data leakage: An unpermitted usage or transfer of a data item.

dependability: Extent to which measured values of reliability, availability, recoverability, and supportability meet or exceed their required thresholds during an in-service interval.

dependability measurement: Data collection and processing to produce values for reliability, availability, supportability, or recoverability of a system.

dependability requirement: A requirement statement that specifies a bounded threshold for one or more parameters of reliability, availability, supportability, or recoverability for a system. *Contrast:* **capability requirement**.

development environment: A workplace facility in which a system is developed and maintained; it has staffing, resources, interfaces, capabilities, and tooling not found in that system's target environment(s).

downtime: An in-service interval during which a system has not performed or cannot perform its required capabilities.

NOTE—Downtime may result from planned or unplanned support activities, system failures, or failures originating in a system's runtime or operational environments.

duration: The positive difference between any two ticks of the same type.

emergency support: Unscheduled in-service modification performed to temporarily keep a system operational, pending corrective support. (Adapted from IESO/IEC/IEEE 14764:2022 [B21])

NOTE 1—Emergency support can include modifications to a system, its configuration, or runtime environment that alter operations, functionality, inputs, outputs, interfaces, storage, and other resources, or by establishing temporary usage alternatives (workarounds) to avoid triggering anomalies or failures.

NOTE 2—Emergency support can be regarded as a corrective support type.

event: In the ISRM, a system behavior, support action, or other condition that can result in a mode change.

event record: A notional data type representing the attributes of an ISRM event.

explicit requirement: A requirement expressed in writing, often using natural language or a semi-formal notation.

failure: The departure of system behavior from system requirements. *See also:* **countable failure**.

failure intensity: A quantitative characterization of in-service reliability as a ratio of the number of observed failures to the duration of an observed span. *Contrast:* **failure rate**.

NOTE—Failure intensity can be expressed with absolute units (e.g., failures per hour), relative units (e.g., failures per active CPU seconds) or natural units (e.g., failures per non-maskable interrupts in a mission).

failure intensity objective (FIO): A failure intensity level to be achieved during pre-release testing as part of a system's release criteria.

failure rate: In software reliability growth models, the rate of change in the number of failures per unit of time according to a model and its parameters. *Contrast:* **failure intensity**.

failure severity: A step on a scale ranking the detrimental effect of failures.

NOTE—For a particular failure, its assigned rank.

history file: A notional data type consisting of a collection of event records that can be processed to obtain durations, counts, and statistics for support actions, events, and modes over a span of interest.

impairment: A non-terminal system failure or change in a runtime environment resource or operating condition that degrades system capabilities but does not result in an immediate terminal failure.

NOTE—An impairment can be a result of a system fault, an as-designed system response, or an environment condition. It can trigger other impairments or failures. Each impairment has a duration and can result in lossage.

in-service: A portion of a system's life cycle that includes support actions to establish, sustain, and deactivate an SOI instance, and the active use of an SOI instance for its intended purpose.

NOTE 1—ISRM modes include prerequisite support actions to establish the SOI, support actions to provide resources and conditions necessary for its execution, its uninstallation, and custodial support actions related to its residue.

NOTE 2—The ISRM IN-SERVICE mode is comprised of modes SUPPORTING, BLOCKED, READY, RUNNING, and INOPERABLE. It may be concurrent with DEVELOPMENT/MAINTENANCE. It is typically concluded with uninstallation, resulting in the REMOVED mode.

in-service period: The time span starting with the first installation of a system and ending with the first of a) its replacement by a materially different version of that system, b) its uninstallation, abandonment, or c) the demise of its runtime or operational environment.

in-service reference model (ISRM): A model of support actions, events, and modes of an in-service software system that supports classification of SOI behavior and subsetting of in-service time spans necessary for dependability measurements.

lossage: The detrimental effect on stored data of a failure or an impairment.

mean: The arithmetic average of a collection of numbers.

NOTE—For n observations $x_1, x_2, \dots, x_n$, the arithmetic mean $\bar{x} = (x_1 + x_2 + \dots + x_n)/n$

mode: In the ISRM, a generic operational activity or system status whose duration is a variable for reliability, availability, supportability, or recoverability.

natural unit: Application-specific alternative to **absolute** or **relative time** to quantify the usage frequency of a software-based product, for example, sensor interrupts, runs, user selection of menu items, pages of output, transactions, telephone calls, SMS messages, jobs, queries, or API calls.

operational environment: A physical context, setting, and circumstances used to support the in-service operation of a system.

