Technical Specification

ISO/IEC TS 20000-15

First edition
2024-05

Information technology — Service management —

Part 15: Guidance on the application of Agile and DevOps principles in a service management system

Technologies de l'information – Gestion des services

Partie 15: Lignes directrices pour l'application des principes "Agile" et "DevOp"s dans un système de gestion des services

Reference number
ISO/IEC TS 20000-15:2024(en)

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

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 40, *IT Service Management and IT Governance.*

A list of all parts in the ISO/IEC 20000 series can be found on the ISO and IEC websites.

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

This document is intended to assist users in relating the requirements specified in ISO/IEC 20000–1:2018 to the principles and practices of two frequently used software and service development frameworks, Agile and DevOps. Organizations can refer to this guidance as a cross-reference between the frameworks to help them plan, implement and improve a service management system (SMS).

ISO/IEC 20000–1 is the International Standard for service management and specifies requirements which can be used as the basis of a conformity assessment.

ISO/IEC 20000–1 specifies an integrated process approach in which an organization establishes, implements, maintains and continually improves an SMS. The services can be delivered to internal or external customers or a combination of both. Other parts of the ISO/IEC 20000 series provide supporting guidance.

Agile is defined as a collection of frameworks and techniques focusing on collaboration, iterative and incremental development and continuous improvement.

DevOps is defined as a set of principles and practices which enable better communication and collaboration between relevant stakeholders for the purpose of specifying, developing and operating software and systems services, and continuous improvements in all aspects of the lifecycle (ISO/IEC/IEEE 32675).

Despite these definitions being focused on software development, both Agile and DevOps principles have also been used in a much broader sense, including the development and delivery of services.

The DevOps framework is based on the Agile framework, adding automation of service development and delivery to it. Many Agile concepts discussed in this document are therefore equally applicable to DevOps.

Organizations can implement and improve their SMS using the requirements specified in ISO/IEC 20000–1 and the guidance in the other parts of the ISO/IEC 20000 series. An organization can adopt Agile and DevOps practices to support the management of their services in alignment with the requirements specified in ISO/IEC 20000–1. Other frameworks and practices can also be used to support ISO/IEC 20000–1.

Within this document, Clause 4 provides an overview of ISO/IEC 20000-1 and the SMS. Clause 5 applies Agile principles to the SMS. Clause 6 applies DevOps principles to the SMS. In Clause 7, the benefits and caveats surrounding the use of Agile, DevOps or a combination of the two in the SMS are discussed. Annex A provides a correlation of ISO/IEC 20000-1 clauses to the Agile and DevOps frameworks.

Information technology — Service management —

Part 15: Guidance on the application of Agile and DevOps principles in a service management system

1 Scope

This document provides guidance on the relationship between ISO/IEC 20000–1:2018 and two commonly used frameworks, Agile and DevOps. It can be used by any organization or person wishing to understand how Agile and DevOps can be used with ISO/IEC 20000–1, including:

a) an organization that has demonstrated or intends to demonstrate conformity to the requirements specified in ISO/IEC 20000–1 and is seeking guidance on the use of Agile or DevOps to establish or improve the SMS and the services;

b) an organization that already uses Agile or DevOps and is seeking guidance on how Agile or DevOps can be used to support efforts to demonstrate conformity to the requirements specified in ISO/IEC 20000–1;

c) an assessor or auditor who wishes to understand the use of Agile or DevOps as a support for achieving the requirements specified in ISO/IEC 20000–1.

Both approaches can be used independently or together. Depending on the context, an organization can deploy Agile frameworks only, DevOps frameworks only, use both Agile and DevOps frameworks in isolation, or use an integrated workflow with combined Agile and DevOps approaches. In any of these situations, this document can be used as guidance for the integration of Agile and DevOps practices in an SMS.

The guidance in this document can assist an organization in planning and preparing for a conformity assessment against ISO/IEC 20000-1, noting that an organization can only claim conformity by fulfilling all requirements specified in ISO/IEC 20000-1.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 20000-1, *Information technology — Service management — Part 1: Service management system requirements*

ISO/IEC 33202:—[1)], *Software and systems engineering — Core Agile practices*

ISO/IEC/IEEE 32675, *Information technology — DevOps — Building reliable and secure systems including application build, package and deployment*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 20000-1, ISO/IEC 33202, ISO/IEC/IEEE 32675 and the following apply.

1) Under preparation. Stage at the time of publication: ISO/IEC/FDIS 33202:2024.

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
collection of frameworks and techniques focusing on collaboration, iterative and incremental development and continuous improvement

Note 1 to entry: In this context, the term "Agile" is usually capitalized.

3.2
bottom-up intelligence
use of information coming from users themselves, so they can develop better options to achieve valuable objectives

3.3
continuous everything
increasing agility throughout the service lifecycle from testing, deployment and monitoring through to integration and delivery

3.4
cradle-to-grave
activities from the beginning of the service lifecycle to its end or disposal

Note 1 to entry: This term is specifically used in this way within the context of DevOps.

3.5
cross-functional autonomous team
team that has the skills and disciplines required to achieve an established goal, such as developing, deploying or operating a service

Note 1 to entry: These teams are fully empowered and self-sufficient for the designing, building, testing, deployment and running of the service.

3.6
customer-centric
doing business and ensuring a positive customer experience at every stage of the *customer journey* (3.7)

Note 1 to entry: When a customer-centric organization makes a decision, its people thoroughly analyze its impact on the customers.

3.7
customer journey
series or sum of customer experiences when engaging with an organization, its products or services

Note 1 to entry: "series" is based on processes; "sum" is based on results.

[SOURCE: ISO 23592:2021, 3.8]

3.8
daily stand-up
short, daily, time-limited meeting used to discuss progress, plans and any blocking issues with each member of an *Agile* (3.1) team

[SOURCE: ISO/IEC TR 24587:2021, 3.7, modified — The term "time-boxed" has been replaced by "time-limited" at the beginning of the definition, and "Agile" has been capitalized. Note 1 to entry has been removed.]

3.9
DevOps
set of principles and practices which enable better communication and collaboration between relevant stakeholders for the purpose of specifying, developing, and operating software and systems products and services, and continuous improvements in all aspects of the lifecycle

[SOURCE: ISO/IEC/IEEE 32675:2022, 3.1.1]

3.10
hypothesis
theory that something can become valuable, even though this will not be known for sure until it is verified in a real environment

3.11
left-shift
shift-left
prioritizing the involvement of relevant stakeholders in applying quality activities, security, privacy, performance, verification, and validation earlier in the lifecycle

Note 1 to entry: In this document, the expression “shift-left” is used, because it is more common in the industry.

[SOURCE: ISO/IEC/IEEE 32675:2022, 3.1.2, modified — Admitted term "shift-left" has been added. Note 1 to entry has been added.]

3.12
minimum viable service
MVS
limited service release that includes the main *hypotheses* (3.10) that demonstrate whether the main product idea makes sense to the customer

3.13
result-oriented plan
plan that focuses on outcome rather than on the process used to deliver a service

Note 1 to entry: Following a result-oriented plan gives a higher chance of being successful. It pushes the organization to take ownership and be flexible in defining priorities. Most importantly, it enables the organization to measure progress against a defined set of requirements.

3.14
retrospective
team meeting at the end of an iterative cycle or at the end of a project to reflect on what went well, what was learned, and what should be done differently next time

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.3488, modified — Preferred term has been changed from "retrospective meeting" to "retrospective". "Software project" has been changed to "project" in the definition.]

3.15
self-organizing team
team using its own knowledge to determine how best to do their job

3.16
servant leader
leader focusing on providing what the team need, removing impediments to their progress and supporting their productivity

3.17
service-thinking
focusing on the end-goal or the outcome of the process or service

3.18
service backlog
list of items ordered by value of what is to be done or to be achieved

3.19
service owner
person responsible for maximizing the value that the service development team creates, including managing the *service backlog* (3.18), identifying and prioritizing improvement opportunities and supporting operations.

Note 1 to entry: Within the context of this document, this term is used specifically in reference to Agile environments. Its use can be different within other service management environments.

3.20
technical debt
deferred cost of work not done at an earlier point in the service lifecycle

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.4181, modified — "product life cycle" has been replaced with “service lifecycle".]

3.21
user story
brief description of required functionality describing the stakeholder roles, goals, benefits and motivation

[SOURCE: ISO/IEC 33202:—, 3.28, modified — Note 1 to entry has been removed.]

3.22
vanity metrics
statistics that look spectacular on the surface but do not necessarily translate to any meaningful business results

4 Introduction to ISO/IEC 20000–1:2018

ISO/IEC 20000-1 specifies requirements for establishing, implementing, maintaining and continually improving a service management system (SMS). An SMS supports the management of the service lifecycle, including the planning, design, transition, delivery and improvement of services, which meet agreed requirements and deliver value for customers, users and the organization delivering the services. The organization in the scope of the SMS can be a whole or part of a larger organization and can also be known as the "service provider".

ISO/IEC 20000-1 is intentionally independent of specific guidance. The organization can use a combination of generally accepted frameworks (e.g. Agile, DevOps) and its own experience. Appropriate tools for service management can be used to support the SMS.

All requirements specified in ISO/IEC 20000-1 are generic and are intended to be applicable to all organizations, regardless of the organization’s type or size, or the nature of the services delivered. While ISO/IEC 20000-1 can be used regardless of the organization’s type or size, or the nature of the services delivered, the document has its roots in IT. It is intended for service management of services using technology and digital information. The examples given in this document illustrate a variety of uses of ISO/IEC 20000-1.

Exclusion of any of the requirements in ISO/IEC 20000-1:2018, Clauses 4 to 10, is not acceptable when the organization claims conformity to ISO/IEC 20000-1, irrespective of the nature of the organization.

