

ISO/IEC FDIS 33202:2024(E)

ISO ~~JTC1~~ / IEC ~~JTC 1~~ / SC 7 / ~~WG 29~~

Secretariat: BIS

Date: 2024-~~03~~-~~2004-29~~

Software and systems engineering — Core agile practices

FDIS stage

Document Preview

ISO/IEC FDIS 33202

<https://standards.iteh.ai/catalog/standards/iso/95f2dba1-0ac6-4316-888a-666dca63cdaf/iso-iec-fdis-33202>

© ISO/IEC ~~2023~~2024

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 either ISO at the address below or ISO's member body in the country of the requester.

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

Published in Switzerland

iTeh Standards
(<https://standards.iteh.ai>)
Document Preview

ISO/IEC FDIS 33202

<https://standards.iteh.ai/catalog/standards/iso/95f2dba1-0ac6-4316-888a-666dca63cdaf/iso-iec-fdis-33202>

Contents

Foreword.....	viii
Introduction	ix
1 Scope.....	1
2 Normative references.....	1
3 Terms and definitions	1
4 Conformance.....	5
4.1 Intended usage.....	5
4.2 Full conformance	5
5 Conceptual foundations.....	5
5.1 Practices and processes.....	5
5.2 Agile practices and concepts.....	6
5.3 Agile practices.....	7
5.4 Criteria for practices.....	7
5.5 Description of practices.....	7
5.6 Some agile concepts	7
5.6.1 Cadence.....	7
5.6.2 Gemba	8
5.6.3 Limiting work in progress	8
6 Core agile practices	8
6.1 Agreement practice.....	8
6.1.1 Purpose.....	8
6.1.2 Overview	8
6.1.3 Outcomes	8
6.2 Acquisition practice	9
6.2.1 Purpose.....	9
6.2.2 Overview	9
6.2.3 Outcomes	9
6.3 Supply practice	9
6.3.1 Purpose.....	9
6.3.2 Overview	9
6.3.3 Outcomes	9
6.4 Cross-functional team practice	9
6.4.1 Purpose.....	9
6.4.2 Overview	10
6.4.3 Outcomes	10
6.5 Feature team practice	10
6.5.1 Purpose.....	10
6.5.2 Overview	10
6.5.3 Outcomes	10
6.6 Component team practice.....	11
6.6.1 Purpose.....	11
6.6.2 Overview	11
6.6.3 Outcomes	11
6.7 Co-located team practice.....	11
6.7.1 Purpose.....	11

6.7.2	Overview	11
6.7.3	Outcomes	11
6.8	Distributed team practice	12
6.8.1	Purpose.....	12
6.8.2	Overview	12
6.8.3	Outcomes	12
6.9	Collective ownership practice	12
6.9.1	Purpose.....	12
6.9.2	Overview	12
6.9.3	Outcomes	12
6.10	Release planning practice.....	13
6.10.1	Purpose.....	13
6.10.2	Overview	13
6.10.3	Outcomes	13
6.11	Deployment planning practice.....	13
6.11.1	Purpose.....	13
6.11.2	Overview	13
6.11.3	Outcomes	13
6.12	Iteration planning practice	14
6.12.1	Purpose.....	14
6.12.2	Overview	14
6.12.3	Outcomes	14
6.13	Synchronized iteration practice	14
6.13.1	Purpose.....	14
6.13.2	Overview	14
6.13.3	Outcomes	14
6.14	Daily synchronization meeting practice.....	15
6.14.1	Purpose.....	15
6.14.2	Overview	15
6.14.3	Outcomes	15
6.15	Cross-team synchronization meeting practice.....	15
6.15.1	Purpose.....	15
6.15.2	Overview	15
6.15.3	Outcomes	15
6.16	Product owner synchronization meeting practice	16
6.16.1	Purpose.....	16
6.16.2	Overview	16
6.16.3	Outcomes	16
6.17	Value stream mapping practice	16
6.17.1	Purpose.....	16
6.17.2	Overview	16
6.17.3	Outcomes	17
6.18	Release review practice.....	17
6.18.1	Purpose.....	17
6.18.2	Overview	17
6.18.3	Outcomes	17
6.19	Iteration review practice	18
6.19.1	Purpose.....	18
6.19.2	Overview	18
6.19.3	Outcomes	18
6.20	Team retrospective practice.....	18
6.20.1	Purpose.....	18

