
**Software and systems engineering —
Agile development — Agile adoption
considerations**

*Ingénierie du logiciel et des systèmes — Développement agile —
Considérations relatives à l'adoption de la méthode agile*





COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2021

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

Published in Switzerland

Contents

Page

Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Agile readiness factors	3
4.1 Overview	3
4.2 Organizational agile readiness factors	4
4.2.1 Funding	4
4.2.2 Management support	4
4.2.3 Organizational change (organization)	5
4.2.4 Change support	5
4.2.5 Agile experience	5
4.3 Customer agile readiness factors	6
4.3.1 General	6
4.3.2 Incremental delivery and feedback	6
4.3.3 Interaction	6
4.3.4 Decision-making	7
4.3.5 Internal/external	7
4.3.6 Agile familiarity	7
4.3.7 Agreements	7
4.3.8 Quality control	8
4.3.9 Documentation	8
4.3.10 Progress reporting	8
4.4 Project team	8
4.4.1 General	8
4.4.2 Project/product team	8
4.4.3 Project type	9
4.4.4 User involvement	9
4.4.5 Size/scope	9
4.4.6 Iteration length	9
4.4.7 Criticality	9
4.5 Management	10
4.5.1 Overtime	10
4.5.2 Motivation	10
4.5.3 Quality ownership	10
4.5.4 Documentation	10
4.5.5 Mistakes	11
4.5.6 Process improvement	11
4.5.7 Progress reporting	11
4.5.8 Management transition	11
4.6 People/team	12
4.6.1 Geographic distribution	12
4.6.2 Office space	12
4.6.3 Roles	12
4.6.4 Self-organizing teams	12
4.6.5 Autonomy	12
4.6.6 Skills	13
4.6.7 Non-development funding	13
4.6.8 Training	13
4.6.9 Coaching	13
4.6.10 Perception of agile	13
4.7 Tools and practices	14

4.7.1	Communication.....	14
4.7.2	Defect management.....	14
4.7.3	Pair programming.....	14
4.7.4	Automated unit testing.....	14
4.7.5	Continuous integration and testing.....	14
4.7.6	Automated system testing.....	15
4.7.7	Specialist testing	15
4.7.8	Project management tool.....	15
Annex A (informative) Agile readiness criteria — Quick reference guide		16
Bibliography		19

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).

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

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

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

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

Agile development is a 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 (see ISO/IEC/IEEE 26515:2018).

Many organizations recognize the benefits of moving to an agile approach to systems and software development. However, for some organizations the move can be taken too early; before the organization is ready for it. This document provides insight into appropriate considerations when adopting an agile approach to software and systems development. In this document, the focus of these considerations is the agile readiness factors that can be considered before making such a move. Using this information to increase organizational and team readiness can make the difference between a successful move to agile and a failure that prevents the organization from deriving the benefits of an agile approach for several years. This document is primarily intended to be used by those managers responsible for deciding on whether a move to agile can be made and those managers who are tasked with preparing an organization for making such a move. The agile readiness factors considered in the document can be applied at the organizational level and to projects or teams within organizations.

As a Technical Report, this document contains data of a different kind from that normally published as an International Standard or Technical Specification, such as data on the “state of the art”.

Software and systems engineering — Agile development — Agile adoption considerations

1 Scope

This document provides an overview of agile readiness factors that are likely to determine whether an organization, project, product or team is ready to start the transition to using an agile approach to their system and software development and maintenance activities.

This document provides a general approach that is applicable to all agile methodologies and does not cover specific agile methodologies, such as Scrum, SAFe and eXtreme Programming (XP).

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

agile development

development approach based on *iterative development* (3.10), frequent inspection and adaptation, and incremental deliveries in which requirements and solutions evolve through collaboration in cross-functional teams and through continuous *stakeholder* (3.16) feedback

Note 1 to entry: Any use of the word “agile” in this document refers to methodology.

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.1, modified — The reference to an external annex has been removed.]

3.2

agile maturity

extent to which an organization, department, project or team consistently applies agile values and principles that contribute to the achievement of its business needs

3.3

agile team

small cross-functional group of people who collaborate on the development of a product, within an agile methodology

Note 1 to entry: A common agile team size is 3 to 10 people.

3.4

agile team lead

individual responsible for ensuring an *agile team* (3.3) adheres to the organization’s agile principles, practices, values and processes

Note 1 to entry: The agile team lead is a facilitator rather than a manager.

3.5
agreement

mutual acknowledgement of terms and conditions under which a working relationship is conducted

EXAMPLE Contract, memorandum of agreement.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.5]

3.6
customer

organization or person that receives a product or service

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.16, modified — The example and note to entry have been deleted.]

3.7
daily stand-up

short, daily, time-boxed meeting used to discuss progress, plans and any blocking issues with each member of an *agile team* (3.3)

Note 1 to entry: Duration is not expected to exceed 15 min.

3.8
increment

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

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.1913]

3.9
iteration

short time frame in which a set of *software features* (3.15) is developed, leading to a working product that can be demonstrated to *stakeholders* (3.16)

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.

Note 3 to entry: Typical iteration length is two to four weeks.

