INTERNATIONAL STANDARD

ISO/IEC/IEEE 26515

Second edition
2018-12

Systems and software engineering — Developing information for users in an agile environment

Ingénierie du logiciel et des systèmes — Développement d'informations pour les utilisateurs dans un environnement agile

Reference number
ISO/IEC/IEEE 26515:2018(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

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

Contents

Page

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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 documents should be noted. This document was drafted in accordance with the rules given in the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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).

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.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in cooperation with the Systems and Software Engineering Committee of the IEEE Computer Society, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

This second edition cancels and replaces the first edition (ISO/IEC/IEEE 26515:2011) which has been technically revised.

The main changes compared to the previous edition are as follows:

— alignment with the widespread use of agile methods to include systems as well as software;

— replacement of the paper-based term “documentation” with the general term “information for users” where appropriate;

— inclusion of agile information development across multiple teams and projects, especially projects in continuous maintenance situations such as DevOps;

— editorial changes;

— new definitions.

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.

Introduction

Software with an application user interface should generally be intuitive for most users or follow common user interface conventions to limit the need for exhaustive or detailed information for users. However, users should be provided with accurate information about how to use the software's functions if questions arise. This information should explain the major features or use cases deliberately created for all types of users. The information should be easily accessible and clearly written to enable quick learning and user proficiency, while reducing help desk support. Hence, well-designed information not only assists the users and helps to reduce the cost of training and support, but also enhances the reputation of the product, its producer, and its suppliers.

Projects that use agile development methods focus on providing rapid and frequent deliveries of high-value software. These methods often involve detailed planning only for the short term and the implementation of processes in parallel, rather than planning for an entire project in distinct phases.

Although agile development methods often advocate less life cycle documentation, the users of a software product still expect and require quality information to be provided with these software products. Although the end results of the process for developing information for users are the same, the methods may be very different in an agile environment.

Agile development methods follow usually short, iterative development cycles based on customer requirements and feedback. In order to fulfil contractual requirements and provide valuable information for users, deliverables for each iteration include information related to the feature set developed during that iteration. The quantity and quality of the information deliverables depend in part on the completeness and maturity of the software features and design after each iteration, specified through user stories, tasks, and personas.

Information developers and other personnel involved in developing information for users should understand the agile development processes and methods used by their organization. This will enable them to tie in seamlessly and provide relevant and useful information for users.

Because of the nature of agile development methods, the traditional means of developing information (both print and onscreen) for end users are not entirely applicable.

This document was developed to assist users of

— ISO/IEC/IEEE 15288, *Systems and software engineering — System life cycle processes*,

— ISO/IEC/IEEE 12207, *Systems and software engineering — Software life cycle processes*,

— ISO/IEC 26514:2008, S*ystems and software engineering — Requirements for designers and developers of user documentation* (also available as IEEE Std 26514-2010, IEEE Standard for Adoption of ISO/IEC 26514:2008, *Systems and Software Engineering — Requirements for Designers and Developers of User documentation*), and

— Other documents in the ISO/IEC/IEEE 265NN family of International Standards.

This document provides requirements and guidance to information developers and related roles on how to adapt the processes described in the ISO/IEC/IEEE 265NN family of International Standards to develop quality information for users.

This document is independent of the agile development methods and tools that are used to produce the software. This document gives an overview of agile methodologies although it neither encourages nor discourages the use of any particular agile methodology. Therefore, this document uses generic agile terminology as much as possible.

Systems and software engineering — Developing information for users in an agile environment

1 Scope

This document supports the interest of information developers and associated roles responsible for producing information for users of software and systems developed within an agile environment. This document takes a process standard approach to specify the way in which information for users can be developed in agile development projects.

This document provides requirements of information management and information development processes appropriate for software projects that are using agile development methods.

Clause 5 covers the overall requirements for information in agile software development.

Clause 6 covers requirements for the information development lead or project manager to plan an agile information development project and manage the information development activities in an agile environment.

Clause 7 covers requirements for designing, developing, and providing information for users in an agile environment.

Annex A describes agile development practices and methods.

This document is intended neither to encourage nor to discourage the use of any particular agile development tools or methods.

This document provides guidance on processes appropriate for information developers of information for users in software and systems projects that are using agile development methodologies. It is not limited to the development phase of the life cycle of information for users, but includes activities throughout the whole life cycle.

This document is intended for use in all organizations that are using agile development or are considering implementing their projects using these techniques. It is assumed that users of this document have experience or general knowledge of information for users (traditionally called “user documentation”) and agile processes.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO, IEC and IEEE maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/

— IEC Electropedia: available at https://www.electropedia.org/

— IEEE Standards Dictionary Online: available at https://ieeexplore.ieee.org/xpls/dictionary.jsp

3.1
agile development
development approach based on *iterative development* (3.11), frequent inspection and adaptation, and incremental deliveries in which requirements and solutions evolve through collaboration in cross-functional teams and through continuous *stakeholder* (3.13) feedback

Note 1 to entry: Any use of the word "agile" in this document refers to methodology. Various agile methods can be found in Annex A.

3.2
agile environment
organizational culture, infrastructure, and methodologies that support *agile development* (3.1)

3.3
agile team
organization or team using *agile development* (3.1) methods and approaches

Note 1 to entry: Typically with roles such as team lead, project manager, user or user representative, software and *information developers* (3.8), and testers.

3.4
backlog
collection of agile *features* (3.7) or stories of both functional and nonfunctional requirements that are typically sorted in an order based on value priority

3.5
done
regarded by the *agile team* (3.3) as complete and ready to use

3.6
epic
major collection of related feature sets broken down into individual *features* (3.7) or *user stories* (3.16) and implemented in parts over a longer period of time

3.7
feature
functional or nonfunctional distinguishing characteristic of a system

Note 1 to entry: Features are considered to add value for the user.

3.8
information developer
person who prepares content for information for users

3.9
information development lead
person who leads the activities of preparing information for users

3.10
iteration
short time frame in which a set of software *features* (3.7) is developed, leading to a working product that can be demonstrated to *stakeholders* (3.13)

Note 1 to entry: Different agile methodologies use different terms for an iteration.

