INTERNATIONAL STANDARD

ISO/IEC/IEEE 32675

First edition
2022-08

Information technology — DevOps — Building reliable and secure systems including application build, package and deployment

Technologies de l'information — DevOps — Création de systèmes fiables et sûrs notamment en matière de compilation, paquetage et déploiement d'applications

Reference number
ISO/IEC/IEEE 32675:2022(E)

Documents
Document Sharing Hub
© IEEE 2021

Full document details: ISO/IEC/IEEE 32675-2022

COPYRIGHT PROTECTED DOCUMENT

© IEEE 2021

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from IEEE at the address below.

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

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

© IEEE 2021 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO/IEC documents should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

ISO/IEC/IEEE 32675 was prepared by the Systems and Software Engineering Standards Committee of the IEEE Computer Society (as IEEE Std 2675-2021) and drafted in accordance with its editorial rules. It was adopted, under the "fast-track procedure" defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© IEEE 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO/IEC/IEEE 32675-2022

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

IEEE Std 2675™-2021

IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment

Developed by the

Software & Systems Engineering Standards Committee
of the
IEEE Computer Society

Approved 9 February 2021

IEEE SA Standards Board

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

Abstract: Technical principles and processes to build, package, and deploy systems and applications in a reliable and secure way are specified. Establishing effective compliance and information technology (IT) controls is the focus. DevOps principles presented include mission first, customer focus, left-shift, continuous everything, and systems thinking. How stakeholders, including developers and operations staff, can collaborate and communicate effectively is described. The process outcomes and activities herein are aligned with the process model specified in ISO/IEC/IEEE 12207:2017 and ISO/IEC/IEEE 15288:2015.

Keywords: agile, continuous delivery, continuous deployment, continuous integration, DevOps, IEEE 2675™, left-shift

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

Copyright © 2021 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 16 April 2021. 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 978-1-5044-7407-8 STD24616
Print: ISBN 978-1-5044-7408-5 STDPD24616

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.

Copyright©2021IEEE.Allrightsreserved

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

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 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 the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. 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 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 does not warrant or represent the accuracy or completeness of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to results and workmanlike effort. 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 his or her 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 development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Copyright © 2021 IEEE. All rights reserved

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual 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, nor be relied upon as, a formal position of IEEE. 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.

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 Standards Coordinating Committees 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 in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system. An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.

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 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, IEEE does not waive any rights in copyright to the documents.

Copyright © 2021 IEEE. All rights reserved

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

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. 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. 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.

Patents

IEEE Standards are developed in compliance with the IEEE SA Patent Policy.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. 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, and interference protection practices and all applicable laws and regulations.

Copyright © 2021 IEEE. All rights reserved

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

Participants

At the time this IEEE standard was completed, the DevOps Working Group had the following membership:

Bob Aiello, *Chair*
Lynn Robert Carter, *Secretary*

Devora Aiello
Sarah Baker
Jaynee Beach
Kristian Beckers
Subroto Bhattacharya
Paul Bruce
Simon Goldsmith
Victoria Hailey
Bob Jenkins
Daniel Katzman
Ruth Lennon
Nithyanandam Mathiyazhagan
Malu Milan
Harvey Nusz
Martin Radley
Tafline Ramos
Annette Reilly
Ayhan Tek
Mark Underwood
Altaz Valani
Chris Walker
Robert J. White
Steve Woodward
Hasan Yasar

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

Bob Aiello
Johann Amsenga
Bakul Banerjee
William Bearden
Juris Borzovs
Pieter Botman
Lynn Robert Carter
Manuel Castro
Ronald Dean
Teresa Doran
Andrew Fieldsend
Dan Friedman
David Fuschi
Louis Gullo
Jon Hagar
Victoria Hailey
Werner Hoelzl
Piotr Karocki
Daniel Katzman
Ralph Kearfott
Stuart Kerry
Edmund Kienast
Yongbum Kim
Naga Sai Kruthiventi
Thomas Kurihara
Jim Lewis
Johnny Marques
Rajesh Murthy
Nick S. A. Nikjoo
Joanna Olszewska
Beth Pumo
R. K. Rannow
Annette Reilly
Robert Schaaf
Stefan Schlichting
Stephen Schwarm
Subrato Sensharma
Carl Singer
Friedrich Stallinger
Thomas Starai
Walter Struppler
Gerald Stueve
Max Turner
Mark-Rene Uchida
Altaz Valani
John Vergis
Marcel Winandy
Hasan Yasar
Yu Yuan
Oren Yuen
Janusz Zalewski

When the IEEE SA Standards Board approved this standard on 9 February 2021, it had the following membership:

Gary Hoffman, *Chair*
Vacant Position, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Edward A. Addy
Doug Edwards
Ramy Ahmed Fathy
J. Travis Griffith
Thomas Koshy
Joseph L. Koepfinger*
David J. Law
Howard Li
Daozhuang Lin
Kevin Lu
Daleep C. Mohla
Chenhui Niu
Damir Novosel
Annette Reilly
Jon Walter Rosdahl
Dorothy Stanley
Mehmet Ulema
Lei Wang
F. Keith Waters
Karl Weber
Sha Wei
Howard Wolfman
Daidi Zhong

*Member Emeritus

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

Introduction

This introduction is not part of IEEE Std 2675™-2021, IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment.

The complexity of software systems has increased to an unprecedented level. This has led to new opportunities, but also to increased challenges for the organizations that create and utilize systems. One of the greatest challenges has been to address the increased rate of change in modern development methodologies, including agile and even rapid iterative development. These challenges exist throughout the life cycle of a system and at all levels of architectural detail. This document highlights the manner in which DevOps can help address the challenges inherent in accelerated development methodologies and achieve end user goals for increased productivity and quality.