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.10, modified — Note 3 to entry has been added.]

3.10
iterative development

repeated use of concurrent planning, developing, and testing activities

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.11]

3.11
persona

model of a user with defined characteristics, based on research

[SOURCE: ISO/IEC/IEEE 26513:2017, 3.29]

3.12
product owner

stakeholder (3.16) responsible for the capabilities, acceptance and use of a product

Note 1 to entry: The product owner shares the product vision, required features and their priorities, and acceptance criteria.

3.13
release

distribution of a product *increment* (3.8) to a *customer* (3.6) or users

3.14**software feature**

software characteristic specified or implied by requirements documentation

EXAMPLE Functionality, performance, attributes, design constraints.

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.3814]

3.15**sponsor**

person or group who provides resources and support for the project, program, or portfolio and is accountable for enabling success

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.3908]

3.16**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.]

3.17**taskboard**

visual display of tasks to be completed by an *agile team* (3.3) and recent progress made by the team

3.18**user story**

simple narrative illustrating a user requirement from the perspective of a *persona* (3.11)

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.16]

4 Agile readiness factors**4.1 Overview**

The agile readiness factors presented in this document have been derived for use by an organization that wants to determine their level of readiness to adopt an agile methodology for their system and software development. Use of these factors is not intended to provide a decision on whether to move to agile, rather it is to provide guidance on how to make the journey to agile easier. By considering these factors, an organization can gain an understanding of the main risks associated with transitioning into using an agile approach and the factors they need to consider when preparing to transition towards agile ways of working.

The factors are grouped into six logical areas:

- organizational;
- customer;
- project;
- management;
- people/team;
- tools and practices.

In the following subclauses, each of the logical areas is considered in turn by identifying the individual factors relevant to the area and providing a description of how the factor can be used to assess the agile readiness of an organization or project. A quick reference guide to the factors is provided in [Annex A](#).

If an organization believes that they have most of the factors satisfactorily covered, then the organization can consider themselves ready to move to an agile development approach. If there are several factors where it is clear the organization is lacking, then the organization can first decide to address these factors before they adopt their chosen agile methodology.

For most of the agile readiness factors, it is not a simple case of satisfying them or not. There can be degrees of coverage to consider for a given factor and each organization will need to make a subjective judgement about each factor and how important it is to them. For instance, having employees that are experienced in using agile methodologies is a commonly used agile readiness factor. An organization attempting to transition to agile without any people experienced in agile is far less likely to succeed than one with many people that are experienced in agile. However, when considering this factor, an organization will be faced with making a subjective determination of the level of agile experience within their organization (both current level and the minimum level needed to mitigate the risk to transition). This can be based on how many people have agile experience, in what roles and perhaps the length of that experience, among other subjective considerations.

It is extremely rare for any organization to fully satisfy all of the listed agile readiness factors, and it is not expected that an organization would wait until every factor is fully satisfied before making the move to agile. In practice, satisfying only a few of these factors will benefit the user in most situations. With each factor there is a corresponding risk to making the agile transition and this risk will vary based on the context of the organization. Each organization will balance the risks associated with not fully satisfying a given set of readiness factors against the benefits expected to be achieved by moving to using an agile approach. Organizations can also consider the option of customising the agile practices suggested by the agile readiness factors to better fit their individual situations.

Many of the factors not only apply to determining agile readiness, but they can also be used for determining a measure of agile maturity. An organization can have already adopted an agile approach to their development and can then use the factors to help them decide on areas where they can improve, even several years after having transitioned to using an agile approach. In this case, the organization can use their experience of using an agile approach to help decide which outstanding factors are most likely to provide them with the most benefit at the least cost.

NOTE This introduction has focused on an organization's agile readiness, but most of the same factors also apply when determining the agile readiness of a department, single project or team within an organization. Where a factor is specific to assessing readiness for one or the other, it is highlighted.

4.2 Organizational agile readiness factors

4.2.1 Funding

The transition to using an agile development approach typically requires significant funding, such as for training, tooling, resourcing and to support process transition.

The move to agile is considered as a significant change management activity, and perhaps as a project in itself, and, as with any project, it needs to be resourced. Any associated transition plan will include estimates for necessary funding.

4.2.2 Management support

The transition to using an agile development approach requires management support^{[6][7]}.

Management support at all levels needs to be visible to all concerned.

If it is an organizational transition, then the support will start at board and executive management level but will also include all levels of management throughout the organizational structure.

If a project or team is transitioning to agile, then all project stakeholders, including the customer, project manager/product manager, delivery leads and participating users, need to buy into the transition.

4.2.3 Organizational change (organization)

The transition to using an agile development approach is more likely to succeed if the organization has experience of successfully making organization-wide changes in the past.

Example organizational level changes include substantial restructuring (e.g. reorganization of Google from a monolithic organization into Alphabet comprising several parts) and entering a new market area.

The move to agile will be considered as a significant change management activity, and perhaps as a project in itself, and, as with any project, it will be far more likely to succeed if the project is led by change leaders who are committed to the change.