Note 2 to entry: Some agile methodologies are not based on iterations.

3.11
iterative development
repeated use of concurrent planning, developing, and testing activities

3.12
persona
model of a user with defined characteristics, based on research

3.13
stakeholder
individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.44, modified — The example and note to entry have been deleted.]

3.14
stand-up meeting
brief daily project status or planning meeting used in *agile development* (3.1) methodologies

Note 1 to entry: Different agile methodologies use different terms for stand-up meetings.

3.15
use case
description of behavioural requirements of a system and its interaction with a user

Note 1 to entry: A use case describes the users' goal and the requirements including the sequence of interactions between users and the system.

3.16
user story
simple narrative illustrating a user requirement from the perspective of a *persona* (3.12)

4 Conformance

This document may be used as a conformance or a guidance document for projects and organizations claiming conformance to ISO/IEC/IEEE 15288 and/or ISO/IEC/IEEE 12207.

Throughout this document, “shall” is used to express a provision that is binding, “should” to express a recommendation among other possibilities, and “may” to indicate a course of action permissible within the limits of this document. Use of the nomenclature of this document for the features of an agile methodology or the parts of information for users (that is, stand-up meetings, iterations, chapters, topics, pages, screens, windows, etc.) is not required to claim conformance.

Conformance to this document may only be claimed by an organization if all of the requirements in this document can be met by the organization. When conformance is claimed for a multisupplier program, it may be the case that no individual supplier may claim conformance because no single contract calls for all the required activities. Nevertheless, the program, as a whole, may claim conformance if each of the required activities is performed by an identified party.

This document may be included or referenced in contracts or similar agreements when the parties (called the acquirer and the supplier) agree that the supplier shall deliver information for users in accordance with this document. It may also be adopted as an in-house standard by a project or organization that decides to develop information for users in accordance with this document.

Organizations, projects, or multisupplier programs intending to claim tailored conformance should consult ISO/IEC/IEEE 12207:2017, Annex A.

5 Information development process

The basic process phases as described in ISO/IEC 26514, such as analysis, design, development, and review, still apply. However, agile practices eliminate the distinct separation of phases both within a single iteration and across several iterations.

The following agile practices affect the information development process:

a) Development in short iterations replaces long sequential process phases.

b) Information developers try to create accurate and complete information by the end of an iteration. However, a review can lead to the development of additional information in later iterations. Information developers receive training and practice to estimate and complete information for users in time for the release of the iteration. Self-organizing agile teams replace formal roles and contributions.

    i) Software developers, testers, and team leaders may contribute to the analysis, design, development, and review of information for users.

    ii) Information developers may contribute to the design and test of the software as well as to project artifacts and life cycle documents, such as user stories, use cases, and personas.

c) On agile projects, use cases and oral communication are used more frequently than formal specifications and design documents.

    i) Use cases, user stories, and personas clearly reflect the purpose and user benefit of each feature.

    ii) Demonstrations and peer reviews validate how development has achieved the intended purpose and benefit.

    iii) Information developers are embedded members of the agile team to promote efficient communication and the quality of information deliverables.

d) Early, frequent feature shipments or releases replace scheduled release milestones. This means that information deliverables should be part of the acceptance criteria of a complete functional release or shippable increment.

NOTE For more information about designing and developing information for users in an agile environment, see Clause 7.

Agile development is an iterative and incremental approach to development performed in a highly collaborative manner by self-organizing teams. There are many specific agile development methods that promote development iterations, teamwork, collaboration, and process adaptability throughout the life cycle of the project. Agile development methods frequently discourage the creation of detailed engineering support documentation and detailed technical specifications. This means that information developers often do not have source documentation from which to extrapolate feature details.

The short time span of iterations means that the participation of each team member is essential. In particular, unavailability of the information developer on the team may make it impossible for information deliverables to keep pace with software development.

6 Management of information development

6.1 Change management for agile development

The information development lead or project manager shall decide, in consultation with the designers and product managers, on the following:

— whether the information development team will follow agile development methods and integrate with the development organization, or

— whether they will continue to operate using other methods. For example, some projects may use agile development for the production of online help but not for printed information for users.

The information development team shall be educated on agile development practices before their organization moves to the use of agile development methodologies. In addition, the information

development lead, or project manager, or another appropriate member of the information development team should be involved in defining what the new processes will be for the organization and negotiate how agile development will affect their processes and team members.

The representative from the information development team should then communicate the plans to the wider information development team and may facilitate their implementation.

6.2 Composition of agile development teams

6.2.1 General

An agile team is typically composed of an architect, developers, testers, information developers, user representatives, and other stakeholders. In agile development teams, one individual may be expected to perform multiple roles. For example, information developers may cover different roles and tasks such as team facilitator, user interface designer, integration of user information with the product, testing, and project tracking. This role sharing may depend on the requirements of the project or the availability of resources.

The team member responsible for developing information for users should be skilled in conducting user and task analysis, designing appropriate content deliverables for the identified users, conducting reviews, and assessing information products with test participants.

If an information developer is unavailable, the information development lead or project manager should decide the following:

— whether this team member will be replaced by another information developer from the information development team or pool, or

— whether one of the other members of the agile development team will perform the information developer role.

If another team member who is not an information developer by profession covers the information developer role on the project, arrangements should be made to educate that team member in the skills required and apply special attention to the peer review process.

6.2.2 Communication in agile development teams

Effective communication in an agile development team is key to the success of an agile development project. Because the communication in agile development is real-time rather than through the use of detailed life cycle documentation, information developers should participate in team meetings with the other team members.

It is important to consider the difference between two distinct situations:

— a small, co-located team where the information developer(s) work daily and closely with the development team, and

— a globally distributed team where two or more groups (of the same company or of different software companies) are working together to produce different parts of a software package.

In the first case, all the agile principles and methodologies can be more easily applied. In the second case, which is very common in large companies, there is sometimes an information development group that is spread globally with some members assigned to different groups.