The organization cannot demonstrate conformity to the requirements specified in ISO/IEC 20000-1 if other parties are used to provide or operate all services, service components or processes within the scope of the SMS.

ISO/IEC 20000-10 includes the concepts for an SMS, the vocabulary used for the ISO/IEC 20000 series, a description of each part of the series and related standards.

Guidance is available in other parts of the ISO/IEC 20000 series in the form of:

— ISO/IEC 20000-2, *Guidance on the application of service management systems*;

— ISO/IEC 20000-3, *Guidance on scope definition and applicability of ISO/IEC 20000-1*;

— ISO/IEC TS 20000-5, *Implementation guidance for ISO/IEC 20000-1*;

— ISO/IEC 20000-6, *Requirements for bodies providing audit and certification of service management systems*;

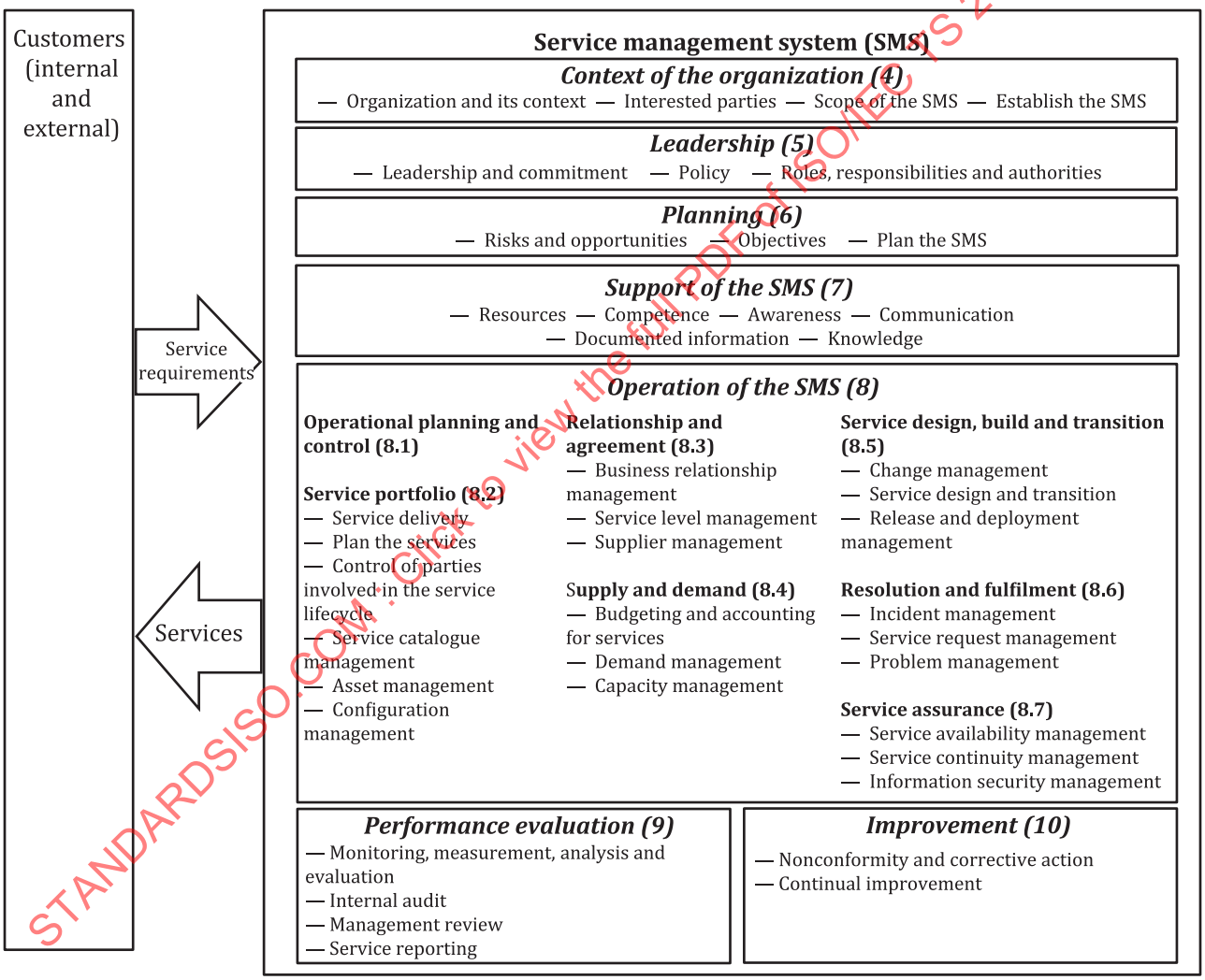
— ISO/IEC TS 20000-11, *Guidance on the relationship between ISO/IEC 20000-1 and service management frameworks: ITIL®*;

— ISO/IEC TS 20000-14, *Guidance on the application of Service Integration and Management to ISO/IEC 20000-1*;

— ISO/IEC TS 20000-16:—[2], *Guidance on sustainability within a service management system based on ISO/IEC 20000-1*;

— ISO/IEC TR 20000-17:—[3], *Scenarios for the practical application of ISO/IEC 20000-1*.

Figure 1 illustrates an SMS showing the clause content of ISO/IEC 20000-1.



NOTE Numbers in parentheses indicate ISO/IEC 20000-1 clause numbers.

Figure 1 — Service management system

2) Under preparation. Stage at the time of development: ISO/IEC WD TS 20000-16:2024.

3) Under preparation. Stage at the time of development: ISO/IEC CD TR 20000-17:2024.

5 Agile within an SMS based on ISO/IEC 20000–1

5.1 Background of Agile

To understand how Agile can help in service management, it is necessary to understand why Agile arose and the problem it was aiming to solve.

In its initial conception, Agile emerged to respond to a common problem identified in software development. In essence, it provided some guiding principles for answering the following question:

— How can one deliver valuable software to a customer when not even the customer is sure about what they want?

In other words, Agile focuses on customer satisfaction above all else, but in an unpredictable environment, also known as a high-uncertainty environment or a complex system. Value, the customer and uncertainty are the key points from which all Agile philosophical principles stem. These principles are summarized in the Agile Manifesto[13] (see 5.2), and they are the source of inspiration for all Agile frameworks that exist today.

From the key points of the Agile Manifesto, a series of Agile Principles are derived (see 5.2).

Although Agile emerged to respond to a common problem that exists in software development, its benefits have extended far beyond this discipline, and it has been tested in different sectors and contexts, some of them far removed from the world of software, such as education or health.

In the following subclauses, several Agile concepts will be discussed and applied to the requirements of ISO/IEC 20000-1.

5.2 Agile Principles

5.2.1 Overview

The Agile mindset has been expressed in terms of the Agile Manifesto and its related Principles (see Reference [13]). This subclause contains a summary of the Agile Manifesto and the Agile Principles. The Agile Manifesto and Agile Principles were written with software development in mind, but they can be interpreted for the context of services.

NOTE In the following subclauses, the term "software" as used in Reference [13] has been replaced with the term "service".

5.2.2 Agile Manifesto

The focus of Agile is to uncover better ways of delivering services by providing those services and helping others also to provide services. This work has led to the valuation of:

— individuals and interactions over processes and tools;

— working services over comprehensive documentation;

— customer collaboration over contract negotiation;

— responding to change over following a plan.

Within this list, whilst the primary points (first element in each line) are considered the most important, the secondary elements (second element in each line) are also considered valuable.

NOTE See Reference [13] for further details and original text.

5.2.3 List of Agile Principles

The twelve Agile Principles are listed as follows:[13]

1) "Our highest priority is to satisfy the customer through early and continuous delivery of valuable services."

2) "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage."

3) "Deliver working service enhancements frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale."

4) "Business people and the service provider must work together daily throughout the service lifecycle."

5) "Build services around motivated individuals. Give them the environment and support they need, and trust them to get the job done."

6) "The most efficient and effective method of conveying information to and within service teams is face-to-face conversation."

7) "Working services are the primary measure of progress."

8) "Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely."

9) "Continuous attention to technical excellence and good design enhances agility."

10) "Simplicity (the art of maximizing the amount of work not done) is essential."

11) "The best architectures, requirements and designs emerge from self-organizing teams."

12) "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly."

Agile principles and practices have been used successfully in areas such as product development, service development and other disciplines in a fast-moving, complex environment. In such a context, there is a significant level of uncertainty in the scope of the project. To manage this uncertainty, Agile focuses on the rapid release of a partial, but functional service, which is built upon in subsequent iterations until completion. The customer is continually involved in the development to test and verify the desired functionality. After each release, the work methodology and resulting functionality is evaluated and, if needed, improved. The overall focus of an Agile approach is on the iterative and incremental creation of value for the organization and for the customer through development of the service. This approach has led to the development of a number of Agile frameworks [such as Scrum, Scaled Agile Framework (SAFe) and dynamic systems development method (DSDM)], which all have commonalities that are described and can be applied to the SMS in the following subclauses.

5.3 Agile and services

5.3.1 Value in a service environment

Within the services domain, there is an interest in maximizing the value that the service delivers to the customer in environments of high uncertainty. The concepts of value, customer and uncertainty should encourage reflection on how to approach the design, implementation and maintenance of the SMS and the services.

The same issues that brought about the rise of Agile in software development can also be relevant to services, for example:

— how to offer a valuable service to a customer without knowing if the service will help to best meet the customer's objectives;

— how to know if a service level agreement (SLA) or a key performance indicator (KPI) really adds value to the end-customer;

— how to make service management evolve for the better, always keeping the customer's objectives in mind.