DevOps is an interdisciplinary approach and means to enable the realization of successful software systems. It focuses on defining stakeholder needs and required functionality early in the development cycle, documenting requirements, and performing design synthesis and system validation while considering the complete problem. It integrates the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation and maintenance. It considers both the business and the technical needs of stakeholders with the goal of providing a quality product that meets the needs of users and other applicable stakeholders. This life cycle spans the conception of ideas through to the retirement of a system. It provides the processes for acquiring and supplying systems. It helps improve communication and cooperation among the parties that create, utilize, and manage modern software systems. In addition, this framework provides for the assessment and improvement of the life cycle processes.

This document is appropriate both for organizations that are unused to applying engineering process standards, and for those who have used longstanding standards, who have the goal of implementing effective information technology (IT) controls, embracing and managing risk, while enabling more rapid development (higher velocity). Organizations that are already embracing IEEE standards can find IEEE Std 2675 to be essential in helping them to implement the DevOps transformation. Organizations that choose IEEE Std 2675 as their first industry standard can subsequently apply a broader family of IEEE standards.

This document is closely aligned with the life cycle processes in ISO/IEC/IEEE 12207:2017 [B14][1] and ISO/IEC/IEEE 15288:2015. Configuration management is the basis of DevOps and hence it is also closely aligned with IEEE Std 828™ [B3], along with other related standards.

[1] The numbers in brackets correspond to those of the bibliography in Annex A.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

Contents

1. Overview 9
1.1 Scope 9
1.2 Purpose 10
1.3 Word usage 10

2. Normative references 11

3. Definitions, acronyms, and abbreviations 11
3.1 Definitions 11
3.2 Acronyms and abbreviations 14

4. Conformance 16
4.1 Compliance criteria 16
4.2 Full conformance to outcomes 17
4.3 Full conformance to tasks 17
4.4 Tailored conformance 17

5. DevOps concepts 17
5.1 Value of DevOps 17
5.2 DevOps principles 18
5.3 DevOps and organizational culture 20
5.4 DevOps and life cycle processes 23

6. Relation of software life cycle processes to DevOps 23
6.1 Agreement processes 23
6.2 Organizational Project-Enabling processes 27
6.3 Technical Management processes 38
6.4 Technical processes 61

Annex A (informative) Bibliography 88

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment

1. Overview

1.1 Scope

This document provides requirements and guidance on the implementation of DevOps to define, control, and improve software life cycle processes. It applies within an organization or a project to build, package, and deploy software and systems in a secure and reliable way. This document specifies practices to collaborate and communicate effectively in groups including development, operations, and other key stakeholders.

This document applies a common framework for software life cycle processes, with well-defined terminology. It contains processes, activities, and tasks that are to be applied to the full life cycle of software systems, products, and services, including conception, development, production, utilization, support, and retirement. It also applies to the acquisition and supply of software systems, whether performed internally or externally to an organization. These life cycle processes are accomplished through the involvement of stakeholders, with the ultimate goal of achieving customer satisfaction. The life cycle processes of this document can be applied concurrently, iteratively, and recursively to a software system and incrementally to its elements.

This document applies to software systems, products, and services, and the software portion of any system. Software includes the software portion of firmware. Those aspects of system definition needed to provide the context for software systems, products, and services are included.

There is a wide variety of software systems in terms of their purpose, domain of application, complexity, size, novelty, adaptability, quantities, locations, life spans, and evolution. This document describes the processes that comprise the life cycle of software systems. It therefore applies to one-of-a-kind software systems, software systems for wide commercial or public distribution, and customized, adaptable software systems. It also applies to a complete stand-alone software system and to software systems that are embedded and integrated into larger, more complex, and complete systems.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

1.2 Purpose

The purpose of this standard is to specify required practices for operations, development, and other key stakeholders to collaborate and communicate to deploy systems and applications in a secure and reliable way. This document provides a defined set of processes and methods to facilitate DevOps principles and practices, including improved communication between stakeholders throughout the systems life cycle, not just during development and operations. This document is written for DevOps stakeholders, which includes, but is not limited to, acquirers, suppliers, developers, integrators, operators, maintainers, managers, quality assurance managers, compliance managers, auditors, and users of software systems, products, and services. It can be used by a single organization in a self-imposed mode or in a multi-party situation. Parties can be from the same organization or from different organizations, and the situation can range from an informal agreement to a formal contract.

The processes in this document can be used as a basis for implementing DevOps while establishing organizational environments, e.g., methods, procedures, techniques, tools, and trained personnel. The processes in this document provide guidance on the use of DevOps principles and practices for processes used by an organization to construct software life cycle models appropriate to its products and services. An organization, depending on its purpose, can select and apply an appropriate subset to fulfill that purpose.

This document can be used in one or more of the following modes:

a) By an organization—to establish DevOps principles and practices in support of an environment of desired processes. These processes can be supported by an infrastructure of methods, procedures, techniques, tools, and trained personnel. The organization may then employ this environment to perform and manage its projects and progress software systems through their life cycle stages. In this mode, this document is used to assess conformance of a declared, established environment to its provisions.

b) By a project—to establish DevOps principles and practices to help select, structure, and employ the elements of an established environment to provide products and services. In this mode, this document is used in the assessment of conformance of the project to the declared and established environment.

c) By an acquirer and a supplier—to establish DevOps principles and practices to help develop an agreement concerning processes and activities. Via the agreement, the processes and activities in this document are selected, negotiated, agreed to, and performed. The acquirer and supplier can be part of the same organization or separate organizations.

d) By process assessors—to establish DevOps principles and practices in a process reference model for use in the performance of process assessments that may be used to support organizational process improvement.

1.3 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*).[2, 3]

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*).

[2] The use of the word *must* is deprecated and cannot be used when stating mandatory requirements, *must* is used only to describe unavoidable situations.
[3] The use of *will* is deprecated and cannot be used when stating mandatory requirements, *will* is only used in statements of fact.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

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*).