NOTE—Includes direct factors (developmental, technological, business, operational, and organizational) and in some cases indirect factors (political, ethical, economic, legal, regulatory, ecological, and social) that may have a material effect on the dependability of a system.

perfective support: In-service modification of a system to provide enhancements. (Adapted from IESO/IEC/IEEE 14764:2022 [B21])

preventive support: In-service actions undertaken to correct latent faults or vulnerabilities before they are manifested as failures, to replenish or update resources, or to restore initial operating conditions to prevent software aging failures. (Adapted from IESO/IEC/IEEE 14764:2022 [B21])

recoverability: The ability of a system to mitigate the duration and lossage of impairments; the extent to which a system can self-correct after operating at a degraded level acceptable to stakeholders or dependent systems. *Syn:* resilience.

NOTE—As a quantitative component of dependability, recoverability and its elements can be expressed as a requirement parameter (threshold), as a probability (a prediction or estimate), or as a measurement obtained during test or in-service operation (observation).

relative time: A tick from a source that increments it according to a reference activity other than absolute time, such as the number of CPU seconds executing an application process in a multi-core virtualized environment, or the customary business hours and days in a certain locale. *Compare:* **absolute time**, **natural unit**.

release package: A collection of configuration items and other artifacts necessary to install or update a software system.

NOTE—Typically includes application software media, documentation, utilities, configuration files, licensing materials, and installation utilities. These items are typically associated with a unique configuration identifier or version number of the SOI and can be further identified by a license key or other identifier of the user or target environment.

reliability: The ability of a system to operate without countable failures in a specified environment for a specified time span; its failure intensity.

NOTE—As a quantitative component of dependability, reliability can be expressed as a requirement parameter (threshold), as a probability (a failure rate prediction or estimate), or as a failure intensity measurement obtained during test or in-service operation (observation). See 5.1.2.7.

reliability class: A category for subsetting system requirements and behaviors according to 1) the canonical quality characteristics of safety, security, functionality, performance, or utilization, 2) the universal reliability class (no subset of capability types), or 3) one or more system-specific reliability classes.

NOTE—A reliability class is the basis for specifying, observing, and evaluating the failure intensity of that class. See 5.1.2.

repair: A corrective, emergency, or preventative support action undertaken to mitigate a system fault or a failure.

requirement: A condition or capability that is necessary in a system, system component, product, or service to satisfy an agreement, standard, specification, or other formally imposed documents.

revealed requirement: A requirement that is derived when a system produces an anomaly or behavior that no explicit requirement prohibits or allows.

NOTE—An anomaly or behavior can be judged to be acceptable or detrimental, resulting in a revealed requirement to allow or prohibit the behavior. A revealed requirement is considered to have the same standing as an explicit requirement.

run: The entire activation, execution, and shut-down cycle of an SOI, however simple or complex its runtime architecture may be.

NOTE 1—When an SOI is installed at multiple sites, each execution cycle at each site constitutes a distinct run.

NOTE 2—The use of “run” herein does not imply any limitation to batch or transactional execution technologies. The specific operational actions, system architecture, and software activation strategies of a run are SOI-specific.

runtime environment: A specific instance of a configuration of hardware and software, as provisioned for a specific operational environment, on which an installed SOI runs.

NOTE—A runtime environment includes executable application software, data stores, utility programs, locally hosted services, operating system(s), network interfaces, and physical computer systems necessary for SOI execution, but not the SOI itself.

safety: The expectation that a system does not, under defined conditions, lead to a state in which human life, health, property, or the environment is endangered. (ISO/IEC/IEEE 24765)

safety failure: System behaviors or their absence such that one or more requirements of the safety reliability class are not met.

security: Protection against intentional subversion or forced failure, containing a composite of four attributes: confidentiality, integrity, availability, and accountability, plus aspects of a fifth, usability, all of which have the related issue of their assurance. (ISO/IEC/IEEE 24765)

security failure: System behaviors or their absence such that one or more requirements of the security reliability class are not met.

severity: The relative detrimental effect of a failure.

slice: A subset of a downtime span.

software maintenance: Activities undertaken to make material alterations to a software system, such as adding, changing, or deleting source code and regenerating executable files for a release package. *Contrast:* **software support**.