ISO/IEC 20000-1:2018, 5.1, requires that top management ensure that what constitutes value for the organization and its customers is determined. Customers often know what problems they seek to solve or which new opportunities they want to take advantage of, but it is also common that an alternative solution or opportunity for increased value is not identified. In such cases, Agile hypothesis and experimentation can identify new opportunities to better serve the customers' needs. Top management should therefore encourage the organization's service portfolio (see ISO/IEC 20000-1:2018, 8.2) to meet the customers' needs; an Agile approach can assist in identifying and exploiting previously undiscovered approaches.

The service that an organization delivers to a customer is itself a way of delivering value to the customer. However, uncertainty is part of the reality of services. Ultimately, it is not possible to tell if a theoretical, lab-designed, "perfect" service will actually fix problems or provide new opportunities for the customer until it is implemented in a real-life environment.

Uncertainty relates to the management of issues and risks. ISO/IEC 20000-1:2018, 4.1 requires the organization to consider internal and external issues affecting its ability to achieve the outcomes of the SMS. ISO/IEC 20000-1:2018, 6.1 requires the organization to consider these issues and determine any risks and opportunities that need to be addressed. An uncertain environment is a source of external and internal issues that can constitute risks that require some form of actions to be addressed. Introducing the Agile framework can be a way to address issues and risks resulting from an uncertain environment.

5.3.2 Value, customers and uncertainty in services

Services should be supportive of the customer's business outcomes. Value creation is the primary objective of setting up an SMS, as mentioned in the Introduction of ISO/IEC 20000-1:2018. However, it is possible for an SMS to become too process-focused, losing the focus on the customer. ISO/IEC 20000-1:2018, Clause 4, requires the organization to look at its context to support the design of the SMS which in turn helps to avoid a purely process viewpoint and supports the reduction of risk associated with uncertainty. Essentially, the organization should focus its efforts upon maximizing the value delivered to the client and in doing so should find the right balance between the internal process focus and the focus on the customer's needs and expectations.

A service implemented in the real world with which the customer is satisfied is the best way to prove that the service is valuable. With this in mind, ISO/IEC 20000-1:2018, 8.3.2, requires the organization to measure satisfaction with the services based on a representative sample of customers. However, the activities carried out within a service lifecycle are not always perceived as valuable by the customer. For example, it is possible that the customer will not perceive value in configuration management which is largely invisible to them but is essential to the smooth operation of a service. It can therefore be necessary to perform internal procedures, write documentation or follow a methodology for the customer to ultimately perceive the value of a service element. Ultimately, the customer perceives value if the service they receive solves a problem or helps them to realize an opportunity. It is this value that the organization should determine and use as guidance for the development of the SMS and the services.

With this in mind, all activities carried out as part of an SMS can be questioned at any time from the perspective of whether or not they are truly necessary and have a real value contribution or, to the contrary, are simply tasks to be adapted, or even eliminated, without this entailing a reduction in the value delivered to the customer. The service management plan, as described in ISO/IEC 20000-1:2018, 6.3, is intended to document what is necessary in an SMS to create value for the customers and the organization itself.

In Agile, value creation includes making sure that the organization offers the customer value using processes that are designed for what the customers want to achieve with the services. This is distinct from offering only services that provide basic functionality, but not much value to customers, with the rationale that they are easy for the service provider to implement.

Using Agile methodologies can help to accelerate value creation for the customer and the service provider in the following ways:

— through closer collaboration between the service provider and the customer, the customer's needs and expectations of the service are more clearly defined and implemented in the service;

— through iterative service development, service elements become available more quickly to the end users;

— through more efficient and effective testing methods, the quality and reliability of the service is improved;

— through the use of hypothesis testing (see 5.6), the elements which are valuable for the customer can be more accurately determined.

5.4 Defining objectives

Setting objectives seeks to add "certainty" to an environment of "uncertainty". A clear objective should be understood and shared by customer and the organization. This is essential for working effectively on the development of the service.

Agile's focus on value creation leads to a focus on the customer: the customer is after all the receiver of the value created by the services. The organization should listen carefully to the customer and take feedback about the provided services with the aim of improving them. In ISO/IEC 20000-1:2018, 5.1, top management is considered responsible for ensuring that what constitutes value for the organization and its customers is determined.

Leveraging Agile, the design of a service should start with the definition of a customer objective:

— What problem is to be solved for the customer?

— How is it possible to make the customer happier?

— What advantages can be given to them?

In this way, the customer can play an active role in determining the nature and shape of the services they are willing to pay for.

The customer objective then acts as a source of inspiration. For this to hold true, service and service management objectives (see ISO/IEC 20000-1:2018, 6.2) should focus on why that service and its supporting SMS are necessary. There is sometimes a tendency to define the service objectives in terms of what should be done, or the activities to be carried out. But it is preferable to ask "why is it important to perform these activities?" in order to achieve the true objective.

EXAMPLE 1 A service objective is initially defined as "we want to offer an incident ticketing service to our users based on a cloud system". When asking, "why is it important that a cloud system is used?", the answer can be "having a cloud system is not essential. The goal is to facilitate the day-to-day work of users with respect to ticket management". In other words, the location of the ticket system is not important to the users; what they want is for incident ticket management to improve so that their daily lives can be more efficient and comfortable, and so that communication between the people involved can improve.

Presenting the service objective as initially shown in Example 1 forces the choice of a cloud system and does not foster reflection on what the customer's real problem is and how best to solve it. By questioning the purpose behind the initial objective, as shown in the example, focus can be placed on understanding the customer's needs and expectations first (see ISO/IEC 20000-1:2018, 4.2), before moving on to proposing solutions. This latter approach is more inspiring and provides greater flexibility for generating potential solutions in the way the SMS and the services are developed.

Example 2 provides a continuation of Example 1.

EXAMPLE 2 During the implementation of the incident ticketing service, an end-user training gap is detected that has much more importance for achieving the customer's objective than implementing a new cloud-based technology, as specified initially. Setting an objective based on the issue to be solved allows for greater flexibility, as well as for the prioritization of new opportunities that will potentially arise later on, instead of forcing the team to carry on with what was initially planned (or even documented in an agreement). In this case, being flexible, adapting the proposal and focusing on training users over other activities can bring the organization much closer to offering a service that is truly valuable for the customer.

Agile achieves an understanding of the customer's requirements by closely working with the customer to determine what value they want to create with the service and how these expectations translate into service requirements. By iteratively implementing service features based on these requirements and having these tested by the customer, it can be verified whether or not the service really meets the customer's expectations. If it does not, or if the customer's requirements have changed, the next iteration can be used to modify service features to better meet the customer's requirements. Using short delivery iterations in this way can significantly reduce the overall risk of delivering a service a long time after the customer has given their initial requirements and not meeting them in the end.

While objectives arising from customer issues or perceived opportunities do not usually change from one day to the next, it is possible for objectives to change and become obsolete. In such case, as the priority is always to deliver the maximum value to the customer, rethinking the service and giving it a new approach to satisfy the new objective is not synonymous with failure. On the contrary, it shows strategic intelligence, and that it is understood that the work is carried out in complex systems. It also shows that delivering value to the customer and the organization is the key priority and that the only way to deliver value is by adapting the service to changes in the environment as early as possible. This supports the principle of continual improvement (see ISO/IEC 20000-1:2018, 10.2).

A focus on customer collaboration also impacts and elevates the role of the business relationship manager (BRM)(see ISO/IEC 20000-1:2018, 8.3.2). In an Agile environment, the role of the BRM is similar to that of a service owner on the service provider's side. This shares aspects of the product owner's role in Agile. In Agile, a product (or service) owner is responsible for maximizing the value that the team delivers. As such, the service owner is also responsible for managing the service backlog, which is a list of user stories ordered by value of what is to be done or to be achieved. With these responsibilities in mind, the service owner works closely with the stakeholders in order to know and manage their expectations, and closely with customer or end user, performing market analysis, examining competition and handling many other activities. The same responsibilities should be performed in a service context so that the service always offers the maximum possible value.

The service backlog is used to gather user stories for the Agile team to work on. User stories can include new service elements or change requests, as well as work on incident resolution and root-cause analysis for problem management. Work items in the service backlog are prioritized by the team based on the urgency indicated by the end users. Incidents may therefore be prioritized over problems, which in turn may be prioritized over changes. The principle of working on several work items in the available time in an iteration remains the same, irrespective of the nature of the work.

5.5 High-uncertainty contracts

Service agreements with the customer should have increased focus on maintaining the relationship with that customer.

In a service agreement, it is common to try to specify the details of the service that the organization has agreed to provide to the customer. However, considering that the service may be implemented in high-uncertainty environments, it is not possible to anticipate the exact details of the service that the organization will provide to the customer in a reliable way. In other words, trying to anticipate the details of the service in an agreement will be a barrier in the relationship between both parties, making it difficult to adapt the best solutions and inevitably leading to a decrease in the value delivered to the customer.

Therefore, the key action is to separate the certain from the uncertain. While it is true that there is an element of uncertainty that makes it impossible to anticipate all details (which would make writing such details into an agreement inadvisable), there are always elements of certainty to focus on. For example, the

Agile Manifesto introduces concepts such as the necessary collaboration between the customer and the organization, the importance of the quality of the service that is delivered, the value of direct communication, the team's need for autonomy when making decisions, the benefits of frequent deliveries and of incorporating flexibility to changes, etc. These are key points when it comes to being Agile, and they should be anticipated and reflected in a service agreement.

Further to this, Agile states that the people who are part of the team should share a common objective, regardless of whether or not the service team is mixed (i.e. made up of the organization's and customer personnel). A common problem in the relationship between customer and supplier is that the organization has an objective that is not compatible with the customer's objective. In this case, progress becomes impossible, as the organization and customer focus on two separate objectives. In this context, identifying a common objective that benefits both parties will enable the customer and the organization to work in the same direction on a day-to-day basis, as the benefit of achieving the objective will be shared.