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.

This document has no normative references.

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.[4]

For additional terms and definitions in the field of systems and software engineering, see ISO/IEC/IEEE 24765 [B20], which is published periodically as a "snapshot" of the SEVOCAB (Systems and software Engineering Vocabulary) database and is publicly accessible at computer.org/sevocab.

NOTE—While the aim is to provide consistency in terminology throughout the IEEE standards, it is worth noting that, particularly from the DevOps perspective, there are often alternative terms for similar roles or processes. The applicability of terms to development, operations, testing, security, and performance was separately considered so that the terminology used was applicable in every case.[5]

aligned: Group agreement and alliance to one or more shared objectives.

NOTE—Key concepts are that each member understands critical inputs (i.e., information, context, and constraints), acts according to a plan that is communicated to all members, accepts responsibility for their part in requisite activities and tasks, and harmoniously collaborates with other members and external resources.

archive: Location of system elements that are no longer present in runtime environments, but are available for examination for audit, regulatory, and other processes.

audit: Independent, continuous examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria for the purpose of providing assurance against risk.

NOTE 1—Generating evidence of information technology (IT) controls that support audit is often automated where practical.

[4] *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.
[5] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

NOTE 2—Audit can be orchestrated and integrated as part of a DevOps pipeline.

build: Process of generating an executable and testable system from source versions or baselines. (IEEE Std 828™.)

business value: Strategic priorities set forth by the business as it relates to revenue, cost, risk, security, privacy, ethics, and compliance.

competency: Ability to demonstrate and apply the combination of knowledge, formal and informal skills, training, experience, and behavioral attributes to achieve intended organizational and technical results.

compliance: Continual fulfillment to internal and external requirements, rules, and regulations. (SEVOCAB.)

configuration management: Technical and organizational activities, comprising configuration identification, control, status accounting, and auditing.

continuous delivery: Software engineering practices that allow for frequent releases of new systems (including software) to staging or various test environments through the use of automated tools.

continuous deployment: Automated process of deploying changes to production by verifying intended features and validations to reduce risk.

continuous integration: Technique that continually merges artifacts, including source code updates from all developers on a team, into a shared mainline to build and test the developed system.

continuous risk management: Continuous process, which may be automated, that identifies, applies, and monitors controls to treat risks for a planned activity, project, or program, to achieve a desired outcome.

cryptographic hash: Method to verify the authenticity of a system element or software via the production of a checksum.

data-driven: Informing an activity by evidence.

defect: Imperfection or deficiency in a work product or characteristic that does not meet its requirements or specifications.

deployment: Stage of a life cycle in which a system is put into operation and transition issues are resolved.

DevOps: Set of principles and practices which enable better communication and collaboration between relevant stakeholders for the purpose of specifying, developing, and operating software and systems products and services, and continuous improvements in all aspects of the life cycle.

disposal: Removal or archiving, but not deletion of an artifact, so it can be made available for traceability and auditability.

error: Discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. (ISO/IEC/IEEE 15026-1:2019 [B15], 3.4.5.)

infrastructure: Facilities such as power, cooling, and physical security of the data center, networking, hardware, and software needed to support the systems life cycle and maintain information technology (IT) services.

NOTE—Does not include the associated people, processes, or documentation. In DevOps, software-defined infrastructure enables elasticity.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

infrastructure as code (IaC): Definition, management, and provision of infrastructure components using software.

NOTE—In DevOps, infrastructure as code facilitates the automation of the systems life cycle enabling consistency, performance, and security across the system and resources.

knowledge management: Multi-disciplinary process of obtaining, preserving, sharing, using, and refreshing knowledge.

NOTE—In DevOps, knowledge management guides and facilitates the automation of system operations, system problem identification and remediation, and reporting system health.

left-shift: Prioritizing the involvement of relevant stakeholders in applying quality activities, security, privacy, performance, verification, and validation earlier in the life cycle.

life cycle: Evolution of a system, product, service, project, or other human-made entity from conception through retirement.

NOTE—In DevOps, the systems life cycle is supported by automated elements that produce meaningful and actionable audit logs.

life cycle model: Framework of processes and activities concerned with the life cycle, which can be organized into stages, acting as a common reference for communication and understanding.

machine readable: Pertaining to data in a form that can be automatically generated by and input to a computer.

minimum viable product: Version of a work product with just enough features and requirements to satisfy early customers and provide feedback for future development.

monitoring: Determining the status of a system, a process, or an activity. (ISO/IEC 19770-1:2017, 3.35.)

package: To combine related components into a single, deployable item.

platform as a service (PaaS): Provision of a complete environment of IT resources, such as programming languages, libraries, services, and tools supported by the service provider.

NOTE—The level of control over the service provided to the customer can vary with the service provider. In DevOps, PaaS is automated as part of the DevOps pipeline.

pipeline: Software or hardware design technique in which the output of one process serves as input to a second, the output of the second process serves as input to a third, and so on, often with simultaneity within a single cycle time.

policy: Intentions and direction of an organization, as formally expressed by its top management. (ISO/IEC 19770-1:2017, 3.43.)

NOTE—Direction includes mandates set by the organization.

portfolio: Projects, programs, and operations managed as a group to achieve business objectives.

portfolio management: Centralized management of one or more portfolios to achieve business objectives.

problem: Difficulty or uncertainty experienced by one or more persons, resulting from an unsatisfactory encounter with a system in use.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

quality assurance: Part of quality management focused on continually providing confidence that requirements are being fulfilled.

role-playing session: Technique of engagement including the relevant stakeholders in an interactive simulation of the dynamic interactions that are part of the system design.

software as a service (SaaS): Access to resources from client devices through thin client interface or program interface.