If the organization has unsuccessfully attempted organization-wide change in the past, those relevant factors that are thought to have contributed to the failure will have been addressed in a transparent manner – otherwise they will need to be addressed during this transition to agile.

4.2.4 Change support

The move to agile needs to be visibly supported across the whole organization^[6].

Depending on the size of the organization, it is sometimes necessary to appoint a marketing role (a role that can sometimes be the responsibility of a change manager) to spread positive information about the transition. Publicity and public events about the transition, such as a transition news website and “question and answer” meetings can be organized. A change champion (or a team of champions) can be appointed as change leaders or change agents. Lack of open support for the transition to agile by the whole organization is sometimes perceived as opposition, or at least a lack of confidence in the change. Experience and capability in change management practices within the organization will also assist staff during the transition.

4.2.5 Agile experience

4.2.5.1 Agile experience (organization)

The transition to using an agile system or software development approach is more likely to succeed if the organization has experience of successfully using an agile approach in another area of the organization, such as in sales, finance or HR^[6].

If the organization has unsuccessfully attempted to transition to agile in another (non-software) part of the organization in the past, those relevant factors that are thought to have contributed to the failure will have been addressed in a transparent manner – otherwise they will need to be addressed during this transition to agile system and software development.

Moving a whole organization to using agile development is often easier if a successful pilot project is run within the organization first. The pilot project needs to be carefully selected. The lessons learned from the pilot can be used in training material, and, very importantly, the participants in the pilot need to be retained and empowered to spread their experience and motivation to other agile projects.

4.2.5.2 Agile experience (projects/products)

The use of an agile development approach on a new project or for a new product is more likely to succeed if the organization has experience of successful agile projects or product teams in the past. Using staff with experience on other successful agile projects or product teams on each new project or product team can be a major factor towards making new projects or product teams a success^[6].

If previous agile projects or product teams were unsuccessful, those factors that are thought to have contributed to the failure are addressed in a transparent manner – otherwise they will need to be addressed during this agile project or product team.

4.3 Customer agile readiness factors

4.3.1 General

The customer agile readiness factors are focused on the customer that ultimately pays for agile work to be performed. In some organizations, a marketing director, sales director, product manager or similar takes on the role of the customer and also represents the end user. A common reason that the role of the customer is taken on by someone internal to the company is if the actual end customers do not make themselves available or are many and varied. By using an internal person to take on this role, the customer can be represented by a single easily accessible person who is more familiar with the needs of the agile team. However, this will only be successful if the internal person accurately represents the customer.

Several of the factors are based on maintaining frequent and clear communications between the customer and the developers based on a significant level of trust.

NOTE Where an organizational change to agile is being considered, then those factors concerned with the customer are based on a typical customer of the organization.

4.3.2 Incremental delivery and feedback

The transition to using an agile system or software development approach will be more successful if the customer is able and willing to accept a product across a series of incremental deliveries and within agreed timescales.

NOTE Agile development approaches with frequent iterations can be successful even if releases to end users are scheduled at longer intervals for business reasons.

If the customer will only accept a complete product and will not provide feedback on intermediate deliverables, then it is unlikely that an agile approach will succeed. Conversely, if the customer is willing to receive incremental deliveries (often after having performed iterative user acceptance testing) and will also provide feedback on each increment, then this will make a positive contribution to a successful agile adoption.

Feedback can be through the customer themselves, but for many agile projects, the customer will appoint a product owner to act as their representative. In some circumstances, the customer will also identify experts in specific areas to provide feedback.

4.3.3 Interaction

The use of an agile development approach is more likely to succeed if the customer is able and willing to provide frequent and timely input to the project, such as specifying and clarifying user needs and providing feedback on proposed new functionality^[6].

In some traditional projects, the customer expects to state their full requirements up-front and then wait for the deliverable that meets all those requirements. In agile, the customer will typically still provide a description of their needs up-front but will then prioritize and expand on those requirements for each iteration (or sometimes for each release), allowing them to decide their most important needs on a frequent basis, making it more likely that the delivered product meets their current needs. Delaying the specification of requirements has to be balanced with knowing enough of the requirements or vision up-front to make the key architectural decisions early to avoid significant rework later.

When requirements are only specified in detail for the next iteration (or sometimes for each release), there is no unnecessary specification of requirements that are not needed or delivered. If developers are able to ask for clarifications of the requirements, when needed, then the requirements can be specified

in a lightweight manner, such as user stories, and it is then easier for customers to state their needs themselves. Such high-level requirements are less likely to unnecessarily constrain the developers, and frequent interaction between developers and the customer allow them to deliver solutions that strike a balance between meeting user needs quickly and the cost of development.

Where specific expertise on user requirements is required, the move to agile will more likely succeed if experts are available to explain these user needs whenever the developers need such detailed information.

4.3.4 Decision-making

The use of an agile development approach on a new project product is more likely to succeed if customers make decisions in a timely manner.

Customers need to decide what functionality is required, in what order, and whether proposed solutions are acceptable, among other things. If decisions are made more quickly, less time and effort are wasted on delivering sub-optimal deliverables.