Priority should therefore be given to writing and negotiating a contract or service agreement that facilitates and promotes the flexibility required to foster an Agile approach to service delivery and improvement. If this is not the case, the agreement itself can potentially inhibit the Agile approach and even render it impossible. This has an impact on several other areas where contracts and agreements are used:

— planning the SMS (see ISO/IEC 20000-1:2018, 6.3), when contractual requirements need to be considered or suppliers are used;

— the design of new and changed services (see ISO/IEC 20000-1:2018, 8.5.2.2), when agreements that support the services need to be considered;

— the development of SLAs with customers (see ISO/IEC 20000-1:2018, 8.3.3) and suppliers (see ISO/IEC 20000-1:2018, 8.3.4), when such agreements are involved.

In summary, the organization should establish common service objectives between the parties in the agreement that promote close collaboration and teamwork between the organization and the customer. This should provide greater flexibility for rethinking and adapting the details of the service when necessary, in order to meet the common service objectives.

5.6 How to achieve the service objectives

As previously highlighted, a clear service objective should be stated early on. However, it is not possible to determine how such an objective will be achieved from the outset, as it is not possible to know which methods of attempting to achieve the customer's objective will or will not work. This leads to the concept of "hypothesis," which is widely used in Agile.

In Agile, due to the high degree of uncertainty, anything formulated before being brought into a real environment is considered a hypothesis, i.e. a theory that something can become valuable, but where this value cannot be proven until it is verified in a real environment. Therefore, importance is not placed on formulating a perfect hypothesis, but rather on refining a hypothesis based on feedback from the real environment.

This concept is associated with the management of risks and opportunities (see ISO/IEC 20000-1:2018, 6.1) and the establishment of service management objectives (see ISO/IEC 20000-1:2018, 6.2). Processes associated with these elements should be used to determine hypotheses, monitor their potential value and manage them by determining whether the hypothesis is valid or not.

5.7 Agile learning

Agile learning is the concept used in Agile to reduce the risk derived from high levels of uncertainty. Only when a service improvement has been tested will it be possible to know whether or not it provides the benefits originally anticipated.

In Agile, teams work with short planning-development-delivery iterations, intended to observe what happens in a real environment and to learn as quickly as possible from that observation (known as working with "feedback loops"). Frameworks such as Scrum, for example, speak of iterations of a maximum duration

of one month (called sprints). In other frameworks, it is established that a minimum viable product (MVP) should be a limited product version that includes the main hypotheses, demonstrating whether the main product idea makes sense to the customer. In this way, the MVP is placed into the hands of customers (the early adopters) with the aim of receiving early feedback from them and testing, as soon as possible, whether the customer sees value in the main product hypotheses.

The same concept should be applied to services. Working with complex systems, it cannot be foreseen whether or not the original service concept will truly be of value to the customer. In service design, build and transition (see ISO/IEC 20000-1:2018, 8.5), to reduce risk, the organization should develop a minimum viable service (MVS) that contains the most important service elements and then bring them to a real controlled environment of early adopters as soon as possible. The objective is not to have a perfect service at that moment, but to validate whether the service is valuable and useful for the intended customer, or whether, on the contrary, it is necessary to formulate new service requirements and adapt the service proposal in the next planning-development-delivery rapid iteration. In this way, the MVS is built in short iterations and adapted based on what is observed in a real environment. The eventual service should be developed in this iterative and incremental way so that feedback from the customer can continually be incorporated into it. Only when the MVS has been shown to be truly useful can the service be considered "valid" and deployed on a larger scale.

Using this MVS approach has an impact on release and deployment management (see ISO/IEC 20000-1:2018, 8.5.3). An initial release is an MVS, which is then iteratively and incrementally built upon until the full service has been developed. The acceptance criteria of the release are the hypotheses that the organization has been working with to create value for the customer. An iteration in Agile always results in a release, as it will always result in new functionality being made available to the end users. Therefore, a "release" in Agile environments conforms to the definition of this term given in ISO/IEC 20000-1:2018, 3.2.13.

Using this Agile service development perspective, the risk of delivering a service that is not useful for the customer is reduced, and the risks of incurring high costs of of having customer expectations that will not yield the expected benefits are greatly minimized.

For the usual plan-design-transition-operate-improve service lifecycle, this means that there will be an initial release of the service that has gone through the first three phases (plan-design-transition) and the resulting MVS will then go into operation and become available for the customer. Subsequently, this cycle is repeated for additional service elements that were not part of the initial service release until the full agreed service has been developed and moved into the operation phase. After this, all continual improvement follows the same cycle.

This also applies to the service management processes: each process is first developed as the bare minimum viable process that is able to support the MVS. In every subsequent iteration of service development, processes can be further developed to manage increased complexity of the service or to add the new processes necessary.

When it comes to hypotheses, possible contractual barriers should be removed and organizational issues that prevent the organization from validating whether such hypotheses hold true should be resolved. A change of mindset in organizations is crucial: a “perfect” or “complete” service should no longer be created from the outset; instead, the service proposal should be adapted based on the observation of the service performance in a production environment.

5.8 The Agile team in a service context

5.8.1 Customer collaboration

In Agile, the team that supports the service is responsible for the service itself. The team organizes itself by making the necessary decisions to provide the best possible results. The team is composed of the very people who are in daily contact with end-users of the service, who best know their needs and are best placed to decide how to offer a better service. This is the concept of "self-organizing teams".

Micro-managing the work of people who are part of a service is inefficient and demoralizing for many reasons. First, the message that is sent to those who are part of the service delivery is that their ability and competence to make decisions for themselves is not trusted. And yet, it has already been established that

those who are closer to users have a greater quantity and quality of unfiltered information coming from the end-users themselves, so they can develop better options for achieving valuable objectives. This is known as "bottom-up intelligence".

The idea is simple: if the customer's objectives are known and the team is the one who maintains contact with service users, this team will be the best prepared to improve the service offered to users and keep the customer satisfied. Organizations that provide this flexibility to people are proven to perform better.

In an Agile service team, customer participation within the service is essential. In Agile, the customer is not a disconnected entity known only to be part of a service agreement. Rather, the customer is integrated with, participates in, and collaborates with the service team because the customer and the organization share the same goal.

Customer collaboration is essential and can be achieved in two ways: customers can be part of the service team or an interested party (see ISO/IEC 20000-1:2018, 4.2) of the service. In both cases, the organization should maintain frequent, direct, constructive and honest communication to foster the collaboration between both parties and the achievement of objectives. This, again, makes the practice of business relationship management (see ISO/IEC 20000-1:2018, 8.3.2) very important.

The uncertainty surrounding services forces the team to provide innovative solutions. Innovation is the end-goal of creativity. In order for the team to be creative, the organization should ensure an environment of security and trust in which experimentation and decision-making are encouraged and errors are not penalized. This is crucial for the service team to take responsibility for delivering the best results.

5.8.2 Servant leadership

The traditional figure of the manager assigning tasks is replaced by a "servant leader", who encourages the team and ensures that the team can exercise its responsibility and give its best. A certain level of supervisory management is required to make sure that the output of the service teams is aligned with the service management objectives and with the overall vision and mission of the company. Management is also needed for "removal of impediments" to productive working and for people management tasks such as performance management and coaching. Still, the responsibility for the service will remain with the entire team, not with a particular individual.

Certain activities should be deployed by the servant leader that can provide input into the continual improvement of the service (see ISO/IEC 20000-1:2018, 10.2), for example, daily stand-up meetings and retrospective meetings.

Daily stand-up meetings are short meetings (maximum 15 min duration) with the objective of sharing high-level information relevant to the team's efforts in achieving its objectives. The servant leader should work with the team to remove impediments to achieving the objectives.

Retrospectives should be used to evaluate how the work went in a service development iteration that has just concluded or during other key moments of the service lifecycle. These are longer meetings evaluating all aspects of the service lifecycle, determining improvements needed in processes, organization, tools and other aspects of the service.

5.8.3 Continuous improvement

The service team will have a shorter cycle to plan-develop-deliver the service and within that short cycle of feedback and learning, continuous improvement is achieved in a natural, organic and inherent manner (similar to that described in ISO/IEC 20000-1:2018, 10.2, "Continual improvement"). For example, using retrospective sessions at the end of each iteration. It can also be interesting to hold a review session in which the service team itself and the interested parties discuss the progress and next steps to be taken in the development of the service, inspecting and adapting its evolution and progress towards achieving the objective.

The implementation of continual improvement opportunities should also be subject to Agile practices themselves: just like any service element, improvements should be part of iterative and incremental development, either as an element of a single iteration or spread across multiple iterations.

5.8.4 Competence

An SMS should have a focus on the individual competence of the people performing their jobs within the context of the SMS. Skills should be assessed, both in the context of job interviews to find the right candidate and to create a close team of individuals who not only have their own specializations, but also have a broader development which gives them the flexibility to deploy their activities in other areas. This permits them to interact more effectively across functions with other team members. Agile is strongly in favour of having staff available that can perform multiple roles if required. The focus on evaluating competences in Agile is completely aligned with requirements for competence management in ISO/IEC 20000-1:2018, 7.2, where the organization should determine the competence needed for the job and ensure this competence is developed.

Closely working, collaborative teams need people with a cooperative and collaborative attitude, including openness to other people's ideas and perspectives, an active interest in trying to find new ways to achieve greater value, a drive for innovation and a continual focus on improvement in general. Other aspects of attitude highlighted by Agile include a high level of trust in each other; being able to take responsibility for one's results and a high degree of adaptability to cope with change in the environment, objectives or any other aspect. Similarly, although the ideal is always for team members to have as many skills and competences as possible to reduce internal bottlenecks, there can be certain levels of specialization required to support the service.