NOTE—The level of control over the resource provided to the consumer can vary with the service provider. In DevOps, SaaS can be automated as part of the DevOps pipeline.

software life cycle: Project-specific sequence of activities that is created by mapping the activities of a standard onto a selected software life cycle model. (SEVOCAB.)

stage: Period within the life cycle of an entity that relates to the state of its description or realization. (SEVOCAB.)

stakeholder: Individual, group, or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations. For example, legal, financial, risk, compliance, security, privacy, and end user organizations; supporters, developers, producers, trainers, implementers, maintainers, operations, disposers, acquirers, supplier organizations, and regulatory bodies.

NOTE—Some stakeholders can have interests that oppose each other or oppose the system.

validation: Confirmation in a timely manner, through automated techniques where possible, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled.

NOTE—A system is able to accomplish its intended use, goals, and objectives (i.e., meet stakeholder requirements) in the intended operational environment. The right system is operating to meet business objectives.

velocity: The rate of current work unit completion, measured as work units completed per fixed time period, such as story points, delivered features, functions, function points, user stories, use cases, or requirements completed in a given time period.

NOTE—Used as a measure of burndown rate or burnup rate.

verification: Confirmation in a timely manner, using automated techniques where possible, through the provision of objective evidence, that specified requirements have been fulfilled. (ISO 9000:2015.)

NOTE—Verification is a set of activities that compares a system or system element against the required characteristics. This includes, but is not limited to, specified requirements, design, descriptions, and the system itself. The system is operating as per business objectives.

3.2 Acronyms and abbreviations

ALM application life cycle management

CD continuous delivery, continuous deployment

CI configuration item, continuous integration

CM configuration management

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

COTS	commercial-off-the-shelf
DRP	disaster recovery plan
ETL	extract, transform, load
FCA	functional configuration audit
IaaS	infrastructure as a service
IaC	infrastructure as code
KPI	key performance indicator
MTTD	mean time to detection
MTTR	mean time to recovery
OLA	operations level agreement
OSS	open source software
QA	quality assurance
QM	quality management
QoS	quality of service
PaaS	platform as a service
SaaS	software as a service
SHA	secure hash algorithm
SLA	service level agreement
SLI	service level indicator
SLO	service level objective
SME	subject matter expert
SOI	system-of-interest
SOP	standard operating procedure
SoS	system of systems
TDD	test-driven development
V&V	validation and verification

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

4. Conformance

4.1 Compliance criteria

This document provides requirements for a number of processes suitable for usage during the life cycle of a software service, system, or product. Requirements stated for an organization may be applied to any size of organization, program, project, or entity.

Particular projects or organizations may not need to use all of the processes provided by this document. Therefore, implementation of this document typically involves selecting and declaring a set of processes suitable to the organization or project. Each process has a set of objectives (phrased as "outcomes") and a set of activities and tasks that represent one way to achieve the objectives.

Options for conformance are provided for needed flexibility in the application of this document. There are two criteria for claiming full conformance. Achieving either criterion suffices for conformance, and the chosen criterion (or criteria) shall be stated in the claim.

— Claiming "full conformance to outcomes" asserts that all of the required outcomes of the declared set of processes are achieved.

— Alternatively, claiming "full conformance to tasks" asserts that all of the requirements of the activities and tasks of the declared set of processes are achieved.

Users who implement the activities and tasks of the declared set of processes can assert full conformance to tasks of the selected processes. Some users, however, might have innovative process variants that achieve the objectives (i.e., the outcomes) of the declared set of processes without implementing all of the activities and tasks. These users can assert full conformance to the outcomes of the declared set of processes. The two criteria—conformance to task and conformance to outcome—are necessarily not equivalent since specific performance of activities and tasks can require, in some cases, a higher level of capability than just the achievement of outcomes.

NOTE 1—When this document is used to help develop an agreement between an acquirer and a supplier, clauses of this document can be selected for incorporation in the agreement with or without modification. In this case, it is more appropriate for the acquirer and supplier to claim compliance with the agreement than conformance with this document.

NOTE 2—An organization (for example, nation, industrial association, company) imposing this document as a condition of trade, can specify and make public the minimum set of required processes, outcomes, activities, and tasks which constitute suppliers' compliance with the conditions of trade.

Requirements of this document are marked by the use of the verb *shall*. Recommendations are marked by the use of the verb *should*. Permissions are marked by the use of the verb *may*. However, despite the verb that is used, the requirements for conformance are selected as described previously.

Demonstrable evidence may be represented as, for example, documented standard operating procedures (SOPs) or work practices, workflows showing process checks, quality metrics, or in-process review steps that are formally documented as part of the configuration management process. The specificity of the demonstrable evidence will vary with the business domain, organizational policies, and other factors. In DevOps, demonstrable evidence may be collected, where possible in real time and in an automated fashion, to verify conformance to a requirement.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

4.2 Full conformance to outcomes

A claim of full conformance declares the set of processes for which conformance is claimed. Full conformance to outcomes is achieved by demonstrating that all of the outcomes of the declared set of processes have been achieved. In this situation, the provisions for activities and tasks of the declared set of processes are guidance rather than requirements, regardless of the verb form that is used in the provision. Full conformance to outcomes permits greater freedom in the implementation of conforming processes and can be useful for implementing processes to be used in the context of an innovative life cycle model.

One intended use of this document is to facilitate process assessment and improvement. For this purpose, the objectives of each process are written in the form of "outcomes" compatible with the provisions of ISO/IEC/IEEE 24774 [B21].

NOTE—ISO/IEC 33002 [B12] provides for the assessment of processes, which can be a basis for process improvement. Users intending process assessment and improvement can use the process outcomes written in this document as the process reference model required by ISO/IEC 33002.