6.2.3 Globally distributed teams

For globally distributed teams, the application of agile principles may involve other considerations such as:

— coordination of several information developers located across different development teams, and

— logistic issues and communication challenges related to different time zones and different languages.

For these reasons, coordination processes should be provided for:

— addressing the communication challenges associated with a globally distributed team, and

— optimizing the workload of information developers according to the iteration schedule.

In this way, the information developers embedded in the different local development teams can facilitate the implementation of agile principles with an approach that is more similar to a small co-located team.

When team members are not co-located, sufficient methods for establishing effective, efficient, and reliable communication shall be put into place. These may include the following communication methods:

— web-based video conferencing for meetings,

— teleconferencing for meetings, and

— one-to-one communication between the information developer and other agile team members.

While agile methods emphasize face-to-face conversation over written documentation, distributed teams need more emphasis on written and formal communications.

The following may also be used to facilitate effective communication:

— collaborative document repositories such as wikis and databases,

— design/feature documents for sharing information,

— use case descriptions, and

— user stories.

When members of the agile development team work in different time zones or do not speak the primary language of communication, special communication accommodations are needed. Temporary assignment to the main location of the agile development team for remote team members may help solve communication issues and establish better communication and relationships between team members when the remotely-based team members return to their locations.

6.3 Management of information development across teams using agile development

In an agile development project, multiple teams may be working on different features of the software. Information developers working on separate agile development teams may create different information deliverables that do not fit together when the product is brought together as a whole. To address this concern, the information development lead or the project manager shall make available standards at the beginning of a project that are to be followed by all information developers for that project, including the:

a) information architecture that defines acceptable information elements such as topic types, their structure, and their intended content,

b) style guide that defines conventions for language (such as tone, voice, mood), graphics (such as file type, size, captions and callouts), and the user interface (such as references to windows, fields, product functions), and

c) delivery formats such as web help or PDF.

The information development lead or the project manager shall allocate sufficient time shortly before each release to consolidate the information for users so that it is consistent and free of contradictions, gaps, and redundancies.

6.4 Management of information development tasks across iterations

6.4.1 Planning the project as a whole

In a project using agile development, information for users is ideally developed in parallel and on the same schedule as the software. This enables software to be regularly released to customers with sufficient information for users. Providing the information for users as early as possible provides time to review and test the information for users alongside the software.

Within each agile development team, a high-level overview of the planned software content of each iteration should be developed. These plans are likely to be subject to change because new user requirements can come into the project for inclusion in future releases at any time.

Development of information for users that relates to the product as a whole shall be included in the planning by the information development lead or project manager who shall plan, schedule, resource, and track these tasks as a part of the project. Ideally, these tasks are determined at the beginning of the project in the project planning stage and are then allocated and drafted as early as possible in the cycle. Planning for the project as a whole should include the minimum information requirements for the product and the information for users.

In an agile environment, it can be difficult to produce detailed plans beyond a single iteration. However, the project manager should produce a project plan that includes the iterations to be developed. The iteration milestones should include software and information deliverables for manufacturing and translation schedules.

The information development lead or project manager should reserve time before the end of a release cycle for final production edits. The information development lead or project manager should set specific freeze dates for the various artifacts related to information for users. The information development lead should determine how much information for users is required and the minimum requirements for the inclusion of task, concept, reference, and troubleshooting information for the content to be usable and ready for release to customers.

Not all information development work will necessarily take place in a single iteration; for example, work related to an information architecture strategy may need to be completed over multiple iterations. Tasks that fit into the bigger picture need to be considered and planned in addition to tasks defined for the individual iterations. Some tasks can be performed in special iterations or at other points in a project's life cycle that can have an impact on the information for users; for example, acceptance test, translation, and accessibility testing. There can also be iterations that focus on consolidation or infrastructure tasks not requiring concurrent information development.

There are different information delivery models depending on the team's resources and the organization:

— For teams with dedicated information developers, the information for users can be released in parallel with the software.

— For organizations that have centralized information developers who are tasked with supporting multiple development teams, more formal and structured approaches such as setting a freeze date are more feasible.

— A hybrid approach is also possible, where an information developer drafts the information roughly in parallel with software development, but has a freeze date at the end of the project to complete and deliver the information for users.

There can also be information development work that needs to be performed but that is not associated with a software feature produced by an agile development team.

Some types of information can be difficult to develop in individual iterations, especially content related to the project as a whole. Examples of information for users that relate to the project as a whole may be installation and configuration guides, tutorials, high-level concepts, reference material, and

troubleshooting information. It may be possible to schedule the development of these information items at the beginning of a project between the acquisition of requirements and the start of software development, sometimes known as iteration zero, or during iterations dedicated to testing or quality assurance. The information for users that is produced for these items should be reviewed at the end of the iteration and considered if any changes occurred during the iteration.

The project planning components to develop information for users in an agile project are described in the following subclauses of this document.

6.4.2 Sizing and resourcing each iteration

6.4.2.1 General

Planning for the project requires planning the content for individual iterations. The information for users produced with the product functions in each iteration is likely to be mostly task-oriented information, although it can also be possible to schedule feature-related concept and reference materials within the individual iterations. Planning for each iteration should include the identification of tasks based upon the features to be completed within that iteration. Development of concept and reference materials can be easier to schedule at the beginning of an iteration, while task-based information may be easier to develop after coding or prototypes have been completed.

Planning for each iteration should include tasks for information development:

— research activity,

— familiarization with the application and how the user would use it,

— adequate review of content, structure, and style, and

— testing of the information for users.

The information developer, information development lead, or project manager should plan for and determine that the details and sizing for the information development work — including user stories, features, and backlog — are included in the plan for each iteration. The information developer, information development lead, or project manager should plan and schedule the required time and resources so that they are included in the iteration length and work item sizing to cover the review, test, and editing of the information.