5.8.5 Agile culture

Organizationally, a culture should be developed that permits teams dealing with service management to be empowered to handle issues themselves as much as possible. This firstly requires a management culture that is happy to delegate and at the same time provide support to their teams where needed. This is contrary to the top-down management structure found in more traditional companies. It relates to the management support required according to ISO/IEC 20000-1:2018, Clause 5, while supporting the Agile idea of self-managing teams, where the decision-making power is delegated as far as possible to the teams who are supporting the SMS and the services. This gives a different interpretation to the requirements in ISO/IEC 20000-1:2018, 5.3 when compared to working in other environments. In an Agile environment, a significant number of responsibilities and authorities should be delegated down to the team-level, including, for example, the authority to approve changes and decisions on tool-selection.

5.9 Metrics

While service management objectives (see ISO/IEC 20000-1:2018, 6.2) provide guidance for deciding what to do and what not to do, metrics (see ISO/IEC 20000-1:2018, 9.1) help teams to know whether they are on the right track. In Agile, metrics focused on showing whether value is being delivered are especially important.

"Vanity metrics" are those which provide information that can be useful in some contexts, but do not tell teams if they are really delivering value. Both customer-focused metrics and more internally-focused metrics are concepts to consider in service management, provided that these emphasize the creation of value.

EXAMPLE A service desk with the strategic objective of guaranteeing that customers receive high-quality "boutique" support can have a metric that indicates that the customer receives a response in less than 24 hours or that an incident is closed in less than 48 hours. However, this metric does not provide information on whether the customer really receives quality support. Sometimes, for example, automatic emails perceived as "annoying" are sent to the customer in order to comply with the KPI requiring that a response be provided in less than 24 hours.

A balance should be found between adhering strictly to service level targets and finding the most valuable solution that satisfies the customer's needs and expectations. Specifically, a mechanism for measuring customer satisfaction (see ISO/IEC 20000-1:2018, 8.3.2) should be used in order to get a clear metric to determine progress in relation to the service objective, which, in the case of the Example, is to ensure that the customer receives high-quality support. This metric is focused on what the customer experiences, due to the actions that the service team takes, and provides the freedom to make decisions and select actions that will help improve the service. The metric guides actions and encourages innovation to find those actions that offer the best results. Furthermore, the right actions contribute toward activities that deliver the services to the customer, focusing on understanding the underlying problems that need to be solved for the customer in order to achieve the best outcome possible.

ISO/IEC 20000-1:2018, 9.1 contains requirements for monitoring, measurement, analysis and evaluation. In that context, the organization should determine what needs to be monitored, how and when these elements are to be monitored, and when the results of the monitoring are to be evaluated. In an Agile context, the organization should primarily focus on those metrics that determine whether value is being provided to the customers. In addition, more internal metrics can be monitored that determine efficiency of the workflow, work-in-progress limits, and other productivity metrics.

It is very important to use the appropriate metrics depending on the context. When looking to verify the quality of a service, it is much more appropriate to use metrics that focus on the perspective of the value received by the customer, rather than, for example, those that focus on measuring the actions carried out.

Reporting on these metrics is covered in ISO/IEC 20000-1:2018, 9.4. The organization should determine which reports on the performance and effectiveness of the SMS and the services are to be created. Most importantly, decisions should be based on these reports and actions should be taken as a result.

In Agile, relevant and appropriate metrics should be shared directly with those teams that can potentially benefit from such information in their decision making focussed on the generation of value.

One way to approach metrics in an Agile service is to consider them as hypotheses: fulfilling a metric brings the team closer to achieving service objectives, but it cannot be said that this is a certainty. This way of understanding metrics promotes their use as a form of learning, constantly pushing the team to rethink whether the metrics used provide valuable information, and to stop seeing them as absolute truths or as appropriate criteria for measuring people's performance and productivity.

5.10 Change management

5.10.1 Change and the SMS

Change applies to an SMS and services. Change expresses itself in changing service requirements, changing services, a changing competitive landscape, changing tools and technology and many more other variations of change.

Agile encourages organizations to embrace change in order to be adaptive to changing customer needs. It is this type of change that occurs in non-software environments as well: between contract-signature and implementation of a service, the customer can have changed their mind and decided they want something quite different from what was originally agreed. Embracing change should also cover other types of change, such as market change, technological change, etc. Services should be developed taking into consideration a continual need for change. This is already covered in ISO/IEC 20000-1:2018, 8.1 where it is stated that the organization should control planned changes to the SMS and review the consequences of unintended changes.

Coping with continual change can be achieved by making services as flexible as possible from the start. The use of new technology permits increased flexibility compared to in the past: services such as software-defined networking and cloud computing cater for adaptation to changing customer requirements. The SMS itself needs to be made flexible as well. This can be achieved by favouring simplicity: simple, intuitive processes, metrics (see 5.9) and reporting (see the requirements for service reporting in ISO/IEC 20000-1:2018, 9.4), that are easy to use, straightforward to produce, well-understood by customers and that can be equally effortlessly changed when needed.

5.10.2 Operational change management and release and deployment management

Customers want their change requests to be implemented successfully, without disruption to the live environment. To help ensure a change is implemented successfully, a number of aspects should be taken care of.

a) Thoroughly developed and tested service enhancements — this is part of the release and deployment and service design and transition processes in ISO/IEC 20000-1:2018, 8.5.2 and 8.5.3.

b) Iterative and incremental development — this should also be used in change management where each request for change should become part of a service development iteration, either as a whole or in incremental parts if that is better for service stability.

c) A well-communicated release schedule — when working in an Agile way, iterative service enhancements should be accompanied by a release schedule that is visible to all stakeholders. ISO/IEC 20000-1:2018, 8.5.3 contains requirements related to the planning of releases that can facilitate this.

d) An efficient, simple and flexible change management process — one that does not act as a roadblock for changes but does perform the necessary checks to ensure changes can be implemented in the next release. This can be achieved within the scope of the requirements in ISO/IEC 20000-1:2018, 8.5.1, by developing a change management policy and process aligned with Agile practices.

e) Thorough testing of implemented changes before the release is implemented and business verification after deployment — if these are not successful (or nobody is available to do business verification), the changes need to be rolled back. ISO/IEC 20000-1:2018, 8.5.3 requires releases to be verified for success or failure and causes of failures to be analysed so future releases can be faultlessly deployed.

In this way, the change management and release and deployment processes can be turned into a flexible set of processes that help the company achieve a more dynamic way of implementing and improving services, thus increasing the value they provide.

5.11 Simplicity and efficiency

With the objective of maximizing the value that the client perceives from the service, the service team should always seek to be as efficient as possible. To this end, they should be self-critical and question whether the work they are doing can be improved, and can be done in a simpler, more direct and clearer way.

The role of the manager (or servant leader) and the organization itself should be to help the team in whatever way is necessary to make them more efficient, for example by resolving impediments, eliminating barriers, providing knowledge of solutions that have worked in other teams, and encouraging communication between teams. The retrospective thus becomes a particularly important session for addressing these points. Unfortunately, there is no way to completely standardize efficiency in teams, as in highly uncertain environments what works for one team and client will not necessarily be efficient for another team and client. Even so, common elements can always be found across teams and customers, so standardizing these common elements can help teams to be more efficient.

It is important to remember that, being in a highly uncertain environment, what is standardized today because it is useful can potentially be obsolete tomorrow and no longer useful. The organization should be as efficient as possible in order to deliver maximum value. That is the only thing to be considered as a certainty in this context.

As stated in the Introduction to ISO/IEC 20000-1:2018, the main aim of establishing an SMS is to provide ongoing visibility, control of services and continual improvement, leading to greater effectiveness and efficiency. Using continual improvement, daily stand-ups, retrospectives and standardization is a way to achieve this effectiveness and efficiency.

5.12 The role of documented information

Agile tries to reduce overhead that has little to do with the final product and does not add value, such as extensive project plans, detailed requirements documents that risk becoming obsolete as soon as written, and other non-core documentation. The focus is on creating a working service, as that is what provides value to the customer.

ISO/IEC 20000-1 requires the organization to document a number of policies, processes and provide evidence of conformance in the form of documented information (see ISO/IEC 20000-1:2018, 7.5.4). This potentially seems contradictory to the aims of Agile in terms of reducing documentation overhead. However, the primary aim of an SMS is to provide a framework for working more effectively and for efficiently providing services by requiring a number of service management elements (including documented information) to be put into place. A balance should therefore be found between the documentation requirements of ISO/IEC 20000-1

and Agile principles by creating the minimum necessary set of documented information that is required to support the SMS and the services in a valuable way.

NOTE What ISO/IEC 20000-1 refers to as documented information does not always have to consist of a large number of separate documents. Some documents can be merged together. For example, a process and procedure can be written into one document, change and release management can be merged into one document, and procedures can be configured into service management tools.

5.13 Compatibility of Agile with service management

It can seem that Agile approaches and its principles are not entirely compatible with service management and the requirements of ISO/IEC 20000-1, as the latter has traditionally been driven by process, documentation and tools. However, the growing trend of providing "everything-as-a-service" means that, increasingly, the end customer places less importance on how a service will be delivered and more on what will be delivered and why, linking it to the direct value it brings and leaving the details of how to achieve it more open.

This does not mean that it is not important to rely on processes and systems to perform the delivery of services with a high level of quality, security and efficiency. However, an Agile approach can coexist perfectly in the field of service management, continually and sustainably promoting that the service delivered provides maximum value to the customer and that the service design, SLAs, KPIs and associated resources are focused on delivering such value.