4.3 Full conformance to tasks

A claim of full conformance declares the set of processes for which conformance is claimed. Full conformance to tasks is achieved by demonstrating that all of the requirements of the activities and tasks of the declared set of processes have been achieved. In this situation, the provisions for the outcomes of the declared set of processes are guidance rather than requirements, regardless of the verb form that is used in the provision.

NOTE—A claim of full conformance to tasks can be appropriate in contractual situations where an acquirer or a regulator requires detailed understanding of the suppliers' processes.

4.4 Tailored conformance

When this document is used as a basis for establishing a set of processes that do not qualify for full conformance, the clauses of this document are selected or modified. The tailored text, for which tailored conformance is claimed, is declared. Tailored conformance is achieved by demonstrating that the outcomes, activities, and tasks, as tailored, have been achieved. Small to medium enterprises can use tailored conformance as a technique to begin the process toward obtaining full conformance.

NOTE—Annex A of ISO/IEC/IEEE 12207:2017 [B14] provides requirements and guidance for tailoring a process.

5. DevOps concepts

5.1 Value of DevOps

This document is intended to demonstrate how DevOps is appropriate for building, packaging, and deploying reliable and secure systems. This document presents a more holistic view of DevOps than its literal meaning of combining development and operations resources and procedures to perform continuous integration, delivery, and deployment.

The term *DevOps* evolved from the availability of fully automated application build, package, and deployment tools, along with the recognition that information technology (IT) organizations were not

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

prepared to use those tools effectively. Developers and operations professionals traditionally had different (and, at times, conflicting) perspectives. An extreme, oversimplified characterization is that developers want to go fast and are mostly concerned with velocity (delivering value to end users as quickly as possible), while the operations team is concerned with assuring reliability and enforcing cybersecurity. In that sense, DevOps can appear to ignore the traditional safeguards of standard life cycle processes, such as separation of duties, multiple levels of reviews, and independently tested and infrequently scheduled releases. From that perspective, DevOps can appear to be unsuitable for organizations concerned with security and risk to critical applications.

In practice, DevOps aims to satisfy a dynamic and competitive marketplace that favors products that balance the V requirements (volume, velocity, variety, veracity, value, and others). DevOps seeks to achieve a balance between velocity and system reliability and stability. DevOps was created to provide solutions to constantly changing complex problems, where reducing organizational risk and improving security and reliability are critical requirements. The diverse international domains where reliable and secure systems are used often have rigorous regulatory and legal requirements. Therefore, DevOps requires identifying, engaging, and dynamically collaborating with a set of supplier and acquirer (producer and consumer) entities forming a holistic "ecosystem" of both internal and external organizations and stakeholders (e.g., customers, business units, support organizations, suppliers, collaborators, and partners).

This clause presents key concepts of DevOps for reliable and secure systems: its main principles and practices, its changes to organizational culture, and its relationship to traditional life cycle processes.

5.2 DevOps principles

5.2.1 Business or mission first

DevOps focuses on business and organizational goals ahead of procedural and technical considerations. DevOps utilizes information-rich feedback loops to understand progress and threats to attaining business and mission goals. Taking a business or mission first view helps to balance the concerns of risk and the activities which provide the most value to the customer. Continuously finding a dynamic balance between opportunities and risks and applying risk assessment and treatment at the organizational level enables the realization of DevOps without thrashing and wasted resources. Realizing the full promise of DevOps requires maintaining a strategic balance between honoring today's commitments while enabling the organization to survive risks and develop the capabilities to move forward.

5.2.2 Customer focus

DevOps takes a customer-centric view, prioritizing and designing work to deliver value to the customer, as well as identifying and managing risk. In short, if it makes sense for the customer and meets a customer need, then it is likely to be the right approach from a DevOps perspective. For example, for both customers and suppliers, privacy is a customer focus: not only protection of individuals' data, but also protection of enterprise data. Privacy may include data categorization (e.g., health data, financial data) and classification-driven methods (e.g., confidential, restricted, internal use, public) as often required by regulatory and legal requirements.

DevOps relies on keeping stakeholders informed and aware of changes that can impact them, by means of automation when practicable. To reduce information overload and to allow stakeholders to focus on those changes that exceed their individual risk thresholds, typical tools include automated support to inform involved stakeholders when an issue is approaching their individual threshold and when actions are initiated if a threshold is crossed. Especially when systems have multiple stakeholders, collaboration mechanisms to agree on decisions and resolve conflicting concerns in real time should be designed and

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

established before issues occur. Figuring out what to do during a crisis is much more difficult when those involved are overloaded and under significant stress.

Example: An existential financial risk to one stakeholder can be a life-critical risk to another.

5.2.3 Left-shift and continuous everything

The normal DevOps practice is information-driven, risk-based, continuous everything. DevOps continuous everything means using the same practices in development as in operations and sustainment. DevOps practices are founded on automation for continuous integration, delivery and deployment, and operations and sustainment. The approach to DevOps in this document is to build systems to be secure and verifiable from the very beginning. Risk management, quality assurance (QA), and testing are the practices that make accelerated velocity and continuous delivery possible.

The continuous delivery, testing, and QA practices are shifted left (earlier in the workflow) to be planned and executed at the same time as design and development. For example, in DevOps, automated tests are built along with the product from inception. Most efforts begin with effective reviews of requirements, test strategies, and coding standards. Common methods include test-driven development (TDD), automatic code scanning (on build), automated regression testing, and functional and non-functional (e.g., performance) testing. Continuous QA and testing are essential in any DevOps-centric effort. Reducing rework and waste contributes to the achievement of improved velocities, a hallmark of well-implemented DevOps.

Similarly, information security cannot be tacked on to the end of a development effort. The DevOps view of security is sometimes referred to as DevSecOps, but in reality, there is no DevOps without a continuous focus on security. This includes building systems to be secure from the very beginning of the systems and applications life cycle and continuing throughout their life cycle, including code that is deemed ready to be safely deprecated.