If the customer can appoint a single, knowledgeable person (typically known as the product owner) with the authority to make decisions on behalf of the customer, and who is always available, then a move to agile is more likely to succeed. The alternative is for decisions to be made by a group, but this adds a time-lag into the decision-making, so slowing down the overall development process and increasing wasted effort.

4.3.5 Internal/external

The use of an agile development approach on a new product is more likely to succeed if a product is being built for a customer that is easily accessible and understands the organisational structure. This is more likely to be the case for an internal customer than for an external customer.

The ease of communication and the better understanding of the organization provided by an internal customer, who is also easier to reach, make it more likely that a new agile project will succeed with an internal customer rather than with an external customer. This is particularly important for the pilot project (see 4.2.5.1).

4.3.6 Agile familiarity

The transition to using an agile system or software development approach is more likely to succeed if the customer understands how agile development is supposed to work.

The Agile Manifesto and Agile Principles are often used as a starting point for this understanding^{[8][9]}.

Customer familiarity with agile will ideally come through experience as a customer from previous successful agile projects. However, when the customer lacks experience, training specifically targeted at customers of agile projects can be made available and strongly recommended.

4.3.7 Agreements

The transition to using an agile system or software development approach is more likely to succeed if the development agreement with their customer is flexible and dynamic, based on the parties agreeing the next most valuable deliverables.

The responsibilities of both the customer and the supplier are stated. The Agile Manifesto covers this with one of its four statements as valuing “customer collaboration over contract negotiation”^[8].

If the agreement for development of the product is of a fixed price type based on a pre-defined set of deliverables, then moving to agile will be more difficult. Many agreements are based on specified deliverables, but because agile uses incremental delivery, where it is not known in advance what will be delivered, it can be necessary for agreements to be changed to support the agile approach.

4.3.8 Quality control

The transition to using an agile system or software development approach is more likely to succeed if the developers do not have to implement a development process that aligns with traditional quality controls imposed by the customer, such as regular milestone reviews, audits and reviews of interim work products that do not contribute to the ultimate quality of the deliverable.

This does not mean no quality controls are needed, only that they are lean and impose no unnecessary workload on the agile team.

When quality control practices are mandatory for an organization (e.g. as is the case in regulated industries, such as banks), then the chosen agile methodology can be tailored to support the necessary audits and regulatory practices.

4.3.9 Documentation

The transition to using an agile system or software development approach is more likely to succeed if the customer is willing to accept a lightweight or lean approach to documentation that will more easily fit with an agile approach.

The Agile Manifesto covers this with one of its four statements as valuing “working software over comprehensive documentation”^[8].

If the customer expects a heavyweight documentation approach, then this can create additional work for the agile team.

4.3.10 Progress reporting

The transition to using an agile system or software development approach is more likely to succeed if the customer is willing to accept a move to progress reporting that is based more on deliverables than on documentation.

Progress reporting is required by most customers, but, ideally, it will not create an unnecessary overhead for the developers on an agile project. Verbal communication, or lightweight reports in the case of there being multiple teams, can be sufficient for most situations where the customer cannot immediately determine progress from agile information radiators, such as the burn-down chart. Progress reporting by demonstrating the functions to be delivered by the current iteration is an ideal approach for agile.

4.4 Project team

4.4.1 General

The project/product team agile readiness factors are focused on the first project or product team being run using an agile approach.

NOTE Where an organizational change to agile is being considered, then those factors concerned with the project are based on a typical project run by the organization.

4.4.2 Project/product team

The transition to using an agile system or software development approach is more likely to succeed if the organization is structured to support product teams, rather than working on projects.

The concept of projects in agile is being replaced by product teams, which are responsible for the whole product life cycle from initial concept to retirement, where the team’s commitment is to recurring delivery of a product rather than a one-pass approach. If the organization is already following the product team approach and the intention is to use agile product teams then the transition to using an agile approach will normally be easier.

4.4.3 Project type

The transition to using an agile system or software development approach is more likely to succeed if the selected project is a new project^[7].

At a high level, projects can be simply categorised as either new or maintenance projects. A new project is a new one that is starting with no existing development to build upon. This allows agile practices to be implemented on the new project without having to transition from existing non-agile practices. There will also be no constraints or compromises based on using existing non-agile deliverables, such as requirements specifications, design specifications, test specifications, code and tests, and there will be no pre-existing technical debt to overcome.

4.4.4 User involvement

The transition to using an agile system or software development approach is more likely to succeed if the expected users of the deliverables from this project are available from the start of the project to provide their input to the development.

Such input can be through providing clarification on required functionality and providing feedback on delivered functionality (e.g. through user acceptance testing). This input will ideally be constantly available, and the likelihood of success of the project will diminish as the level of user availability drops.

4.4.5 Size/scope

The transition to using an agile system or software development approach is more likely to succeed if the risk associated with the project due to its size/scope is lower.

Projects that involve multiple agile teams (i.e. using more than 10 people working together) are less likely to succeed as the project will then have the added complexity of managing multiple teams and integrating the work from these teams.

From an organizational perspective, selecting a project that has an expected duration far into the future can be problematic. It is difficult to measure success on a partially completed project and waiting several years for a result on whether the “new” agile project succeeded can mean the results, even if successful, arrive too late to be useful.