Agile provides a refreshing perspective where value creation and the customer's perspective are central. ISO/IEC 20000-1:2018, 5.1 and 8.2.2, emphasize the importance of determining value and customer requirements.

Agile's focus on value delivery to the customers naturally leads to a closer collaboration with customers in the services area. The aim to involve the customer more closely in the development and delivery of services is an effective way to ensure that customer satisfaction is maximized.

Agile emphasizes that the well-being of the people who actually perform the work is fully in agreement with the requirements concerning management responsibility and support for the SMS in ISO/IEC 20000-1:2018, Clauses 5 and 7.

Agile seeks to reduce unnecessary documentation, as some documents are potentially never read by anyone or become obsolete as soon as they have been written due to new developments and requirements. The key is that the need for any documentation should be determined by its value, as defined by stakeholders or requirements of the SMS. Creating documentation that does not add value should be avoided.

NOTE A list of mandatory documented information for an SMS is provided in ISO/IEC 20000-2:2019, Annex A.

Agile embraces change as a constant, but raises a valid point stating that the service provider should be flexible about change. This applies to services as well as to software development.

Iterative and incremental service provisioning is an area of which Agile is the great proponent, but it is also the area that is most difficult to apply to services. The question of whether a minimum viable service, on top of which incremental enhancements can be regularly provided, can actually be developed, depends on the type of service an organization is providing. It also depends on the customer agreeing with this approach, where the benefit is that services can be made available earlier, but subsequent enhancements are developed in a collaborative way. The agreement with the customer needs to reflect and facilitate the flexibility required in an Agile approach. This is often a departure from classic service agreements.

6 DevOps within an SMS based on ISO/IEC 20000-1

6.1 General

In the following subclauses, a number of DevOps concepts are discussed and applied to the requirements of ISO/IEC 20000-1.

ISO/IEC 20000-1 promotes the development of an effective, efficient, flexible, coordinated and integrated system for the management of services and their continual improvement.

DevOps attempts to bridge the gap between development and operations teams by unifying them for better, faster outcomes, focusing on providing business value through agility, quality and stability.

6.2 DevOps principles

DevOps can be defined as a set of principles and practices which enable better communication and collaboration between relevant stakeholders for the purpose of specifying, developing and operating software and systems services with continuous improvement (called continual improvement in ISO/IEC 20000-1:2018, 10.2) in all aspects of the service lifecycle.

ISO/IEC/IEEE 32675:2022 provides the following introduction to DevOps:

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

DevOps encompasses several principles, as follows.

a) Customer-centric action: customers, and their requirements, needs and expectations, are expected to always be central to organizations which want to be successful. Organizations will know their customers well and involve them in activities from concept to final delivery of a service.

b) Create with the end in mind: having a clear vision of the outcomes is necessary for moving in the right direction. Developing successful services that meet customers’ needs can only be achieved when there is empathy for the customer and an accurate understanding of their requirements.

c) End-to-end responsibility: customers are not only concerned with the services but also about the whole experience of doing business with an organization. End-to-end responsibility for activities from concept to delivery of a service is essential for creating a positive customer experience. It requires organizations to put themselves in the shoes of their customers to understand each requirement and assume responsibility for ensuring that every aspect of doing business creates a consistently positive experience.

d) Cross-functional autonomous teams: cross-functional autonomous teams consist of people from different business units. They work in a collaborative way to achieve the stated objective. They are self-organized and own the end-to-end responsibility of determining solutions to problems, or opportunities, within the business constraints.

e) Continuous improvement: this is an on-going improvement process to make breakthrough enhancements in products, services and processes that generate the most value for customers. Organizations can achieve this by identifying opportunities for streamlining work, minimizing waste, and optimizing speed, costs and ease of delivery.

f) Automate everything you can: automation enables people to be more productive and innovative by freeing them from repetitive manual tasks. Automating repetitive manual tasks helps reduce the number of errors (improves accuracy), improves speed to market (reduces delays), and maximizes productive and innovative activities.

g) Shift-left and continuous everything: the continuous delivery, testing and quality assurance practices are shifted left (earlier in the workflow) to be planned and executed at the same time as design and development. The DevOps concept of "continuous everything" means using the same practices in development as in operations and improvement.

DevOps principles are particularly applicable in an environment where there is a need for rapid delivery of services. These services are usually automatically and incrementally built, tested and released into the production environment.

6.3 Customer-centric action

DevOps encourages an open organizational culture to become customer-centric. This focus on customers can be related to business relationship management and service level management (see ISO/IEC 20000-1:2018, 8.3.2 and 8.3.3).

In a customer-centric and open culture, people are open to feedback and have the authority to take calculated risks to achieve the desired goal. They are focused on creativity to produce valuable ideas to enhance the customer experience.

Short feedback loops with the customers help organizations to focus on what the customers need. As an example, some elements of online platforms can be seen as customer-centric in that they alert users to items which are potentially of interest to them.

6.4 Create with the end in mind

Creating with the end in mind takes a holistic view of both the creation and use of the service. If an organization does not know where it is going, it will not know whether it has lost its way. Creating with the end in mind focuses on the outcome, which is the value created for the customer and the organization by the service. It requires mutual trust among different teams and team members to establish this focus and work towards an agreed outcome.

Organizations should begin with a complete understanding of a customer's requirements, leading to the creation of a result-oriented plan. Such an approach requires organizations to appreciate service-thinking and determine the issue the customer wants to address. Reference is made to service requirements in ISO/IEC 20000-1:2018, 4.2, Clause 5, 8.2.2, 8.3.3, 8.4.3 and others.

When creating with the end in mind, the organization should establish a safe environment where staff can share the required knowledge (see ISO/IEC 20000-1:2018, 7.6), identify and address risks (see ISO/IEC 20000-1:2018, 6.1), and be more creative and innovative.

It is essential to receive feedback early in the service development phase to allow changes to services to be made quickly with minimum impact (see ISO/IEC 20000-1:2018, 8.5.1). This feedback can come from customers evaluating early service implementations or from (automated) testing of the service.

6.5 End-to-end responsibility

DevOps encourages organizations to practice the concept of cradle-to-grave to enable teams to accept end-to-end responsibility for service delivery.

Accepting the end-to-end responsibility, authorities and responsibilities for the SMS and the services (see ISO/IEC 20000-1:2018, 6.3d) as well as the authorities and responsibilities for design, build and transition activities (see ISO/IEC 20000-1:2018, 8.5.2.1a) is potentially the most crucial ingredient for DevOps. When people care about what they do, and have the required skills, knowledge and resources, they can collaborate to meet their responsibilities. If they care, they will learn, adapt, improve and provide great services and value.

In traditional organizations, solutions are created in development teams and handed over to operations teams to deploy and maintain. This can be seen as a horizontal approach. In a DevOps environment, teams should be organized vertically to be responsible, from beginning to end, for the services they deliver. This should lead to better quality services where applicable.

The make-up of these vertical teams needs to be as stable as possible to help develop and embed effective working practices to support the achievement of their goals.

Teams are organized vertically to deliver as much customer value as possible, rather than just producing outputs. They own each stage of the product development process to deliver independent, valuable products

and are responsible for taking a decision, gathering information, and assessing alternative resolutions. They have the authority to take calculated risks and make effective decisions.

Having end-to-end responsibility encourages people to collaborate, learn, adapt and improve, in order to deliver the customer experience. It helps them to acquire new knowledge and enhance their existing knowledge, skills, behaviours, values, and preferences.

The teams are responsible for providing end-to-end support for products, which helps to enhance overall performance.

6.6 Cross-functional autonomous teams

DevOps encourages organizations to create cross-functional autonomous teams which share the same goals and successes, and which are willing to learn from failure.

When people from different functional units, each with a variety of skills, work together with the same goal, they can solve even complex problems. Such teams should be independent and have the freedom to control their activities and make effective decisions when required.

People in these teams work for a purpose, within the boundaries they have set themselves, in order to achieve the service objectives. They have deep knowledge and skills in a specific area of specialization and are capable of applying their specific and broad knowledge across disciplines.

When people with different skillsets work together for a purpose, they should share their different viewpoints with the team, which promotes a greater overall understanding of any given issue.

6.7 Continuous improvement

Continuous improvement (called continual improvement in ISO/IEC 20000-1:2018, 10.2) is a concept borrowed from Lean development that focuses on adaptability and learning through structured problem-solving. Service teams should focus on experimenting with new ideas, minimizing waste, optimizing for speed, cost and ease of delivery. Teams should be encouraged to innovate continuously and help their organizations beat the competition. It is important to experiment and learn from failures, hence the term "fail fast".

Problem areas should be identified in a systematic process and the service team should work towards finding the correct solution or identifying opportunities to improve.

There is no improvement if the success of processes, people and tools cannot be measured. An effective performance measuring system should therefore be established. This contributes to identifying the improvement opportunities in the existing processes, tools and services.

6.8 Automate everything

DevOps encourages organizations to move away from manual activities and to automate as much as possible to maximize customer experience and minimize repetitive tasks.

Regardless of the technology platform or development practices, every organization uses processes to develop and deliver new or changed services (see ISO/IEC 20000-1:2018, 8.5.2). This process can be manual or automated. As far as possible, organizations should move from manual to automated development, which improves efficiency and consistency within and across processes. Automation should be fully validated and should take into consideration the cost of automation itself and its frequency of use.

Automation helps to enhance quality and maximize the flow of work. It can also help to increase the speed of delivery, reduce the cost, and enhance the quality of services. The following list provides examples of areas which can benefit or already benefit from automation.