Time should be provided for testing of the information with the working software, but it can be appropriate to schedule this testing in a separate iteration. Time for updates to the information for users as a result of defects or problems found during review and test should also be included in the plans. Sometimes it is not practical to provide complete and thorough information for users for a software feature developed in a single iteration. Therefore, planning the information for users to accompany the software feature should focus on the most essential information required to support the user. In some agile development teams, planning includes core and stretch items. Instructional information that is required for the user to be able to use the software shall be part of the core items, but supporting information such as overviews and concept documents can form part of the stretch items. The core items should be created as a priority and any agreed items that are not completed shall be carried over into the next iteration. In this case, the planning of the next iteration shall consider the related impact.

6.4.2.2 Sizing information development work

The information development lead, project manager, or information developer should size the information development tasks based on estimates from previous development activities, considering the potential for reuse or adaptation of existing material.

NOTE The complexity of the software development effort does not directly relate to the time and amount of detail required in the information for users. The need for information creation can vary widely between teams and iterations. For instance, one team can do back-end refactoring for two iterations with minimal effect on the information for users.

Depending on the iteration, the information developer should break down large tasks into smaller tasks. Tasks that are sized for more than five days can be difficult to estimate adequately in an iteration, and can result in slippage if the task cannot be completed in the iteration. For short iterations—for example, one or two weeks—tasks sized at over one day should be broken down into smaller tasks.

The information developer shall document and plan for any agreed items that are not completed. These items are carried over into the next iteration. A function is not done until:

— software coding has been completed,

— information for users has been completed,

— both software and information for users have been tested, and

— any issues identified from testing have been resolved or a plan exists to resolve them.

6.4.2.3 Assigning information development resources

In an ideal agile environment, the information deliverables for each user story are completed in the same iteration as the development and testing work. The information development lead or project manager shall list and size the information development tasks before the start of each iteration. If the estimates exceed the resources for the iteration, the information developer should coordinate with the team to prioritize the tasks and, if necessary, schedule the overflow for the following iteration.

In some projects there is a single agile development team, and in others there may be multiple agile development teams. The size of the project using agile development and the relative size of the information development team should determine how the information developers interact with the agile development teams. The number of information developers assigned to an agile development team should be decided by the information development lead or project manager based upon the expected resource requirements for the project. This number may vary between iterations.

If an information developer is shared across multiple agile development teams, a percentage of the information developer's time should be allocated to each team so that the availability can be tracked.

When committing resources to an agile development team, the information development lead or project manager should consider how much time the information developers are likely to spend in meetings. The time spent in these meetings should be included in the total time the information developers are estimated to spend on the project. These meetings include regular stand-up meetings and other meetings where they elicit information for the production of the information for users, such as design meetings, demonstrations, and test reviews. Schedules should be arranged so that the information developer can participate regularly in stand-up meetings for each assigned team.

6.4.3 Handling last-minute content changes

The information developer shall establish a procedure with the agile development team for handling last-minute content changes that come in too late for that iteration. For example, the team agrees that changes received after a certain point in the iteration will be deferred to the next iteration.

6.5 Monitoring and analysing progress

6.5.1 General

Tracking progress is an important part of agile development. Committed items should be successfully completed in the iteration and any potential impediments or holdups should be discovered and resolved as early and rapidly as possible.

6.5.2 Stand-up meetings

Many development methodologies, including agile methods, include daily stand-up meetings. These meetings are usually structured to communicate what tasks each member of the team has completed, what tasks they are going to work on next, what help or resources they need, and what problems they have.

These regular stand-up meetings are intended to be short meetings, for example, for a maximum of 15 min, depending on the size of the agile development team. The meeting leader, who often is the team lead, maintains reports on the team's progress and helps to solve any impediments that are blocking the iteration's progress. Regular stand-up meetings should be held daily or as often as required. Team members who are not co-located shall be included in these meetings using electronic communication means.

The information development lead or project manager shall include members of the information development team in regular stand-up meetings. The information development lead or project manager shall also attend if identified as a stakeholder. The information developers should report their status to the attendees of the meeting. If they are unable to attend, they should arrange to communicate their status and raise any issues through other means such as a team document repository, a wiki, or through email to the meeting leader. Sometimes these regular stand-up meetings also include demonstrations of the software code or the information deliverables that have been produced, and can be an effective way of communicating changes or performing peer reviews or walkthroughs of both the software and the information for users.

The information development lead or project manager should ascertain that the members of the information development team are getting the most value out of the stand-up meetings. These meetings should be useful forums for communication and raising concerns but should not be focused on in-depth discussions of technical implementation. If the meetings are frequently diverted away from the status of the project team or are not providing information that the information developer requires, the information development lead or project manager should work with the meeting leader to focus the meetings.

6.5.3 Monitoring progress

In addition to the regular stand-up meetings, tracking of a project using agile development is usually done with a status list of the individual tasks for the iteration. These tasks include both information development tasks related to the development of a new function and also additional concept and reference information for users.

All of the development tasks for the iteration should be listed, such as content to be developed, reviewed, tested, and edited.

Associated with these tasks should be the sizing estimate for each task and the current status:

— not started,

— in progress,

— under review,

— in test,

— in editing, and

— completed.

Team members should record the actual effort expended to allow improvements in sizing algorithms based on actual data. These records can be used to estimate work remaining to complete the task.

6.5.4 Rework and changing requirements

The team lead and information development lead or project manager should provide guidance to the information developers and other members of the agile development teams on how to handle changing or new requirements. One of the key differences from the information developer's point of view between the agile development and traditional development methods is that the information developer spends time in rework or rewriting of the information for users when changes are made to the design as a result of problems discovered during code development or new requirements. Although this cannot be completely mitigated, developing the information for users in parallel with the code and working directly with the software developer on the specific details can help.

The possibility of rework shall be included in the information development plan for the product as well as for individual iterations. Providing the information development deliverables for review in a different manner than their final format for delivery may assist in leaving costly decisions and production to the last minute, after the majority of required changes have been made.

Making information development plans and schedules readily available and establishing rapid review cycles for required changes makes it easier to include changes and new requirements to the plans.