4.4.6 Iteration length

The transition to using an agile system or software development approach is more likely to succeed if the change in iteration length (often the time between when the customer asks for something and it is delivered), from the currently applied approach to an agile approach, is smaller.

This can apply if the transition to agile is for an ongoing project already following an iterative development approach, or there is an organization-wide expectation for the time between when the customer asks for something and it is delivered.

If the change in iteration length is large, then more changes in practices will be required by the new agile team. Although it is rarely possible to change the current iteration length, this risk can be partially addressed by selecting a proposed iteration length on the agile project that is not too short (e.g. select an iteration length of several weeks rather than several days).

4.4.7 Criticality

The transition to using an agile system or software development approach is more likely to succeed if the selected project is not critical^[6].

Criticality can be understood in terms of business criticality, safety criticality, financial criticality, etc. Critical projects are typically considered higher risk.

Failure of a critical project is likely to set back any transition to agile by several years. Conversely, the selected pilot project cannot be considered unimportant, as if the project succeeds it will not be considered a proper test of the agile approach.

If the project deliverable is critical enough to be regulated (e.g. safety related), then there will be associated requirements, as defined by regulatory standards. For instance, the standards can require the use of mandated documentation or practices that can make it difficult to apply typical agile practices.

4.5 Management

4.5.1 Overtime

The transition to using an agile system or software development approach is more likely to succeed if managers do not currently expect staff to regularly work overtime.

Agile practices assume working conditions for sustainable development (i.e. it can continue at the same pace without causing problems). One of the 12 principles behind the Agile Manifesto covers this: “Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.”^[9].

These working practices aim to provide a higher return on investment through higher quality. The organization will find it easier to transition to agile if the management’s expectations for those working on agile projects is that no overtime is expected of the team other than under exceptional circumstances.

4.5.2 Motivation

The transition to using an agile system or software development approach is more likely to succeed if managers within the organization currently motivate their staff at a team level.

Agile development is based on the expectation that each of the developers, testers and business analysts work as part of a cross-functional team, where the team takes priority over the individual (e.g. mistakes are team mistakes and successes are team successes). Moving to an agile approach will be easier if the current approach used within the organization is based on motivating the team rather than rewarding individuals.

4.5.3 Quality ownership

The transition to using an agile system or software development approach is more likely to succeed if the responsibility for quality within the organization is currently considered to be the joint responsibility of all members of the organization.

When following an agile approach, the responsibility for quality resides with the agile team. For organizations where the responsibility for quality is currently held by an external quality assurance team, the move to dispersing this responsibility can be difficult.

In an agile project, quality assurance is performed as an ongoing activity throughout the project. If the current approach is that quality assurance is only performed at the end of projects as part of release management (or something similar), then the move to an agile approach is likely to be more difficult.

4.5.4 Documentation

The transition to using an agile system or software development approach is more likely to succeed if the current expectation is for a lightweight (or lean) approach to internal documentation.

If the organization currently expects a heavyweight documentation approach, then the change to a lightweight or lean approach to documentation, as expected within an agile approach, can make the move to agile more difficult.

4.5.5 Mistakes

The transition to using an agile system or software development approach is more likely to succeed if management accept that making mistakes is inevitable and part of a necessary learning experience.

Ideally there will be an organizational culture that encourages producing value and failing fast, while accepting that this can result in engineers making mistakes.

A negative attitude from management towards mistakes, both at the project, team and individual levels, can be a barrier to the take-up of an agile approach. A “trial and error” approach is expected with agile, where mistakes are not just tolerated but actively encouraged as part of determining the best solution and to ongoing improvement of software and systems development practices. If the current management approach is to penalize those making mistakes, this is likely to make the move to agile more difficult. A blame culture in an organization is hard to break and is likely to impede the transition to agile.

4.5.6 Process improvement

The transition to using an agile system or software development approach is more likely to succeed if process improvement is already considered an important activity by management.

Process improvement is an integral part of agile development approaches and needs to be considered daily and at the end of each iteration. Organizations where process improvement is not considered relevant are likely to find it more difficult to transition to agile.

Even if an organization does have a process improvement activity, it can be formalized as a responsibility of separate management or quality assurance teams. When using an agile approach, everyone is equally responsible for process improvement and so this change in responsibility can be a barrier to the take-up of an agile approach in this area.

4.5.7 Progress reporting

The transition to using an agile system or software development approach is more likely to succeed if progress reporting is currently performed informally and daily and is easily visible.

Teams using an agile approach perform progress reporting through various mechanisms, such as verbally as a team each day at the daily stand-up meeting or using lightweight reports in the case of there being multiple teams, and progress is normally also made visible through information radiators, such as Kanban charts. For organizations where progress reporting is done individually to a manager, or done through submitting documentation, or not done at all, then moving to an agile approach can be more difficult.

4.5.8 Management transition

The transition to using an agile system or software development approach is more likely to succeed if both managers and team members are willing to be flexible about their responsibilities.