Left-shift is particularly valuable for improving the reliability of methods for production software release and deployment. DevOps transforms a common practice in which developers fully control sandbox, development, and initial test environments, with more rigorous release procedures applied to production. This approach can cause delays and integration and testing issues if toolsets, training, support, and procedures for deployment to production are different than those used in the lower environments. In DevOps, “left-shifting” of deployment procedures means lower-risk deployment using the same methods for all environments in the continuous delivery pipeline.

To accomplish continuous delivery more securely and reliably, many firms have turned away from traditional monolithic, sequential, and mostly manual development and operational approaches to one that integrates an ever-growing number of market-proven external solution components (i.e., frameworks, libraries, application programming interfaces [APIs], and software as service solutions) with a more manageable set of targeted custom solution components. There has also been a significant shift to cloud-based hosting that can easily and efficiently scale up and down as needed to satisfy the dynamic load demands of users. To enable this, DevOps requires the use of tailored processes and specialized pipeline tools able to leverage automation wherever practicable, across the entire system’s life cycle.

5.2.4 Systems thinking

Systems thinking counters a myopic approach of utilizing specialists—such as networking professionals, database administrators, and systems administrators—who rarely communicate with either the development or operations teams and lack understanding of the system as a whole. In DevOps, taking a comprehensive view encourages technology professionals to fully understand the system from end to end. Systems thinking can enable resolution of complex and emergent problems that are not easily traceable to a single

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

flaw. Systems thinking should apply to a consistent architecture for the enterprise tools used for DevOps as well as to the system under development.

DevOps systems and their interfacing systems, constituent systems, and enabling systems, including human participants, form complex ecosystems. Systems thinking in DevOps involves integrating the widely differing perspectives and notions of value and priorities across the stakeholders. Consistent systems thinking among DevOps stakeholders can be challenging when there are differing or even incompatible philosophies, policies, procedures, activities, tasks, and tooling.

Efforts to bolt on security, reliability, and privacy solutions to existing operational systems seldom work. Establishing procedures and approaches for these areas should be addressed as early as possible in the systems life cycle to benefit from the active involvement of all stakeholders. To achieve this requires the creation of shared vocabularies, models, and processes so that the impact and consequences of changes can be quickly and effectively assessed, evaluated, and used to drive decisions and implement treatments. Communication and collaboration across stakeholders are critical components of an overarching, ecosystem-wide risk management process.

Systems thinking is especially valuable when deployment, operations, and post-deployment support activities are transitioned from one organization to another (e.g., internationally deployed systems and deep space missions).

5.3 DevOps and organizational culture

5.3.1 Leadership

While DevOps involves cooperation at all levels of an organization, it is most successful when leadership is viewed internally and externally as fully supporting the letter and the spirit of the DevOps policies, principles, and practices. When those in leadership roles exert their authority and influence, the entire DevOps team is better able to interact with stakeholders; uphold the power of organizational procedures; sustain a sophisticated, complex, and effective pipeline; and engage capable human resources.

Addressing DevOps vulnerabilities requires commitment to investments to establish, sustain, and improve capabilities so that the ecosystem remains aligned with evolving policies, processes, mechanisms, practices, and tools. Without clear and lasting support from and continued active engagement of leadership, the benefits of investments in procedures and tools often fade and disappear. It takes time, the long-term leadership commitment of considerable resources, and a level of dedication to fully deploy and realize the continuous improvement of the DevOps principles, practices, and processes described in this document.

DevOps leadership benefits from the use of transparent, information-driven, risk-based decision-making at all levels. To achieve velocity and other quality goals for DevOps, proper utilization of proven forms of automation depends on the involvement of humans receiving and acting on statistically valid information. These mechanisms and processes are particularly challenging to establish, utilize, sustain, and improve due to the differing baseline cultures of the stakeholders. Proactive and engaged leadership can set the example for stakeholders in recognizing the value and applicability of shared data-driven decisions. When the analysis of information shows statistically significant linkages, use of automation can detect and resolve issues within carefully established bounds. This proactive approach can free leaders and other key stakeholders to focus on innovative and strategic opportunities and risks as well as investments to stay ahead of the competition and, if at all possible, the customers.

Leadership benefits from clear and timely procedures for escalation and dynamic information-driven risk management and issue resolution. Decision-making at all levels empowers the team and the organizational leaders with the capability to handle emerging opportunities and risks inherent in the DevOps ecosystem. This approach helps avoid the overload and burnout of leadership and other personnel by involving,

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

empowering, and authorizing the right people with the right capabilities at the right time with the right information to do the right thing.

5.3.2 Organizational structure and dynamics

This document is intended for use by organizations of any size, particularly for organizations that have requirements for regulatory and audit compliance, such as segregation of duties.

For some very small entities (e.g., fewer than ten employees), it can be possible to operate without formal organizational structures. The same team "owns the code" from design creation through to daily operations, and a DevOps approach is inherent. As the system grows in complexity, however, new frameworks, libraries, APIs, automated tools, and more sophisticated processes can be utilized to enhance the organization's capabilities. As a result, new stakeholders are added and become key to system success. The small organization becomes part of a much more complex ecosystem, even though the number of developers in the organization has not increased.

The dynamics of an organization impact the approach to a successful DevOps transformation. For example, startups and small (rapidly growing) financial services firms often have very different dynamics compared to large financial services such as international banks and global enterprise organizations. The organizational dynamics of medical and pharmaceutical companies are also quite different than those found in a government contractor. The approach to a successful DevOps transformation should be aligned to the organizational dynamics, internal and external competitive forces, and constraints on the organization.