— Continuous improvement (including waste reduction) is greatly helped by automation. There have been various innovations and trends that have helped to achieve service process automation, including accelerated processing of service interactions and reduction in the error rate through automatic execution

of predefined processes. This leads to incidents being easier to trace for errors caused by automation, less systemic burden due to fast and stringent implementation of queries, and reduced workload for first and second level support, which helps improve the quality of the services.

— Continuous integration/continuous delivery (CI/CD) focuses on bringing software into production through a fully automated process, multiple times per day without issues (see ISO/IEC 20000-1:2018, 8.5.3). Continuous integration helps in this respect by combining and testing new software elements with the main code automatically. Continuous delivery brings all software elements together and prepares the resulting complete programme for release into the production environment, including possible roll-back if it fails.

— Cloud-based platforms are replacing traditional data centres, making changes to processor or storage capacity faster and easier (see ISO/IEC 20000-1:2018, 8.4.3).

— A container-based infrastructure helps in virtually packaging and isolating applications from the operating system, facilitating release and deployment management (see ISO/IEC 20000-1:2018, 8.5.3).

Automation increases the productivity of people by automating repetitive manual tasks. This allows staff to use their time for more productive, useful and innovative activities. Manual tasks are prone to human error. Automation increases the overall quality by eliminating human errors at a faster rate and lower cost. Automation helps reduce errors as it produces results based on a defined set of rules and increases performance due to the efficiency of the platform.

Automation also helps to make systems scalable, which is essential for meeting the changing requirements of customers. Automation helps to makes systems more flexible by enabling them to perform continuously. This, in turn, increases the turnaround time.

6.9 Shift-left and continuous everything

6.9.1 Shift-left

Shift-left is particularly valuable for improving the user experience of services, increasing reliability and reducing subsequent operation and maintenance costs of services.

NOTE ISO/IEC/IEEE 32675 uses the term "left-shift" instead of "shift-left". In the industry, the expression "shift-left" is more common. This term is therefore used in this document.

Shift-left aims at moving as many activities to an earlier stage (left) of the service lifecycle as possible, so that possible issues can be captured and remediated as early as possible, such as during service design and transition (see ISO/IEC 20000-1:2018, 8.5.2).

The following is an example of how to implement the DevOps principle of shift-left.

EXAMPLE To improve cloud service quality, an organization takes a shift-left approach to the quality, operation and security of that service. To accomplish this, the requirements are identified (see ISO/IEC 20000-1:2018, 8.3.2), based on an assessment of the customer's needs, to direct the design and development of the cloud service. It is then confirmed with the customer that the developed solution meets the actual needs, before releasing it into operation (see ISO/IEC 20000-1:2018, 8.5.3). Environments similar to the production environment are set up for development and test environments, creating a pipeline for deployment to production. In this way, customer requirements are validated as early as possible in the service lifecycle and implemented as part of the live service accordingly.

6.9.2 Continuous everything

The concept of continuous everything implies increasing agility throughout the service lifecycle from testing, deployment and monitoring through to integration and delivery. The entire process from gathering requirements to deployment of the service in production should be continuously controlled. This is done using several "continuous" practices, including:

— continuous planning (see ISO/IEC 20000-1:2018, 8.2.2), where the output from a previous phase is used as input to the next iteration;

— continuous documentation (see ISO/IEC 20000-1:2018, 7.5), where documentation is updated continually by the automated system (e.g. change and incident records), not necessarily by humans;

— continuous testing (see ISO/IEC 20000-1:2018, 8.5.3), where tests are run automatically and continuously to verify releases are ready for deployment and function as required in production;

— continuous integration (see ISO/IEC 20000-1:2018, 8.5.3), where new or changed service components are continually added to a release until it can be deployed in the production environment;

— continuous deployment (see ISO/IEC 20000-1:2018, 8.5.3), where releases are automatically deployed, tested and rolled back if they fail to provide the expected functionality;

— continuous monitoring (see ISO/IEC 20000-1:2018, 8.6.1), where the production environment is continuously monitored for incidents;

— continuous learning (see ISO/IEC 20000-1:2018, 10.2), where the development and operations teams continually learn from the previous development iteration and deployment of new and changed services;

— continuous measurement (see ISO/IEC 20000-1:2018, 9.1), where the functionality of the services is continually measured and tested to verify the services function as required.

As an example, a service team's objective derives from the business goals and priorities. The top priority outcome for the business is time-to-market: they hope to roll out a new product from idea to end customer in a faster timescale.

To meet this requirement from the business, the organization should define different types of demand and flow items based on the context of the organization (see ISO/IEC 20000-1:2018, Clause 4), for example:

— feature: the new business value that is requested by the customer;

— maintaining sustainability: security, availability and risk, which are owned by the service provider;

— continuous improvement: improvements in managing the technical backlog and technical debt by the service owner, which can accelerate development and go-live times to shorten the lead time for deployment;

— experiment: the production of innovative ideas, proposed by all team members.

The priority of these elements can be adjusted, based on the service requirements and the business environment. The business goal is the most important. It should be a balanced choice.

As one of the key DevOps principles is to accelerate the reception of feedback, the service team should collect customer feedback much faster by shortening delivery lead time. This demonstrates the value of DevOps to the business. In the meantime, trust is built between the business and the organization.

6.10 Compatibility of DevOps with service management

DevOps is driven by its seven principles: customer-centric action, create with the end in mind, end-to-end responsibility, cross-functional autonomous teams, continuous improvement, automate everything you can, shift-left and continuous everything. These principles are highly aligned with the requirements for an SMS based on ISO/IEC 20000-1.

Customers should always be central in an SMS, because they are the recipients of the services. The creation of value for the customer and for the service provider is a basic outcome of providing services. Value creation is therefore the actual "end" that should be kept in mind during the complete service lifecycle.

The way to achieve this comes primarily through the organization of the team providing the services: the organization should give the service team end-to-end responsibility by encouraging ownership with autonomy and allowing it to learn from its mistakes. Teams should consist of people with diverse backgrounds so that various perspectives can be used to develop the services.

This naturally leads to an emphasis on continuous improvement: while making mistakes or finding issues in the services, the teams should be allowed to fix these issues continually, thereby improving the value provided to the customer and the organization itself.

The way to do this for software-based services is to implement a very high degree of automation during the complete service lifecycle and developing continuous processes. Development, test, integration, deployment, acceptance, roll-back and delivery should all be automated if the nature of the service allows this. This results in the least amount of disruption in the production environment when making changes and deploying releases.

Shift-left eventually brings the focus back to the customer and efficiency by shortening resolution times for service requests and change requests. Service providers should implement a high level of customer self-service facilities for simple requests and should at the same time make internal processes as efficient as possible by moving activities that can be done early in the service lifecycle forwards.

Information security management (see ISO/IEC 20000-1:2018, 8.7.3) has become of crucial importance in the world. Organizations should incorporate information security controls into their services from the start to meet customer and other interested parties' requirements in this area. The DevSecOps approach is a variation of DevOps that incorporates the design, build and testing of security controls throughout the software development and service lifecycle, leveraging automation. In this way, information security becomes a shared responsibility from development to operations.

7 Applying Agile and DevOps within an SMS

7.1 Benefits of applying Agile in an SMS

Service management and Agile are sometimes seen as incompatible. Agile is viewed as being concerned with development, whereas service management is concerned with operations. Service management is sometimes viewed as process-oriented, defined and controlled, whereas Agile's focus upon people, collaboration, iteration and adaptation can make it appear incompatible with a defined and process-oriented SMS.

Most modern organizations and their SMS operate in environments of high uncertainty, at least in some contexts. Changing demands, new technologies, customer and end-user expectations and competitive landscapes are all elements that can rapidly change and introduce high uncertainty and risk into the effectiveness of an SMS and its ability to deliver the value required. Agile approaches have evolved as a powerful tool to reduce the risk associated with such circumstances and to most effectively achieve the desired results. Therefore, designing and developing an SMS with Agile principles in mind is most beneficial when the SMS is developed in a context of high uncertainty with the risk that is associated with it.

However, although it may seem that high-risk contexts are rare and mainly limited to technological areas of highly disruptive innovation, the reality is that these situations are increasingly common, especially when there is a difference between the expected value (from the customer, business or strategic objective) with respect to the value received. Such instances are clear symptoms that there exists high uncertainty in SMS design and implementation, regardless of whether the service is essentially technological or not.

Examples of common high-risk contexts are as follows:

— the customer is clear about the objective and the business value they want to achieve with the SMS, but the roadmap to ensure that the value is achieved is not clear or cannot be pre-defined;

— the SMS rigorously meets the contractually defined KPIs. However, the customer is not satisfied because the business outcomes and strategic value they expected to receive are not being realized;

— an SMS has been performing with very good business results to date. However, a change occurs in the market (e.g. a change in end-user habits or a new, younger type of customer who behaves differently from the usual type of customer) that causes the SMS business outcomes to decline or to become poor.

In these and many other contexts, the application of Agile approaches and Agile principles can be a key factor to help reduce the contextual risk and to maximize the probability of success of the SMS.

Risk could be reduced in the above situations by applying the Agile approaches as shown in Table 1.

Table 1 — Risk reduction in Agile Approaches

Application of Agile approaches	Explanation/Outcome
Fostering a customer-supplier relationship based on shared objectives.	When both customer and service provider share the same objectives, the relationship is naturally more balanced, making it easier and more intuitive to make successful decisions because, ultimately, both parties are pursuing the same outcomes.
Encouraging collaboration between customer and service provider to achieve shared objectives.	In situations where the solution is not clear, effective collaboration between all parties becomes critical. Likewise, when collaboration is not a reality on a day-to-day basis, success is jeopardized. Therefore, it is emphasized that collaboration is not optional but rather an obligation, even influencing how the contract between the parties is drafted (if the relationship is contractual).
Designing an SMS that facilitates frequent low-risk experiments and evaluation of results to help make better decisions.	In circumstances where it cannot be completely determined how to achieve success, the Agile approach based on observation, hypothesis formulation, quick experiment, analysis of results, and decision making, is recommended. This is a fundamental point of Agile approaches. With a clear focus, there is room to model the development of the SMS with frequent iterations, reviews, improvement retrospectives, daily monitoring and other Agile activities within the day-to-day process-led work of the SMS.