In agile, the teams are self-managing, thereby removing the need for one level of management within the organization, mostly at the team management level. Despite this, many of the activities performed by managers, such as staff development, are still needed – and many managers make the move to be an agile team lead, which involves skills including coaching and communication. Agile team members are also expected to take on more management tasks. For organizations with rigid structures and management career paths based on team management, it can be more difficult to transition to an agile approach within the organization.

4.6 People/team

4.6.1 Geographic distribution

The transition to using an agile system or software development approach is more likely to succeed if the agile team is co-located.

Some of the great communication advantages of an agile team are based on the ability of team members to easily interact with each other. If projects within the organization use distributed teams (e.g. developers in one city and the testers in another, or some of the team in the office while the others work from home), then the success of transitioning to agile will be put at risk.

When co-location of members of an agile team is not possible within a given office, then suitable alternative technologies can be provided to support daily communication and collaboration between team members (see 4.7.1). For instance, this can include team collaboration software that provides an online taskboard, file sharing, instant messaging/chat and audio/video meeting facilities.

4.6.2 Office space

The transition to using an agile system or software development approach is more likely to succeed if the agile team can be provided with a suitable office space.

This is likely to include the ability to work together as a team (e.g. space for stand-up meetings) and to work in isolation individually. The office space also provides plenty of wall space to put up and use a taskboard and whiteboards to track progress and discuss development decisions, etc.

4.6.3 Roles

The transition to using an agile system or software development approach is more likely to succeed if team members are comfortable performing more than a single role or taking on multiple tasks.

An agile team as a whole takes responsibility for delivering high quality functionality within fixed timescales. To achieve this, there are times when some team members take on roles which are not their primary role (e.g. a developer helping out with test automation, when needed). This fluidity of roles, which needs to be supported by the team members having the right skills, allows the agile team to be more efficient.

If the current situation within the organization is that everyone has fixed roles, then ideally this will have to change, which introduces a risk associated with changing how people work (some can be uncomfortable moving to a situation where they have to perform outside their primary role).

4.6.4 Self-organizing teams

The transition to using an agile system or software development approach is more likely to succeed if team members are willing to take on some managerial responsibilities.

A self-organizing team can allocate its own work on a daily and weekly basis and no one person has the role of team manager. The team is formed with individuals having the skills needed to do the work and the flexibility to support the team in new tasks as needed.

4.6.5 Autonomy

The transition to using an agile system or software development approach is more likely to succeed if team members are willing to work independently and manage themselves.

Members of an agile team are enabled and empowered to take responsibility and have no team manager telling them what to do, meaning they manage themselves. For some people this is liberating, and they enjoy the challenge and the freedom, for some others the change to having no manager to provide personal leadership and coaching on a daily basis can be challenging.

4.6.6 Skills

The transition to using an agile system or software development approach will only succeed if team members are familiar with agile skills (including soft skills)^{[6][7]}.

The members of the new agile team (and other relevant stakeholders) will need to know how agile works and be able to work in an agile way. If the organization does not have sufficient people with agile skills, then any transition to an agile approach is likely to fail. These skills are needed as soon as the project starts, and, in practice, an agile team needs at least two experienced members who have performed agile before, even if all the team have been trained in agile.

A more specific skill area that can be problematic when moving to an agile approach is that of architectural design skills. In traditional development, architectural design is the responsibility of specialists, who, typically, create the architectural design for the whole system early in the life cycle. With an agile approach, the architectural design can be evolving iteration by iteration – and it is the responsibility of the agile team to create (and evolve) this design. If the agile team lacks these skills, or no alternative way of injecting these skills to the team is put in place, then a successful transition to agile can be put at risk. Other useful skill areas in the agile team include domain knowledge, continuous integration and continuous deployment.

4.6.7 Non-development funding

The transition to using an agile system or software development approach is more likely to succeed if funding for those stakeholders who are not members of the agile team is provided.

For instance, funding for users and experts to fulfil their roles in the agile development process (e.g. providing expertise in user requirements and taking part in user acceptance testing) is explicitly identified as part of the estimate of agile project costs and made available to these stakeholders so that they can fulfil their obligations. Lack of funding for stakeholders outside the core agile team and the consequent difficulty in getting them to participate in important activities, such as acceptance testing, is a high risk for a successful agile transition.

4.6.8 Training

The transition to using an agile system or software development approach will only succeed if training in agile is made available.

Training in agile, aligned with the selected methodology, customized for the organization, and targeted at specific stakeholder groups is made available, and perhaps mandatory, for all stakeholders. It is made available well in advance of the first project product delivery starting. Training for the agile team alone is not sufficient; training is also appropriate for a variety of other stakeholders, including the customer, users and the product owner.

4.6.9 Coaching

The transition to using an agile system or software development approach is more likely to succeed if agile coaches are made available to agile teams.

Agile coaches, who are familiar with the selected agile methodology, are appointed to support the agile teams working on the first set of project/product delivery adopting an agile approach. Coaches need to be involved from as soon as the project delivery starts. If this coaching is not available when needed, then the transition to agile will be put at risk.

4.6.10 Perception of agile

The transition to using an agile system or software development approach will only succeed if those making the transition believe in it.