NOTE 1—Software maintenance refers to activities that require skills, facilities, and other resources available only in a system's development environment, that is, not required of, expected, or routinely found in operational environments.

NOTE 2—Maintenance actions are typically conducted in parallel with in-service operation, in which case the duration of and effort expended on maintenance actions is not counted as downtime.

software reliability: The extent to which a system has operated without countable failures in a specified environment for a specified time span. *See also:* **failure intensity, failure rate**.

software reliability engineering (SRE): (A) application of statistical techniques to data collected during system development and operation to specify, estimate, or assess the reliability of software-based systems. (B) application of software reliability best practices to enhance software reliability characteristics of software being developed and integrated into a system (IEEE Std 1633-2016 [B15]).

software safety: The ability of software to resist failure and malfunctions that can lead to death or serious injury to people, loss or severe damage to property, or severe environmental harm.

software system: A system for which software is of primary importance to its stakeholders.

NOTE 1—In the most general case, a software system is comprised of hardware, software, people, and manual procedures.

NOTE 2—In a software system, software is essential for meeting the preponderance of system requirements.

software support: In-service activities undertaken to establish, preserve, amend, or remove conditions and resources necessary for the intended use of a system in an operational environment. *Contrast:* **software maintenance**.

NOTE—Support actions are limited to tasks that can be performed with skills, tooling, and other resources available in an operational environment.

span: An in-service interval whose start and end ticks are measurable.

subsetting: The definition or creation of a data subset from that includes some but not all data from its sources.

supportability: The frequency, scope, and consistency of support actions to establish, preserve, and revise conditions and resources necessary for ongoing use of an in-service system.

NOTE—As a quantitative component of dependability, supportability can be expressed as a requirement parameter (threshold), as a probability (a prediction or estimate), or as a measurement obtained during test or in-service operation (observation).

system of interest (SOI): A specific system that is the focus of some effort, such as a development or maintenance project, in-service support action(s), a research study, a performance benchmark, an economic analysis, or a security assessment.

NOTE—Successive versions of an SOI are typically identified as distinct configuration items.

tick: A point in absolute time or relative time, or an occurrence of an event counted as a natural unit.

trustworthiness: A quality characteristic that indicates the extent to which persons believe a system is dependable.

user-defined reliability class: A documented collection of certain kinds of system behaviors whose failures are of sufficient interest to warrant specification, observation, and evaluation apart from the canonical reliability classes of safety, security, functionality, performance, and utilization.

Example—Artificial intelligence hallucinations.

3.2 Acronyms and abbreviations

Aa	achieved availability
Ai	inherent availability
AI	abandon installation
Ao	operational availability
API	application programming interface
AVT	availability threshold
CAPA	corrective and preventative actions
CF	catastrophic failure
CI	configuration item
CV	coefficient of variation
DF	data failure

DTCF	downtime cycle finish
DTCS	downtime cycle start
EF	environmental failure
FIO	failure intensity objective
ISP	in-service period
ISRM	in-service reference model
KPI	key performance indicator
MAO	mean operational availability
MIL	mean impairment lossage
MOE	measure of effectiveness
MOP	measure of performance
MOS	measure of suitability
MDT	mean downtime
MLDT	mean logistic downtime
MTBF	mean time between failures
MTBS	mean time between support
MTBSF	mean time between software failures
MTIR	mean time to impairment recovery
MTTF	mean time to failure
MTTR	mean time to repair
MT _x	mean time to <i>x</i>
NRR	normal run response
NRT	normal run termination
NSR	normal support response
NTF	non-terminal failure
OF	operational failure
PSB	preventive support budget
RC	recovery complete
RFR	recovery failure intensity
ROCOF	rate of occurrence of failures
RPO	recovery point objective
RS	run start
RSR	recovery success rate
RTO	recovery time objective
SCF	support cycle finish
SCS	support cycle start