6.6 Stakeholder involvement

6.6.1 General

Stakeholders are individuals or organizations who have an interest in the software development project. Stakeholders may be directly involved in the projects, or their interest may be affected as a result of the project. The agile team should be responsible for the early identification of stakeholders and determine their involvement with the project. Stakeholders directly involved in the project include project managers, architects, developers, testers, and information developers. Stakeholders who may or may not be directly involved in the project may be users, sales personnel, infrastructure and operational system administrators, and other personnel related to the development of information for users. User involvement is often considered essential in approving iteration content and approving each release.

In agile development, the involvement of stakeholders is essential to the success of the project. The individual stakeholders are directly involved with the development of the software from the beginning of the project and have the opportunity to influence the project and introduce new initiatives. Therefore, the roles and responsibilities of the stakeholders in the project shall be defined and communicated to the stakeholders.

The involvement of stakeholders is important for the information development process as well because it provides opportunities for the information development team to get customer requirements, customer priorities, and customer feedback. Information developers should have the opportunity to actively participate in meetings or other communications with stakeholders in order to clarify requirements and to receive feedback.

6.6.2 Assessing customer satisfaction

The information development lead should work with the project manager and user representative to seek opportunities to acquire feedback from customers through methods such as review comments, beta programs, and usability testing.

Customer feedback may lead to new customer requirements to include in future iterations of the project. These customer requirements may include additions, changes, or updates to the information for users.

7 Preparing information for users

7.1 Relationship of agile development to information development

In agile development it is important that the development of the information deliverables is part of the same processes as the software product life cycle and is performed in conjunction with the development of the software. This enables the software and the information for users to be tested, distributed, and maintained together. In addition, providing the information for users as early as possible allows time to review and test it alongside the software.

The benefits for the information developer of developing the information for users in parallel with the software include the following:

— early access to features under development;

— influence on the software design, particularly of the user interface and the usability of features;

— close interaction with developers, testers, usability experts, and other development roles while the features are under development;

— use of the software at an early stage to find and report defects;

— reduction of information duplication across the development organization;

— access to stakeholders representing the user;

— development of only the information that the user wants and needs;

— receipt of user feedback on the software and the information deliverables early in the development cycle;

— continual user feedback that allows the work load to be spread throughout the development cycle.

In agile development, the software shall not be considered complete without the production and validation of the associated information for users.

7.2 Product design and developing information for users

User stories, personas, and use cases should be developed in the requirements and high-level design stage of the project. These types of information may help inform the information developers and other members of the agile development team about the purpose and uses of the software. The information developers shall be provided with access to this information.

To minimize unnecessary work and the duplication of effort, information developers can be asked to produce or contribute to life cycle documentation. The information for users can itself become the design specification for the product under development where the design details the user interface and how to use it, and not the internal implementation of the software. This approach may have other effects such as changing the order in which information for users is produced or tested. For example, the information for users may exist before the software is implemented and both may be tested at the same time earlier in the cycle in conjunction with features/system test, rather than waiting until the end.

There are a number of techniques that can be used to develop information for users without the use of design specifications, including:

a) pair programming and information development with the developer and information developer;

b) interviews by the information developer to ask specific questions, such as:

   i) Does the user story of this item indicate that users require specific information for using or installing the feature?

ii) How does the item fit into the overall information architecture of the system?

c) prototype or software walkthroughs;

d) reviews and tests of the information for users produced for the iteration.

The information for users may act as a repository for information acquired through direct communication. The iterative approach to development should result in no need to return to information for users produced during an earlier iteration, except where it is required to be updated for new features or defects in a later iteration. It can be useful to record some of the information exchanged for tracking purposes or for changes in personnel. For example, meeting minutes may be taken, outstanding issues may be documented, and walkthroughs may be recorded. Records should be kept in a content management system in accordance with the organization's standard procedures.

The primary way to ascertain that the information for users under development is up to date with the software design is regular communication, either face-to-face or via virtual conferences. The information developer shall attend both regular stand-up meetings and design meetings to keep informed of any updates, have the opportunity to express concerns, and ask for assistance if required. If the information developer is not physically located with the agile team, arrangements should be made to allow the information developer to participate remotely. If the information developer cannot attend a team meeting, a representative in the meeting shall raise concerns on behalf of the information developer and convey any relevant information from the meeting back to the information developer.

The agile development team shall use a controlled source location for published designs and high-level design documents. Comments and discussions around the design should be included with the published designs.

Ideally, the comments and discussions are integrated with the published designs through iterative updates to show how the discussions have changed the design. At the beginning of each new iteration, when an existing design is updated, the latest version of the design should be used as the base design for the new iteration.

Information developers should be able to access both the current design documentation and the software prototypes to verify that the information for users they are developing is correct for the iteration. Interviews with the software developers and reviews of change logs of the design documentation are also helpful.

7.3 Design and development of information for users

7.3.1 User requirements

User requirements should form the basis for information for users produced in an agile development project. Most of the information for users produced will be associated with code, but there can be separate features that relate to specific user requirements that affect only the information for users.

User requirements should be recorded, and the project plans and information development plans should make reference to these user requirements so that new or changed features can be tracked back to the user requirements to which they are related.

The user requirements should be the basis for the creation of design documents such as personas, user stories, or scenarios. User stories or scenarios should be created for tasks that only affect the information for users; for example, an overhaul to the current information navigation or the production of some new information development guidelines. Design documents such as user stories and scenarios can help the agile development team understand the value of these tasks for the users and provide a mechanism for developing testing to validate the tasks as well as provide guidance on how to complete the tasks.

NOTE ISO/IEC 26514 specifies the structure, content, and format for user documentation and also provides informative guidance for the style of information for users.

7.3.2 Design techniques

7.3.2.1 General

A number of techniques are used in agile development to help design both software and information for users based on user needs and requirements. These techniques may be used as part of a formal design phase at the beginning of a project between the acquisition of requirements and the start of software development, sometimes known as iteration zero. Alternatively, these techniques may be used at the beginning of each iteration for the design of individual features.