The transition to using new agile practices will only work if those making this journey start it with a positive perception of the benefits of agile and if they continue to focus on the benefits of agile if/when individuals resist the transition to agile.

4.7 Tools and practices

4.7.1 Communication

The transition to using an agile system or software development approach is more likely to succeed if communication between team members (and teams) is seamless^[6].

Where distributed agile teams are used (see 4.6.1), then it is important that they can communicate easily. Various channels and levels of communication can be provided (e.g. audio, video, on demand, when booked), each giving varying levels of support.

4.7.2 Defect management

The transition to using an agile system or software development approach is more likely to succeed if defect management is supported by tools.

With an agile approach, managing defects is typically treated similarly to any other task performed by the agile team. Defects can be prioritized so that they are not addressed until some time in the future, and with cross-functional teams it is unlikely one team member will have responsibility for fixing defects. This lag in addressing the defects, and with multiple team members taking responsibility for them, means defects are best managed using an automated defect management system.

Where distributed agile teams are used (see 4.6.1), then it is important that any detected defects are shared quickly and accurately between the agile team members. A defect management tool can provide the needed traceability between both test cases and user stories.

4.7.3 Pair programming

The transition to using an agile system or software development approach is more likely to succeed if pair programming is practised by developers.

Pair programming is a development practice for writing code, where two programmers work collaboratively on the same task, sitting side by side at one computer, so allowing immediate peer review. Its benefits include increased code quality, enhanced sharing of knowledge about the code and improved programming skills. Surprisingly, using two programmers working together on a single task does not double the effort used, but only increases it by about 15 % due to their improved efficiency^[10]. Having programmers with experience of this practice makes the transition to an agile development approach easier.

4.7.4 Automated unit testing

The transition to using an agile system or software development approach is more likely to succeed if automated unit testing is practised.

Automated unit testing helps ensure unit tests achieve an agreed level of code coverage, while also supporting the efficacy of the practice of continuous integration and testing. Unit test tools for the relevant language are used to implement it. Having programmers or testers with experience in this practice and the capability to apply it makes the transition to an agile development approach easier, however, this will only be true if the programmers are experienced at achieving high levels of coverage.

4.7.5 Continuous integration and testing

The transition to using an agile system or software development approach is more likely to succeed if continuous integration and testing is used by the developers.

Continuous integration and suitable automated unit tests allow multiple developers to work on a shared codebase with the assurance that any new code they submit is automatically checked to ensure that it has not broken existing functionality (that is tested in the regression tests). If defects are detected, then they are normally easily identified allowing them to be fixed, ideally before other developers are affected. Continuous integration and testing is a “best practice” in software development, and a mainstay of agile development; not using it would put at risk a successful move into using an agile approach.

4.7.6 Automated system testing

The transition to using an agile system or software development approach is more likely to succeed if automated system testing is practised.

Automated system testing helps ensure that when changes are made regression tests can be automatically run to check that other parts of the system are unaffected by the change. System test tools for the relevant language are used to implement it but need to provide an adequate return on investment. Having programmers or testers with experience in this practice and the capability to apply it makes the transition to an agile development approach easier.

4.7.7 Specialist testing

The transition to using an agile system or software development approach is more likely to succeed if agile team members have expertise in specialist test types and if those test types are applied in each iteration, such as performance testing, security testing and accessibility testing.

Testers on agile teams benefit from having experience of various specialist test types and access to specialist test environments if they are to be able to run specialist tests (e.g. performance and security tests) in each iteration. Alternatively, they need access to specialist testing as a service. These tests need to be run if functionality is to be delivered by the iteration, as is expected with an agile approach. Having testers with experience of the relevant specialist test types and the tools to run the tests makes the transition to an agile development approach easier.

4.7.8 Project management tool

The transition to using an agile system or software development approach is more likely to succeed if the agile team have access to an agile project management tool.

With multiple requirements in user stories, short iterations and developers working in parallel, it is very useful to have a project management tool when using an agile approach. It normally becomes necessary if the agile development is distributed (see [4.6.1](#)). If an agile project management tool is currently being used, and it is suitable for use on agile project/product delivery, then this makes the transition to an agile development approach easier.

Annex A

(informative)

Agile readiness criteria — Quick reference guide

[Table A.1](#) provides a quick reference guide for the agile readiness criteria presented in this document. It provides questions to consider when looking at readiness to transition to agile, based on the considerations presented in [Clause 4](#).

Table A.1 — Agile readiness criteria

Area	Factor	Considerations
Organizational	Funding	Has funding been allocated to support the transition to agile? Is there a transition plan and does it include estimates for funding?
	Management support	Does management support the transition to agile, including at a board, executive management and agile team leadership level?
	Organizational change (organization)	Has the organization succeeded in other organization-wide transitions in the past? Can any previous failures be addressed in the agile transition?
	Change support	Is the move to agile visibly supported across the whole organization?
	Agile experience (organization)	Does the organization have experience of successfully using agile in some parts of the organization? Has the organization had failed attempts to transition to agile in the past? Can these failures be overcome in this transition to agile? Can a pilot product be selected to trial this transition to agile?
	Agile experience (projects/product teams)	Has the organization successfully run agile product delivery in the past? Can any past failures be overcome in the current transition to agile?
Customer	Incremental delivery and feedback	Is the customer willing to accept the product across a series of incremental deliveries? Can the customer or a product owner provide feedback on each increment?
	Interaction	Is the customer able and willing to provide frequent and timely input into the project, such as specifying and clarifying user needs and providing feedback on proposed new functionality?