DevOps is as much a cultural approach as it is a change in specific organizational and technical procedures. To achieve the quality and velocity goals that make DevOps attractive, people with relevant knowledge, skills, and abilities need to have the information, tools, resources, and authority to do what is required. Use of DevOps seeks to eliminate the negative impact of organizational silos while maintaining organizational procedures that are required for compliance. DevOps can be practiced using many organizational structures where there is a commitment to collaboration and continuous improvement, such as continuous workforce training, skill acquisition, capability development, and career progression. The commitment of the organizational leaders is particularly valuable to enable and sustain this cultural approach.

DevOps serves teams that work at a consistent sustainable pace with consideration for work-life balance. This continuous workflow contrasts to the expectation that leading up to a release, developers and operations staff have to work long hours and make extraordinary efforts to achieve difficult—even impossible—goals. While the "work-hard/play-hard" culture still exists, there is a growing recognition that stressed and exhausted employees make mistakes, which the team once again goes through heroic efforts to solve. In contrast, DevOps works with many smaller, simpler, and fully automated deployments which can be executed in the normal workday with no perceptible impact to the end user.

5.3.3 Effective communication and collaboration

While internal and external forces can have a tremendous impact on any organization, the way in which its members interact is even more consequential. DevOps values effective and collaborative communication and consequently thrives in organizations where transparency and collaboration are expected and enjoyed. Transparency prioritizes ease of visibility, availability, reachability, and accessibility of information and actions between entities, people, or systems. Transparency enables communications, accountability, and data-driven decision-making.

DevOps is often characterized as a practice to help improve communication and collaboration between stakeholders who have different goals and objectives and may even belong to different internal and external organizations. Developers and operations teams should collaborate, especially when a separate operations team is required to maintain control of the production systems.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

When dealing with new tools supporting "continuous everything," unforeseen issues can arise. DevOps values communication and feedback loops, allowing the organization to experiment with varying approaches. A feedback loop is a virtuous cycle where action produces information which is used in future action to improve results. Team members regularly communicate what is going well and what needs to be improved. DevOps helps implement systems and software life cycle processes with regular checkpoints and rapid course correction when warranted.

5.3.4 Learning organizations and knowledge management

DevOps organizations should embrace iterative or continuous learning, sharing, persisting, and practicing the application of knowledge and best practices. Organizational structures and systems which help to retain, refresh, and leverage technical knowledge and lessons learned are fundamental, as is a culture which enables and encourages its members to share their knowledge and experience. Reflection on the value of lessons learned and their reuse are particularly important when a system or project transitions to other contexts or is handed off to other teams. Knowledge-sharing organizations are most often high-achieving organizations which foster high-performance cross-functional teams. Training, practice, and knowledge sharing for productive and consistent use of enterprise processes toolsets are often key factors in successful enterprise DevOps adoption.

5.3.5 Adaptation, resilience, and organizational change

Central to realizing the benefits of DevOps is successful adaptation of its principles and techniques to the needs of the organization. Adaption is particularly important given the dynamic and competitive nature of DevOps, especially critical in times of societal and domain stresses. The adaptation of DevOps processes to perform at velocity with periods of acceleration is enhanced with experienced-based continuous improvement, a growing reliance on automation, and enlightened risk management. Effective adaptation of DevOps continuously invests in transforming work into automated services that improve velocity, support sporadic acceleration, and enhance other quality attributes, thereby allowing leadership and stakeholders to be more strategic.

While success in a highly competitive domain depends on acceleration and continuously achieving a high level of performance, implementing DevOps involves actually leveraging the available resources with their varying experience levels and skill sets. Exerting extreme efforts whenever mission success, lives, and crucial infrastructures are at risk is not a sustainable tempo in the long term. Ensuring long-term success depends on effectively establishing, utilizing, and improving human resources to gather statistically-meaningful information, analyze the information, and make decisions within mutually-agreed risk tolerances.

Changing an organization to apply DevOps begins with an assessment of current practices to help identify risks and opportunities, and set the direction and priorities for improving practices. Assessment of the existing communications patterns and software development, delivery, operations, and sustainment practices across the ecosystem helps the team to compare as-is and to-be practices and to identify and address gaps. Identifying organizational risks, problems, and issues affecting product development, delivery, operations, and sustainment can act as a catalyst to successful DevOps adoption.

Bringing about organizational change can be a challenging endeavor, especially when external stakeholders need to change their culture or practices. Resilient organizations aim to enable rapid innovation while also minimizing issues due to service interruption, due to the pragmatic cost of quality. Many change agents focus on implementing best practices within their own organizational unit. While it is common to start small—perhaps within one organizational unit sponsoring the DevOps transformation and its ecosystem, it is also essential to consider how these practices may be instantiated consistently throughout the organization. The broader goal is to promote collaboration for systems and processes which enable DevOps across the entire organization.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

Understanding the distinct perspectives of development, operations, and the many other key stakeholders involved in the systems and applications life cycle effort allows DevOps to take hold. Giving equal consideration to other stakeholders, including information security, quality assurance, testing, users, suppliers, and partners can represent a considerable change in organizational dynamics and culture. The involvement of these stakeholders is needed to implement left-shift changes in organizational practice.

5.4 DevOps and life cycle processes

DevOps is sometimes viewed as being mostly focused on systems and applications deployment, and thus applicable mainly at the end of the life cycle. In practice, DevOps is a full life cycle endeavor which gives equal consideration to each stage. DevOps is a set of principles and practices which enable better communication and collaboration between relevant stakeholders for the purpose of specifying, developing, continuously improving, and operating software and systems products and services. It is not just a matter of technical practices affecting other life cycle processes.

Teams using DevOps typically start a systems or applications effort by creating a continuous delivery pipeline (set of tools and procedures) that takes the code from the source code management system and automates the complete application build, package, deployment (including transitions to other environments), operations, and sustainment workflow. Contributors often start with a simple program, write the pipeline, and then iteratively (and rapidly) develop their code. In development, multiple teams often integrate code continuously, automatically deliver the code to a test automation framework, and on to subsequent workflow participants. Transitions to other contexts in the workflow can require special enhancements to the workflow (e.g., simulation-based training or shadow operations) so that quality and velocity are not compromised during and after any transition.