Other examples of Agile principles that promote these successful outcomes include the following:

— leadership that fosters an environment in which the best ideas are given weight above certain other criteria;

— encouraging a transparent and sincere relationship between all those involved in the SMS which helps foster a spirit of "self-scrutiny" to identify and implement better solutions rather than ignoring problems;

— fostering a scale of values that are manifested and tangible in the day-to-day service reality in a way that contributes to a constructive spirit of healthy collaboration and clear focus upon business objectives at all levels of the SMS.

Agility is about risk reduction, efficiency and effectiveness in achieving valuable business objectives above all else. The principles of Agility are useful in multiple contexts of an SMS, especially in some of the major challenges that organizations face today.

Even though Agile principles have been developed for environments with high risk or high uncertainty, they can also be followed in environments that do not have such high risk or uncertainty. The benefits of using Agile methodologies still apply, including a better understanding between the customer and service provider, quicker delivery of working services and greater reliability of the services.

A further example of the use of Agile principles is an existing SMS that transitions to more sustainable operations or a new SMS with a focus on sustainability. This can be considered as a complex system, as the objective is clear but realizing that objective and its impact on the SMS is complex. In this sense, some of the benefits that Agile principles facilitate are as follows:

— all parties involved are clear about the desired outcomes;

— work is done and decisions are made with the sustainability objective in mind;

— experimentation is used to see what works and what doesn't work in pursuit of that objective before scaling unproven solutions;

— an environment is fostered in which new ways of approaching objectives emerge organically and continuously;

— decisions are made focusing on the value of reducing environmental impact over other criteria.

In other words, transitioning to a sustainable SMS is not simply a matter of carrying out initiatives that reduce infrastructure energy consumption. In this example, it is critical to success that an environment be established that affects the entire SMS (environmental, social and economic elements). This conditions the way work is done on a day-to-day basis and implies interactions and decision making that is truly focused on how to be more sustainable, whilst meeting the objectives of the SMS in the right way.

NOTE See ISO/IEC TS 20000-16 for further information.

7.2 Benefits of applying DevOps in an SMS

Like Agile, DevOps is an approach, not a methodology. There are many different definitions of what it is, but they all have one point in common: they represent a cultural and professional movement that stresses communication, collaboration and integration, aimed at establishing a particular culture and environment.

Organizations are sometimes under the impression that they need to produce a lot of documentation, processes, procedures and measurements in order to conform with the requirements of ISO/IEC 20000-1. In contrast, the wording of the Agile Manifesto, on which DevOps is also based, gives the impression of removing such constraints. However, as highlighted in 5.2.2, while the Agile Manifesto values certain elements over others, it does not simply ignore these secondary elements. For example, although the Agile Manifesto states that it values "individuals and interactions over processes and tools", if the effectiveness of a process is ignored, it affects individuals, no matter how much they are valued. Within the DevOps context, documentation is therefore necessary to provide guidance, but it does not need to be extremely long to be effective. Service providers should find a balance between what documentation is required, what can be configured into service management tools, and what can be omitted.

ISO/IEC 20000-1 includes requirements in terms of what organizations need to do, not how they do it. DevOps offers solutions to the question of how to meet such requirements. Both approaches should be combined to provide better service and value.

Many DevOps concepts and practices can be easily applied in an SMS. For example, the need for cross-team collaboration, effective communication with all interested parties, ownership of results, and effective feedback to allow corrections to be made (fail forward), all fit perfectly into the requirements of ISO/IEC 20000-1.

Applying ISO/IEC 20000-1 requirements in a DevOps organization requires special attention because:

— all teams should come together to define the strategy. It is the service owners and lead engineers from different teams who often work to define the strategy;

— teams should be responsible for the end-to-end design, implementation, and execution of the service. This includes acceptance of complete responsibility for the service within the team;

— resources are organized differently: resources should be organized as service teams, resulting in fewer handovers, less complex process descriptions, and minimal documentation;

— service teams work as autonomously as possible: teams should be free to do any tasks or activities required to deliver the product without alignment and approvals of other teams;

— people should be aware that rigid processes can obstruct the ability to adapt to change;

— people should be aware that they need to adapt to a continuously changing environment;

— processes should be automated as much as possible because they have low variability and are an excellent candidate for automation. The major focus should be on automating processes to make these faster, more reliable and repeatable.

Using DevOps in an SMS provides the following benefits:

— DevOps is designed to maximize business output through appropriate just-in-time business processes, such as increasing business speed, or minimizing operating costs;

— DevOps is more flexible, more efficient, and more secure than traditional methods to achieve the requirements of ISO 20000-1, and the integration of the two provides a good approach to help to drive the outcomes and benefits that customers want to achieve;

— DevOps promotes communication and cooperation;

— continuous integration/continuous delivery (CI/CD) can improve effective and efficient service deployment, which supports new or changed services to be delivered at a faster pace. Using CI/CD, the organization can get rapid customer feedback to optimize and improve the service. More frequent releases can bring more and faster value to customers; CI/CD reduces dependencies on staff and helps control delivery of services;

— various deployment modes assist in controlling risk and reducing the impact on services;

— reducing technical debt is a great help in improving availability and business continuity of a service and many requirements are realized at the front-end, which is beneficial for reducing the cost of service operation and improvement;

— visualization can be used for service management to provide more timely and accurate reports with transparent information;

— shift-left can help improve the identification of issues and the prevention of defects escaping to other environments;

— effective deployment of features using automation should be applied to streamline flow processes which helps increases efficiency.

There is no conflict between ISO/IEC 20000-1 and DevOps — they have different perspectives and emphases:

— ISO/IEC 20000-1 emphasizes service management and specifies requirements for establishing, implementing, maintaining and continually improving an SMS;

— DevOps emphasizes how to collaborate to deliver software to live environments rapidly but in a controlled way.

DevOps and ISO/IEC 20000-1 should be integrated to provide good services, to deliver value to the customer and help facilitate outcomes that customers want to achieve.

7.3 Patterns and anti-patterns of applying Agile in an SMS

Table 2 provides a list of patterns and anti-patterns related to Agile and its influence on an SMS.

NOTE "Patterns" in this context are defined as symptoms of proper use of the framework. "Anti-patterns" are risks or other symptoms of improper use of the framework.

Above all, these examples reinforce the idea that a service provider should have an SMS where Agile values and methodologies are real, profound and tangibly contribute to maximizing the value delivered by the SMS and the services, and to the achievement of strategic objectives.

Table 2 provides some common examples but many more exist. Promoting such patterns and identifying and rectifying anti-patterns is a fundamental part of Agile maturity.

Table 2 — Examples of patterns and anti-patterns of using Agile in an SMS

Category	Pattern	Anti-pattern
Objectives and goals	Define outcome-focused metrics that visualize how an organization is performing in achieving the SMS objectives.	Define superficial Agile metrics that say little or nothing about whether the SMS is making progress in achieving valuable objectives (vanity metrics).
	Metrics that visualize how an organization is progressing towards an objective are frequently reviewed and reconsidered as to whether they continue to make sense or not.	Metrics are defined at the outset, but the organization does not review them or consider whether they continue to make sense.
	The people involved in the SMS are clear about the mission of the SMS, what the SMS aims to achieve, how the SMS benefits the organization, and they contribute their full potential with the aim of achieving these goals.	Despite holding events recognized as Agile practices (e.g. daily stand-up, retrospective), people participating in the SMS simply execute tasks and activities without understanding why they are necessary within a broader context.
	An analysis is undertaken to check that an Agile approach to the SMS or a certain objective really makes sense.	Under the notion of "the SMS has to be Agile", an SMS is forced to be designed or modified to have an "Agile look and feel" without analysing whether the context is high risk or high uncertainty.
Relationships	Agreements between client and service provider are modelled with a focus on collaboration and common objectives, being flexible with respect to areas of uncertainty such as scope or KPIs to be met.	Agreements are modelled with a focus on concepts often associated with Agility (e.g. story points or features delivered per iteration) that do not allow for change in a flexible way for the SMS to meet its business objectives.
	The relationship between the service owner and the team is very close, with the manager acting as a true leader who knows how to get the best out of the team and encourages collaboration both inside-out (from the organization towards the customer) and outside-in (from the customer towards the organization), together with the promotion of team self-management and autonomy for the benefit of the SMS.	New roles are established for people in the team that sound Agile (Agile coach, Agile manager, etc.) but in reality there is a traditional organizational hierarchy where the Service Manager is the head of the team who centralizes all decisions and the team's relations with the outside world.
Organization	Planning is done in frequent iterations where focus is placed upon which objective will be the most valuable to achieve in that iteration.	Planning is done in frequent iterations where tasks are assigned by the service owner for the team to execute in that iteration.
	Daily monitoring (e.g. daily stand-up meetings) take place in which the team assesses the status progression towards achieving the objective of the iteration.	Daily monitoring (e.g. daily stand-up meetings) takes place in which the service owner makes decisions about the performance of the SMS team.
	SMS interested parties regularly inspect the outcomes of the SMS in a climate of honesty and transparency to jointly make decisions that will lead to improvements in the SMS and help to achieve its business and strategic objectives.	The service owner often presents the results of the SMS to the SMS stakeholders, in the spirit of self-preservation above all else.