Table A.1 (continued)

Area	Factor	Considerations
	Decision-making	Is the customer able to make decisions in a timely manner? If not, can a product owner be appointed to make timely decisions on behalf of the customer?
	Internal/external	Is the product being built for an internal customer? Can an internal project be chosen for the first pilot of agile?
	Agile familiarity	Does the customer understand how the chosen agile methodology works?
	Contracts	Are contracts with vendors flexible, to support the dynamic changes that occur in incremental agile delivery? Are responsibilities agreed?
	Quality control	Can organizational quality control requirements, such as milestone reviews, audits and reviews of interim work products, be made to work within the chosen agile methodology?
	Documentation	Is the customer willing to accept a lightweight or lean approach to system documentation?
	Progress reporting	Is the customer willing to accept a move to progress reporting that is based more on deliverables than documentation?
Project/product team	Project/product team	Are the teams currently organized by project or product?
	Project type	Is the selected project a new project?
	User involvement	Are end users of the deliverables available from the start of the product delivery to provide input during development?
	Size/scope	Is the project considered high risk due to its size/scope?
	Iteration length	Are iteration lengths small (e.g. two weeks)?
	Criticality	Is the pilot project that is implementing agile highly critical (business, safety, regulated)?
Management	Overtime	Do managers in the organization ensure their staff do not work overtime regularly, to ensure sustainable working practices?
	Motivation	Do managers within the organization aim to motivate teams, rather than individuals?
	Quality ownership	Is quality everyone's responsibility?
	Documentation	Does the organization support lightweight or lean approaches to documentation?
	Mistakes	Does the organization see mistakes as a positive opportunity to experiment, learn and select the best approaches and improve the development methodology?
	Process improvement	Do management already see process improvement as an important activity?
	Progress reporting	Is progress reporting currently performed informally and daily within the organization?
	Management transition	Are managers and team members willing to be flexible about their responsibilities?

Table A.1 (continued)

Area	Factor	Considerations
People/team	Geographic distribution	Will agile teams be co-located? If not, then can collaboration systems be used to support daily communication and collaboration?
	Office space	Do co-located teams have a suitable office space, such as with taskboards, whiteboards and space to run stand-up meetings?
	Roles	Are team members comfortable performing more than a single role?
	Self-organizing teams	Are team members willing to take on some managerial responsibilities?
	Autonomy	Are team members willing to manage themselves?
	Skills	Are team members familiar with agile skills?
	Non-development funding	Is there funding for stakeholders that are not members of the core agile team, such as users participating in testing?
	Training	Will training in agile be made available to staff throughout the organization?
	Coaching	Will agile coaches be made available to agile teams?
	Perception of agile	Do those transitioning to agile believe in agile?
Tools and practices	Communication	Is communication between agile team members and between agile teams seamless?
	Defect management	Are distributed teams supported by defect management tools?
	Pair programming	Is pair programming practised by developers?
	Automated unit testing	Is automated unit testing practised by developers?
	Continuous integration and testing	Do developers perform continuous integration and testing?
	Automated system testing	Is automated system testing practised by developers?
	Specialist testing	Does the agile team have experience in specialist testing (e.g. performance testing, security testing, accessibility testing) or access to a specialist testing service?
	Project management tool	Is an agile project management tool in use?

Bibliography

- [1] ISO/IEC/IEEE 12207:2017, *Systems and software engineering — Software life cycle processes*
- [2] ISO/IEC/IEEE 15288:2015, *Systems and software engineering — System life cycle processes*
- [3] ISO/IEC/IEEE 24765:2017, *Systems and software engineering — Vocabulary*
- [4] ISO/IEC/IEEE 26513:2017, *Systems and software engineering — Requirements for testers and reviewers of information for users*
- [5] ISO/IEC/IEEE 26515:2018, *Systems and software engineering — Developing information for users in an agile environment*
- [6] 14th Annual State of Agile Report, 2020 [viewed June 2020]. Available from: <https://explore.digital.ai/state-of-agile/14th-annual-state-of-agile-report>
- [7] KPMG Global Agile Survey 2019 [viewed June 2020]. Available from: <https://assets.kpmg/content/dam/kpmg/be/pdf/2019/11/agile-transformation.pdf>
- [8] Beck K. et al. The Agile Manifesto. *Agile Alliance*, 2001 [viewed June 2020]. Available from: <http://agilemanifesto.org/>
- [9] Beck K. et al. Principles behind the Agile Manifesto. *Agile Alliance*, 2001 [viewed June 2020]. Available from: <http://agilemanifesto.org/>
- [10] Williams, L., Kessler, R.R., Cunningham, W., Jeffries, R. Strengthening the case for pair programming. *IEEE Software*. 2000, 17(4), pp. 19–25. doi: 10.1109/52.854064