DevOps is suitable for most life cycle process models, and particularly appropriate when teams adopt agile methodologies. DevOps can be just as valuable in an iterative waterfall approach.

Clause 6 of this document details the outcomes and tasks involved in applying DevOps to each of the life cycle processes identified in ISO/IEC/IEEE 12207:2017 [B14] and in ISO/IEC/IEEE 15288:2015 [B16]. The order in which the processes, activities, and tasks are presented in this document does not imply the order in which the processes are to be implemented or performed. The processes can be performed iteratively and concurrently with other processes. Within legal and regulatory constraints (e.g., segregation of duties), the same person or group can perform multiple processes, and a process can be performed by people in different organizations.

6. Relation of software life cycle processes to DevOps

6.1 Agreement processes

This subclause specifies the requirements related to DevOps for agreements with organizational entities external and internal to the organization.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

The Agreement processes consist of the following:

a) Acquisition process—used by organizations for acquiring products or services;

b) Supply process—used by organizations for supplying products or services.

These processes define the activities necessary to establish an agreement between two organizations. If the Acquisition process is invoked, it provides the means for conducting business with a supplier. This may include products that are supplied for use as an operational software system, services in support of operational activities, software elements of a system, or elements of a software system being provided by a supplier. If the Supply process is invoked, it provides the means for an agreement in which the result is a product or service that is provided to the acquirer. (ISO/IEC/IEEE 12207:2017, 6.1.)

NOTE—Throughout Clause 6, boxed text is used to identify direct quotations regarding process groups and process purposes from ISO/IEC/IEEE 12207:2017 [B14].

The Agreement process shall establish a strategy for conducting the acquisition. The Agreement process for secure and reliable DevOps includes a number of outputs beginning with the General Work Statement, which is agreed by the acquirer and the supplier. The agreement includes the more detailed specification referred to as the service level agreement (SLA). The SLA formalizes the agreement between the supplier and the acquirer for the delivery of services within defined criteria and for a specified cost and time-frame.

The Agreement shall include requirements so that the service level agreement and operations level agreement (OLA) work in unison. Organizations can act simultaneously or successively as both acquirers and suppliers of services. The operating level agreement (OLA) shall be put in place to establish the linkages and responsibilities between IT service provider providers whether they are internal or external. The OLA should include software and hardware licensing and maintenance agreement information with any underlying client and vendor contracts. The OLA shall include data to establish, verify, and maintain the agreement for the integrity of the services. The OLA shall establish mechanisms and criteria for the evaluation of the quality, performance, and security of the service. Where the supplier and acquirer are within the same organization, they may still use the Agreement process for consistency throughout all stages of the life cycle.

6.1.1 Acquisition process

6.1.1.1 Purpose of the process

The purpose of the Acquisition process is to obtain a product or service in accordance with the acquirer's requirements. (ISO/IEC/IEEE 12207:2017, 6.1.1.1)

The agreement shall specify supplier responsibility for defining functional and non-functional requirements that have an impact on quality. At the early stages of negotiation, the requirements may be high-level. Agreement wording that reinforces the importance of deriving, reviewing, and testing functional and non-functional requirements is necessary for successful DevOps implementation when the product is delivered. Non-functional requirements such as capacity, security, performance, and maintainability address concerns that become apparent at later stages of implementation when the product is required to fit into the target test and production environments for final shakedown, acceptance, and long-term use by internal or external customers.

Example: The following specifies supplier responsibilities for security: "The supplier shall define capabilities to build in privacy and security by design, meaning any data collected must be securely stored, processed, and use the minimum data necessary to deliver the services and service-related data. It also means that controls to secure the product or service shall be explicitly stated. The Supply process shall define the procedures for gathering and analysis of contractual documents related to information security of the product or service, which can include the service level agreement and the operations level agreement.

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022

The process shall include a description of the most current assurance of the reliability of the service provider's information protection program and the certainty of the security controls supplied by the service provider."

DevOps goals frequently require a higher level of process maturity or capability, as evidenced by automated procedures to meet organizational and stakeholder goals and objectives effectively. Vendors from which acquisitions are being integrated may need to demonstrate their dependability to support DevOps-based processes.

6.1.1.2 Outcomes

As a result of the successful implementation of the Acquisition process, achievement of the following outcomes shall be demonstrable:

a) The risk associated with the proposed supplier is understood.

b) Service level agreements (SLAs), service level objectives (SLOs), and service level measures and indicators are agreed, consistent with DevOps pipelines (capacity).

c) Agreements include completion criteria ("what does 'done' look like") at the complete system level, for each level of work, and for each deliverable.

6.1.1.3 Activities and tasks

The organization shall implement the following activity and tasks in accordance with applicable organization policies and procedures for DevOps with respect to the Acquisition process:

a) Evaluate the capability or maturity of the supplier.
   1) Determine the appropriate models for evaluation.
   2) Review evidence of appropriate process capabilities, such as:
      i) Vendor policies
      ii) Results of penetration testing
      iii) Frequency of applying security patches
      iv) Existing processes and practices
      v) Verification and Validation activities
      vi) User acceptance test plans
      vii) Proof that the software is working as required (functional configuration audit [FCA])
      viii) Test summary reports as evidence of testing completed

b) Align the commonly required service level agreement (SLA) of the acquirers and suppliers, avoiding jeopardizing the dependent providers' and consumers' functional SLA.

6.1.2 Supply process

6.1.2.1 Purpose of the process

The purpose of the Supply process is to provide an acquirer with a product or service that meets agreed requirements. (ISO/IEC/IEEE 12207:2017, 6.1.2.1)

Copyright © 2021 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32675-2022