7.3.2.2 Use cases

Use cases describe whether the sequence of interactions between users and the system satisfies the users' goal. Use cases also represent how the users will use the system before considering the way it will be implemented. Use cases are a valuable source of information for an information developer because they provide information about how a user is expected to use the system, what options the user has, what goals the user is trying to achieve, and what error conditions can occur.

A use case records:

— who (users) does what (interaction) with the system, and

— the purpose (goal) without dealing with system internals.

Use cases focus on the user's point of view and show that the feature provides value to the user. The use case is understood by the different roles on the development team as well as users and other stakeholders involved with the project.

Information for users should support the user in completing the goals defined in the use cases to be developed for the iteration.

7.3.2.3 User roles

For any given product there are likely to be one or more user roles that will use that product. Each user role will have particular tasks to perform or goals to achieve. Each user role may have a set of associated skills and the user may work in a particular environment.

User roles are useful sources of information for an information developer because they provide information about the different types of users using the system. The user role descriptions may also provide information about what skills and requirements each role has and this may affect the information that is produced for each role.

EXAMPLE User roles include end user, developer, systems administrator, technical support officer, lab assistant.

7.3.2.4 Personas

A persona should be developed through rigorous research. The information provided in the persona can assist the information developer in producing information for different user roles. As well as providing information about typical user goals and skills, personas can be used to help design questions and tasks for testing the information for users.

The following information can be included in a persona to help focus on the needs of the intended users:

— character name;

— role;

— experience;

— relevant skills;

— demographic information such as age, education, or gender;

— organization;

— major responsibilities;

— goals and tasks that the persona wants to complete;

— the problem or business opportunity that the product solves for the persona's organization;

— the persona's physical and social environment;

— a representative image of the persona;

— the software and hardware platforms the persona is using;

— size of the organization.

7.3.2.5 User stories

User stories are important for determining, describing, and prioritizing customer requirements and are usually expressed in terms of feature development. User stories are a simpler and user-focused alternative to producing lengthy detailed requirements specifications.

The user stories define how the software will work for the users and provide an important source for the design of the software according to user needs, how the software will be tested, and what the information for users should contain.

User stories are collaborative documents to be used by multidisciplinary teams including software developers, testers, information developers, and other stakeholders. User stories may be written by any member of the software development organization, including the intended users of the software. User stories should also be written for features that only affect information for users, such as a glossary or a search mechanism.

User stories differ in their level of complexity depending on organizational requirements, but they should always be expressed in business terminology that is understood by all stakeholders, not just by the development organization.

The main content of the user story often takes the form: "As a (role), I want (goal), so that I can (business value)." For example, "As a network administrator, I want to be able to check the status of groups of network devices with shared characteristics so that I can identify issues early and reduce administrative efforts." Acceptance criteria are developed to identify when the user story has been successfully implemented and users are able to achieve their goals.

For the previous example, the acceptance criteria might include:

— view the current status of one or more network devices, and

— group resources by network, vendor, service, or status.

A user story should contain the following information:

— the user role that the story applies to;

— the goal that the story allows the user to achieve;

— the value to the customer;

— acceptance criteria to identify when the user story is done.

Requirements regarding the information for users shall be included as part of the acceptance criteria for a user story.

A user story may also contain the following information:

— a unique identifier for the story;

— additional details; for example, the conversation between stakeholders and development, or the planned implementation;

— the name of the user story;

— the priority of the user story;

— the size of the user story.

If the work required to fully complete the implementation, testing, and documentation of a user story is greater than a few days, it should be treated as an epic and broken down into smaller stories. Large user stories can be difficult to size and to successfully complete within a single iteration.

User stories are useful sources of information for information developers because they provide information about user goals and about how the function under development will work for the users.

7.3.2.6 Scenarios

Scenarios extend stories by using a specified user. For example, a persona-based scenario may describe specific user requirements or ways that users can use a product based on their existing knowledge and experience or their environment. A scenario can be a very detailed account of how the user in the persona interacts with a system or can be briefer and focus on high-level requirements.

The difference between a scenario and a user story is that a user story tends to focus on the ideal case where the product performs the exact task the user wants to perform, exactly as they expect, and with no errors or unexpected behaviour from the users' point of view. Problems with non-ideal cases may be highlighted using scenarios which explore alternate paths.

A scenario may contain the following information:

— a description of the persona that the scenario applies to;

— the goal that the completion of the scenario allows the user to achieve;

— the specific needs of the user role defined in the persona;

— the actions the user takes in interacting with the system based on their specific requirements or skills;

— additional details; for example, the conversation between stakeholders and development or the planning implementation;

— the name of the scenario.

The information provided in the scenario description can assist the information developer in producing information items for different user roles and in support of their defined user goals. As well as providing information about typical user goals and skills, the scenarios may be used to help design questions and tasks that can be used in testing the information for users.

7.3.3 Scheduling of design work

The information development lead, project manager, or team lead should aim to do any information architecture or design work early. They should get commitment for this work from the agile development project manager and other stakeholders during the project planning stage.

These information architecture or design work items should be included in the tracking for the project, broken down into small-sized tasks, and allocated to the appropriate resources.

The information development lead, project manager, or information developer should work with the agile development team to identify specific iterations that will focus on issues related to information for users, such as accessibility, usability, and translation.

7.3.4 Planning of information for users

Information development planning shall be conducted for user information as a whole as part of the project or epic, before the beginning of the agile development project iterations. An information development plan should list the expected final deliverables and define the user goals for the product, the tasks users hope to perform with the product, and the supplementary information (concepts, reference information, troubleshooting tips) required to support those tasks and make users more confident and productive in using the product.

Within the project iterations themselves, the information developer should identify the topics from the project plan that are affected by the product development tasks within the iteration. Information development user stories should focus on user goals and actions rather than descriptions of product functionality. As part of the iteration planning process, the information developer should identify what is required from the product development team. For example, the information developer may require access to the prototypes or time with the developers to acquire the details of the product design.