6.20.2	Overview	18
6.20.3	Outcomes	18
6.21	Program retrospective review practice.....	19
6.21.1	Purpose.....	19
6.21.2	Overview	19
6.21.3	Outcomes	19
6.22	Program board governance practice	19
6.22.1	Purpose.....	19
6.22.2	Overview	19
6.22.3	Outcomes	19
6.23	Agile adoption improvement practice	20
6.23.1	Purpose.....	20
6.23.2	Overview	20
6.23.3	Outcomes	20
6.24	Visual management practice	20
6.24.1	Purpose.....	20
6.24.2	Overview	20
6.24.3	Outcomes	20
6.25	Distributed agile governance practice	21
6.25.1	Purpose.....	21
6.25.2	Overview	21
6.25.3	Outcomes	21
6.26	User stories practice	21
6.26.1	Purpose.....	21
6.26.2	Overview	21
6.26.3	Outcomes	22
6.27	Product backlog elicitation practice	22
6.27.1	Purpose.....	22
6.27.2	Overview	22
6.27.3	Outcomes	22
6.28	Product backlog refinement practice	22
6.28.1	Purpose.....	22
6.28.2	Overview	22
6.28.3	Outcomes	22
6.29	User story mapping practice.....	23
6.29.1	Purpose.....	23
6.29.2	Overview	23
6.29.3	Outcomes	23
6.30	Relative estimation practice	23
6.30.1	Purpose.....	23
6.30.2	Overview	23
6.30.3	Outcomes	23
6.31	Active stakeholder participation practice	24
6.31.1	Purpose.....	24
6.31.2	Overview	24
6.31.3	Outcomes	24
6.32	Initial architecture practice	24
6.32.1	Purpose.....	24
6.32.2	Overview	25
6.32.3	Outcomes	25
6.33	Intentional architecture practice.....	25

6.33.1 Purpose.....	25
6.33.2 Overview	25
6.33.3 Outcomes	25
6.34 Architecture description practice.....	26
6.34.1 Purpose.....	26
6.34.2 Overview	26
6.34.3 Outcomes	26
6.35 Incremental architecture practice.....	26
6.35.1 Purpose.....	26
6.35.2 Overview	27
6.35.3 Outcomes	27
6.36 Simple design practice	27
6.36.1 Purpose.....	27
6.36.2 Overview	27
6.36.3 Outcomes	27
6.37 Test-driven development practice	28
6.37.1 Purpose.....	28
6.37.2 Overview	28
6.37.3 Outcomes	28
6.38 Acceptance test-driven development practice.....	28
6.38.1 Purpose.....	28
6.38.2 Overview	28
6.38.3 Outcomes	28
6.39 Behaviour-driven development practice	29
6.39.1 Purpose.....	29
6.39.2 Overview	29
6.39.3 Outcomes	29
6.40 Feature-driven development practice	30
6.40.1 Purpose.....	30
6.40.2 Overview	30
6.40.3 Outcomes	30
6.41 Agile testing pyramid practice	30
6.41.1 Purpose.....	30
6.41.2 Overview	30
6.41.3 Outcomes	30
6.42 Continuous integration practice.....	31
6.42.1 Purpose.....	31
6.42.2 Overview	31
6.42.3 Outcomes	31
6.43 Continuous deployment practice	31
6.43.1 Purpose.....	31
6.43.2 Overview	31
6.43.3 Outcomes	31
6.44 Continuous delivery practice.....	32
6.44.1 Purpose.....	32
6.44.2 Overview	32
6.44.3 Outcomes	32
6.45 Continuous documentation practice.....	32
6.45.1 Purpose.....	32
6.45.2 Overview	32
6.45.3 Outcomes	32
6.46 Lean documentation practice.....	33