Information development tasks may be addressed by separate information for users-specific user stories or as a part of a larger product development user story. In either case, the definition of done for the user story or iteration should include the completion of the related information development tasks. Information development tasks shall include quality assurance tasks such as editing and testing of the content with the product. Stories and tasks for information for users should be prioritized to enable the completion of a minimally viable document starting with task content and adding supplementary material only as time in the iteration allows.

To demonstrate that final information deliverables are cohesive and well structured, a final consolidation or hardening iteration should be scheduled, which includes time for an end-to-end review of the content as well as testing and defect fixing for both the product and user information.

7.4 Reviewing and testing information for users

7.4.1 General

In all projects, reviewing and testing the information for users is important in assuring the accuracy, completeness, fitness for task, and consistency of the information.

NOTE ISO/IEC/IEEE 26513 provides requirements and guidance for reviewing, system testing, usability testing, accessibility, and localization testing of information for users.

7.4.2 Reviewing information for users

At a minimum, the agile development team shall peer review all information development deliverables for adherence to style guidelines and technical accuracy. Other information developers, editors, or other members of the team with strong communication skills shall conduct the peer review; developers and testers shall review the deliverables for technical accuracy.

Reviewing tools whereby members of the agile development team can provide direct feedback about the content are helpful in an agile environment.

Another effective method of reviewing information for users is to hold a walkthrough meeting with the appropriate stakeholders in the team to review the structure and content of the information. Changes and updates may be made to the information either during or after the meeting.

In some cases, the agile development team may decide to offset the development of information for users for a short period of time or to delay it by an entire iteration so that the information does not need to be entirely rewritten in response to product design changes. In the former case, teams should

agree on what part of the information for users has to be completed to enable the iterations to be considered done, even though the information for users has not been delivered. In the latter case, user stories for information for users should be written separately from the product development stories and prioritized into the iteration following the completion of the related product functionality. “Done” may include the following:

— final terminology has been validated;

— user interface text has been reviewed;

— embedded help has been provided;

— conceptual information has been included;

— multimedia assets have been developed;

— translations have been completed.

Required changes or updates should be made before the end of the iteration or scheduled for the next iteration. The information for users should be available in draft form and available for review with sufficient time to enable the information developer to make the corrections or additions.

7.4.3 System test of information for users

If the project is determined by the team lead or by the project manager to require a system test effort, the system test plans shall include plans for testing the information for users. These plans shall be reviewed and approved by the information developer and information development lead or project manager.

Time should be included in the iteration for corrections to be made to the information for users after system testing has been completed.

One effective strategy for incorporating changes and updates rapidly into the information for users during system testing of the deliverables is to use a pair programming approach. Using a pair programming approach, the information developer responsible for the information for users under test sits with the tester while the tests are performed. In some testing environments, the information developer can make changes to the information as the tester identifies the problems.

Where this type of activity is not possible, a problem report shall be created by the tester and resolved according to the problem resolution standards of the organization.

A further review or test cycle may be required so that updates to the information for users have been correctly addressed.

7.4.4 Usability testing of information for users

Usability testing of information for users shall be carried out using real or representative users. Usability testing of information for users may be carried out at earlier stages of the life cycle or when the software is ready for a release.

Usability testing that includes the information for users may be carried out in each iteration, but at a minimum, there shall be one usability test of the information for users. This test may be carried out as a part of a user acceptance iteration or as a part of a customer satisfaction assessment activity.

7.5 Translation and localization of information for users

In order to market products to some countries or some organizations, it may be a requirement to have a translated or localized version available at the same time as a release. The agile development team should include these factors in their planning to keep the translated or localized versions up to date.

The information development lead or project manager shall plan the translation schedule for the information for users. The following items should be taken into consideration in planning the translation schedule:

— whether the information for users needs to be translated or localized for each iteration;

— if the information for users is released to the user for each iteration;

— time required to perform the translation or localization;

— time to package and handle returns of translated or localized information for users;

— time to test translated or localized information for users;

— local translation, local requirements, and regulations.

Some organizations use a strategy where the information for users is translated at the end of an iteration and then incorporated in the next iteration's build.

7.6 Production of information for users

If information deliverables for the project need to undergo additional processing before release to customers, the time required for this activity and scheduling should be included in the planning for the project. For example, printed information for users may require time for preparation and printing.

7.7 Delivering information for users with a continuous delivery process (DevOps)

Agile development teams may leverage continuous integration and continuous delivery practices to minimize the time between authoring new software and pushing the new features to the users or customers. Developers rely on an increased level of automation and extensive instrumentation to service their needs, deploy quickly, and fix unsuccessful changes. In such a scenario, sometimes referred as DevOps, development teams do not rely on a dedicated IT team to service their IT infrastructure, but are able to largely service their own needs. Furthermore, release cycle decisions can be made by the development team and not by dedicated release management staff that are external to the team.

Information for users should be created, reviewed, edited, and published rapidly, frequently, and reliably. Ideally, the content is developed and released in parallel with the software.

For information for users, this means:

— easier and faster updates;

— more people actively writing and editing.

The authoring and publication process shall allow for rapid changes to the content and for reverting content updates to reflect changes and reversions in the product or service.

Annex A
(informative)

Agile development practices

Agile development is not a single process or set of defined methodologies, but rather, a set of principles and related methodologies that grew out of the "Agile Manifesto." Key features of these agile development methodologies are iterative development and the evolution of requirements and solutions over time through the collaboration between self-organizing cross-functional teams. Different agile development methodologies have varying terminology. These differences in terminology associated with particular methods are not described in this document. Agile development methodologies focus on delivering functional value for users and decreasing costs. Costs can be reduced by developing high value requirements as a priority and finding defects early in the development cycle when it is cheaper to fix them.

Agile development methodologies encourage clear and regular communication so all the interested parties can be involved in all stages of development. Because agile development methods are focused on fixed time periods for development, the risk of going over budget is reduced. The timescale and cost of the project is fixed, but the scope and features are flexible.

Functional value is added for users by using iterative approaches where a useful function is developed and delivered at the end of an iteration. Such a functionality is developed, tested, and provided with supporting information for users. Because a function is developed and validated rapidly, this function can be delivered earlier and updated regularly.