6.46.1 Purpose.....	33
6.46.2 Overview	33
6.46.3 Outcomes	33
6.47 Feature toggles practice	33
6.47.1 Purpose.....	33
6.47.2 Overview	33
6.47.3 Outcomes	34
6.48 Refactoring practice.....	34
6.48.1 Purpose.....	34
6.48.2 Overview	34
6.48.3 Outcomes	34
6.49 Test strategy practice	35
6.49.1 Purpose.....	35
6.49.2 Overview	35
6.49.3 Outcomes	35
6.50 User story testing practice.....	35
6.50.1 Purpose.....	35
6.50.2 Overview	35
6.50.3 Outcomes	35
6.51 Feature testing practice.....	36
6.51.1 Purpose.....	36
6.51.2 Overview	36
6.51.3 Outcomes	36
6.52 User acceptance testing practice.....	36
6.52.1 Purpose.....	36
6.52.2 Overview	36
6.52.3 Outcomes	36
6.53 Regression testing practice	37
6.53.1 Purpose.....	37
6.53.2 Overview	37
6.53.3 Outcomes	37
Annex A (Informative) Summary of core agile practices	38
Annex B (informative) Relationship with other standards	40
Annex C (informative) Manifesto for agile software development.....	45
Annex D (Informative) Application of agile in ISO/IEC/IEEE 12207	47
Bibliography	49

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 document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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.

This document was prepared 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.

Introduction

Many organizations in the software development industry are adopting agile life cycle to adapt to rapidly changing and evolving marketplaces. “Agile” is derived from the Agile Manifesto^{[10],[8]]}, which was designed to ‘uncover better ways of developing software by doing it and helping others do it’. It emerged as an alternative to documentation-driven, heavyweight software development processes. The Agile Manifesto^{[10],[8]]} and related values and principles ([Annex C](#)^(Annex C)) have served as reference for agile practices across many organizations.

Agile development is relative to how each organization interprets the agile values and principles (See [Annex C](#)^(Annex C)). The agile practices discussed in this document offer a lightweight, adaptive and collaborative development approach based on empiricism with the focus on rapid business-value delivery.

Agile standardization refers to adopting agile practices and concepts across an organization or project for software suppliers, software acquirers, software service providers, large delivery and maintenance projects and system integration engagements having software components.

Standardization of agile practices is useful for the following reasons:

- a) ~~a)~~ for suppliers: it provides a basis for delivering meaningful products and services;
- b) ~~b)~~ for acquirers: it provides a basis to differentiate between different suppliers and provides guidance on their role in the development.

The purpose of this document is to catalogue the core agile practices (see [Annex A](#)^(Annex A)) and concepts demonstrated by organizations or projects. These practices and concepts enable the application of agile in acquisition, supply, development, operation, maintenance and disposal of software systems, products and services.

This document can be used by acquirers, suppliers, developers, integrators, operators, maintainers, managers, quality assurers and users of software systems, products and services, for a range of situations and contexts.

<https://standards.iteh.ai/catalog/standards/iso/95f2dba1-0ac6-4316-888a-666dca63cdaf/iso-iec-fdis-33202>

Software and systems engineering — Core agile practices

1 Scope

This document defines a set of core practices and concepts that have wide acceptance in organizations and industries using agile practices and concepts. This document defines a set of core practices that are present in agile methodologies.

The practices and concepts defined in this document are applicable to a single agile team, as well as to multiple agile teams. These practices and concepts are applicable throughout the life cycle of software systems, products and services.

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 and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1 acceptance criteria

criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.32, definition 1]

3.2 agile

approach to development, delivery and maintenance of products and services by enabling rapid response to feedback

Note 1 to entry: The need specification for the software under development is elaborated with specific requirements only when the work is started. This lean principle is meant to avoid waste of work and to provide *agile team* (3.6(3.6)) with means of prioritizing their work.

3.3 agile concept

fundamental building blocks of *agile* (3.2(3.2)).

3.4 agile development

software development approach based on iterative development, frequent inspection and adaptation, and incremental deliveries, in which requirements and solutions evolve through collaboration in cross-functional teams and through continuous stakeholder feedback

[SOURCE: ISO/IEC/IEEE ~~24748-1:2018~~24765:2017, 3.4.119]

3.5 agile environment

organizational culture, infrastructure, and methodologies that support *agile development* (3.4(3.4))

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.2]

3.6 agile team

organization or team using *agile development* (3.4(3.4)) methods and approaches

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.3, modified — Note 1 to entry has been removed.]