Early release also provides opportunities to acquire user feedback and new requirements to feed into the next iteration. Continuous validation through testing and stakeholder involvement provides frequent status on the progress of the product, including the level of quality and fitness for the purpose of the product. The users or a representative for the users are involved in each stage of the development process and can check that value is being added and the right requirements are being addressed. Development may stop if new requirements emerge or changes are requested so that the work items can be correctly prioritized.

Agile development methods rely on the individual teams to decide how to develop and deliver a product that satisfies the requirements. Involvement from requirements to delivery encourages all team members to have the same knowledge and understanding about the user requirements and how they will be fulfilled. Because the team works together and has the same knowledge of the what, why, and how, there is less immediate need to produce and understand detailed specifications for both the project life cycle and the tracking project status. Any problems that occur may be discussed and resolved quickly or identified as severe blocking problems where current work is stopped in favour of moving to something more productive.

Agile development methodologies also provide regular opportunities to look at how the team or the project is working and make changes to make the process more effective and efficient.

Most agile development methods use some form of iterative development. Agile projects have a planning and scheduling strategy, in which the various features of the product are developed to completion during a specified period of time, called an iteration. Iterations of two to six weeks are the most common. During this time the feature is planned, designed, developed, and tested, along with the information for users.

At the end of the iteration, the feature and the product for which it is being developed is complete enough to deliver to users, as either a beta or a product.

Table A.1 describes several typical agile methods in alphabetical order.

Table A.1 — Typical agile methods

Method	Source and date	Description
Crystal Clear	Alistair Cockburn, early 2000s	Puts particular emphasis on the interactions with the team through "osmotic communication." With osmotic communication, questions and answers flow naturally and with surprisingly little disturbance among team members.
Disciplined agile delivery (DAD)	Scott Ambler and Mark Lines, 2010s	Combines elements from Scrum, Lean Software Development, and other agile methodologies to support incremental and iterative software solutions on an enterprise scale.
Extreme Programming (XP)	Kent Beck, Ward Cunningham, Ron Jeffries, 1995+	The basic advantage of XP is that the whole process is visible and accountable. The developers will make concrete commitments about what they will accomplish, show concrete progress in the form of deployable software, and when a milestone is reached they will describe exactly what they did and how and why that differed from the plan. This allows business-oriented people to make their own business commitments with confidence, to take advantage of opportunities as they arise, and eliminate dead-ends quickly and cheaply.
Kanban	1940s for manufacturing Mid-2000s for software	Model for introducing change through incremental improvements with an emphasis on continual delivery while not overburdening the development team.
Lean Software Development	Mary and Tom Poppendieck, early 2000s	Lean Software Development owes much of its principles and practices to the Lean Enterprise movement and the practices of companies like Toyota. Lean Software Development focuses the team on delivering value to the customer and on the efficiency of the "Value Stream," the mechanisms that deliver value.
Scrum	Ken Schwaber and Jeff Sutherland, early 1990s, based on work of Hirotaka Takeuchi and Ikujiro Nonaka	Emphasizes empirical feedback, team self-management, and striving to build properly tested product increments within short iterations.
Rational Unified Process (RUP)	Rational Software Corporation, 1990s	RUP is an adaptable process framework focused on iterations. It divides the software development life cycle into iterative cycles, with each cycle working on a new version of the product. Each cycle is broken up into four phases: Inception, Elaboration, Construction, and Transition. Within each phase, the tasks are categorized into nine disciplines, six engineering disciplines (business modeling, requirements, analysis and design, implementation, test, deployment) and three supporting disciplines. RUP supports the use of UML (Unified Modeling Language) as its primary notation for requirements, architecture, and design. RUP emphasizes the adoption of various best practices in order to reduce the risks in software development.

Bibliography

[1] Manifesto for Agile Software Development, 2001. http://agilemanifesto.org/

[2] ISO/IEC/IEEE 12207:2017, *Systems and software engineering — Software life cycle processes*

[3] ISO/IEC/IEEE 15288:2015, *Systems and software engineering — System life cycle processes*

[4] ISO/IEC/IEEE 15289, *Systems and software engineering — Content of life-cycle information items (documentation)*

[5] ISO/IEC/IEEE 26511, *Systems and software engineering — Requirements for managers of information for users of systems, software, and services*

[6] ISO/IEC/IEEE 26513, *Systems and software engineering — Requirements for testers and reviewers of information for users*

[7] ISO/IEC 26514:2008, *Systems and software engineering — Requirements for designers and developers of user documentation*

IEEE notices and abstract

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Standards Documents."

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

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute 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 content 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. 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: PROCUREMENT OF 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 should be considered the approved IEEE standard.

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, or 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 his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide 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 revisions to an IEEE standard is welcome to join the relevant IEEE working group

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board

445 Hoes Lane

Piscataway, NJ 08854 USA

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 imply 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.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. 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.

Photocopies

Subject to payment of the appropriate fee, 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. 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 ten years. When a document is more than ten 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 the IEEE-SA Website at http://ieeexplore.ieee.org/xpl/standards.jsp or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

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 http://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.

Participants

The list of IEEE participants can be accessed at the following URL: https://standards.ieee.org/standard/26515-2018.html.

Abstract and keywords

This document specifies the way in which information for users can be developed in agile development projects. It is intended for use in all organizations that are using agile development or are considering implementing their projects using these techniques. It applies to people or organizations producing information for users, to those undertaking a single project, and to information for users produced internally, as well as to information for users contracted to outside service organizations. This document addresses the relationship between the information development process and the software life cycle process in agile development. It describes how the information developer or project manager may plan and manage the development of information for users in an agile environment. It is intended neither to encourage nor to discourage the use of any particular agile development tools or methods.

Keywords: agile development, user documentation, information for users.

ICS 35.080

ISBN 978-1-5044-5297-7 STD23402 (PDF); 978-1-5044-5298-4 STDPD23402 (Print)

Price based on 22 pages