3.7 architecture

fundamental concepts or properties of an entity in its environment and governing principles for the realization and evolution of this entity and its related life cycle *processes* (3.28)

[SOURCE: ISO/IEC/IEEE 42010:2022, 3.2, ~~modified – The notes to entry have been removed~~]

3.8 architecture description

work product used to express an *architecture* (3.7(3.7))

[SOURCE: ISO/IEC/IEEE 42010:2022, 3.3, ~~modified – The abbreviated term and notes to entry have been removed.~~]

3.9 backlog

collection of features or stories of both functional and non-functional requirements that are typically sorted in an order based on value priority

Note 1 to entry: Backlog traditionally means things that are not completed.

3.10 continuous delivery

software engineering *practices* (3.20(3.21)) that allow for frequent releases of new systems (including software) to staging or various test environments through the use of automated tools

[SOURCE: ISO/IEC/IEEE 32675:2022, 3.1]

3.11 continuous deployment

automated *process* (3.28) of deploying changes to production by verifying intended features and validations to reduce risk

[SOURCE: ISO/IEC/IEEE 32675:2022, 3.1]

3.12

continuous improvement

ongoing cycle of identification, implementation and evaluation of changes to how *practices* ~~(3.20)~~(3.21) are used that strengthen the ability of an organization or project to meet its objectives

3.13

definition of done

DoD

statement on the required quality attributes that work meets before the work completes a specified life cycle activity or task and is ready ~~to~~for use

3.14

definition of ready

DoR

statement on the required quality attributes a *backlog* ~~(3.9)~~(3.9) item or other work product meets before it starts a specified activity or task

3.15

gemba

observation of work *processes* ~~(3.28)~~ actually in use

3.16

improvement backlog

~~an~~ intentional, emergent, ordered list of items of how to ~~improve~~make improvements

3.17

increment

tested, deliverable version of a software product that provides new or modified capabilities

3.18

iteration

short time frame in which a set of software features is developed, leading to a working product that can be demonstrated to stakeholders

Note 1 to entry: Some *agile* ~~(3.2)~~(3.2) methodologies are not based on iterations.

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.10, modified — The original note 1 to entry has been removed; note 2 to entry has been renumbered as note 1 to entry.]

3.19

iteration backlog

subset of a *backlog* ~~(3.9)~~(3.9) chosen for inclusion in the current *iteration* ~~(3.18)~~(3.18)

~~3.2120~~

practice

way of consistently performing activities that contributes to achieving a specific purpose

~~3.2221~~

product backlog

prioritized listing of product requirements

~~3.2322~~

product owner

designated stakeholder accountable for defining and accepting outcomes of the work and managing *backlog* (3.9(3.9)), while aligning with the stakeholder needs

~~3.2423~~

release backlog

subset of *product backlog* (3.21(3.22)) planned for the next release

~~3.2524~~

statement of work

statement of the expected outcomes and outline of the work required to achieve the outcomes

Note 1 to entry: Statement of work is unique and distinct for each project and often provides a measurable *definition of done* (3.13(3.13)). It is usually a contractual document prepared during project initiation and planning that describes what outcomes are expected and outlines the work required to achieve these outcomes.

~~3.2625~~

team backlog

~~user and enabler stories that originate from the program backlog (3.9), as well as stories that arise locally from the team's local context~~

~~3.27~~

testing pyramid

graduated series of tests which includes many simple and automated tests (unit tests) with less frequent integration tests and few lengthy end-to-end or manual tests

~~3.2826~~

user story

brief description of desired functionality

Note 1 to entry: User story can describe the stakeholder roles, goals, benefits, and motivation

<https://standards.iteh.ai/catalog/standards/iso/95f2dba1-0ac6-4316-888a-666dca63cdaf/iso-iec-fdis-33202>

Note 2 to entry: Work in backlog can be referred in other ways like epics, features

~~3.2927~~

working agreement

statement of the expected working relationship between two or more parties in an agile environment

EXAMPLE Between members of an *agile team* (3.6(3.6)), between acquirer and supplier, or between multiple agile teams.

~~3.3028~~

process

set of interrelated or interacting activities that transforms inputs into outputs

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.33]