International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.3525

(09/2020)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Cloud Computing

Cloud computing – Requirements for cloud service development and operation management

Recommendation ITU-T Y.3525

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	**Y.3500–Y.3599**
BIG DATA	Y.3600–Y.3799
QUANTUM KEY DISTRIBUTION NETWORKS	Y.3800–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.3525

Cloud computing – Requirements for cloud service development and operation management

Summary

Recommendation ITU-T Y.3525 describes the overview of cloud service development and operation management and its functional requirements. It provides the lifecycle of cloud service development and operation management based on five processes and eight stages. Additionally, this Recommendation also specifies the functional requirements of cloud service development and operation management derived from the corresponding use cases.

History

Edition	Recommendation	Approval	Study Group	Unique ID[*]
1.0	ITU-T Y.3525	2020-09-29	13	11.1002/1000/14403

Keywords

Cloud service, development and operation management, framework.

[*] To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Y.3525

Cloud computing – Requirements for cloud service development and operation management

1 Scope

This Recommendation provides functional requirements and typical use cases of cloud service development and operation management. This Recommendation covers the following aspects:

– overview of cloud service development and operation management;

– functional requirements of cloud service development and operation management;

– typical use cases of cloud service development and operation management.

2 Reference

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.1601]	Recommendation ITU-T X.1601 (2015), *Security framework for cloud computing.*
[ITU-T Y.3502]	Recommendation ITU-T Y.3502 (2014) | ISO/IEC 17789:2014, *Information technology –Cloud computing – Reference architecture.*
[ITU-T Y.3522]	Recommendation ITU-T Y.3522 (2016), *End-to-end cloud service lifecycle management requirements.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 cloud computing [b-ITU-T Y.3500]: Paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on demand.

NOTE – Examples of resources include servers, operating systems, networks, software, applications, and storage equipment.

3.1.2 cloud service [b-ITU-T Y.3500]: One or more capabilities offered via cloud computing invoked using a defined interface.

3.1.3 cloud service customer [b-ITU-T Y.3500]: Party which is in a business relationship for the purpose of using cloud services.

3.1.4 cloud service partner [b-ITU-T Y.3500]: Party which is engaged in support of, or auxiliary to, activities of either the cloud service provider or the cloud service customer, or both.

3.1.5 cloud service provider [b-ITU-T Y.3500]: Party which makes cloud services available.

3.1.6 service level agreement (SLA) [b-ITU-T Y.3500]: Documented agreement between the service provider and customer that identifies services and service targets.

NOTE 1 – A service level agreement can also be established between the service provider and a supplier, an internal group or a customer acting as a supplier.

NOTE 2 – A service level agreement can be included in a contract or another type of documented agreement.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 continuous integration: A software engineering approach of merging CSN:cloud service developers' working copies to a shared mainline at any time.

NOTE – In the continuous integration, each integration is verified by the automated building, including compiling, release, automated testing, and deployment, in order to as early as possible to find the integration errors.

3.2.2 continuous delivery: A software engineering approach in which the CSN:cloud service developers produce software in short cycles, ensuring that the software can be reliably released at any time.

NOTE – Continuous delivery does not mean that each change of the software needs to be deployed in the product environment as soon as possible, but means that each change is verified to be deployed at any time.

3.2.3 automated testing: A method in software testing that makes use of special software tools to control the execution of testing and then compares actual testing results with predicted or expected results.

NOTE – Automated testing is done automatically with little or no intervention from the CSN:cloud service developers.

3.2.4 continuous operation: The management processes associated with cloud service management to deliver the right set of services with the high quality and competitive costs for the cloud service customer (CSC) in a sustainable way.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CCRA	Cloud Computing Reference Architecture
CI	Continuous Integration
CD	Continuous Delivery
CSC	Cloud Service Customer
CSP	Cloud Service Provider
CSN	Cloud Service partner
E2E	End-to-End
NaaS	Network as a Service
NFV	Network Function Virtualization
SLA	Level Agreement
VM	Virtual Machine
VNF	Virtualized Network Function

5 Conventions

The keywords "**is required**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

In the body of this Recommendation and its annexes, the words shall, shall not, should, and may sometimes appear, in which case they are to be interpreted, respectively, as is required to, is prohibited from, is recommended, and can optionally. The appearance of such phrases or keywords in an appendix or in material explicitly marked as informative is to be interpreted as having no normative intent.

6 Overview of cloud service development and operation management

We are living in a volatile, uncertain, complex and ambiguous world where the environment is extremely sensitive. Along with the expansion of cloud computing technology and the demands of cloud service customers (CSCs), the complexity of the system architecture has greatly evolved and has never been more challenging. The traditional way of development and operation management is no longer effective and compatible with new technologies, e.g., container and micro-services. Therefore, the end-to-end (E2E) development and operation management of cloud services becomes one of the most important ways for success.

According to [ITU-T Y.3522], cloud service lifecycle management integrates and optimizes processes such as service provisioning, service assurance, service fulfilment, service charging and service change management for particular workloads respecting relevant governance and polices.
E2E cloud service development and operation optimize procedures, including development plan, automated tests, continuous integration (CI), continuous delivery (CD), monitoring state as well as quality assurance, and accelerate the delivery of high-quality cloud service to the CSC.

This approach has been reconstructing the way to effectively develop cloud service and safely try out hypothesis. First, the flow through the lifecycle, which is divided into plan, design, code, build, test, release, deployment and operation, should be identified and accelerated by automated test and CI/CD. Then, measurements are needed to be performed in each part of the flow, and the monitoring system is required to provide instant feedback. Quality assurance covers the whole lifecycle of development and operation, which is beneficial for identifying defects in the earlier stage and preventing problems from flowing downstream.

6.1 Cloud service development and operation management framework

Cloud service development and operation management framework refers to unifying the development, delivery and operation of cloud services. Based on the cooperation of the whole organization and on application architecture optimization, it is able to achieve rapid continuous delivery and high-quality cloud services, while ensuring stability. It can help the cloud service partner (CSN) and the cloud service provider (CSP) to improve development and operation efficiency, so as to flexibly respond to the rapidly changing CSC demand and market environment. The cloud service development and operation management framework and other related functional components of the cloud computing reference architecture (CCRA) are shown in Figure 6-1.

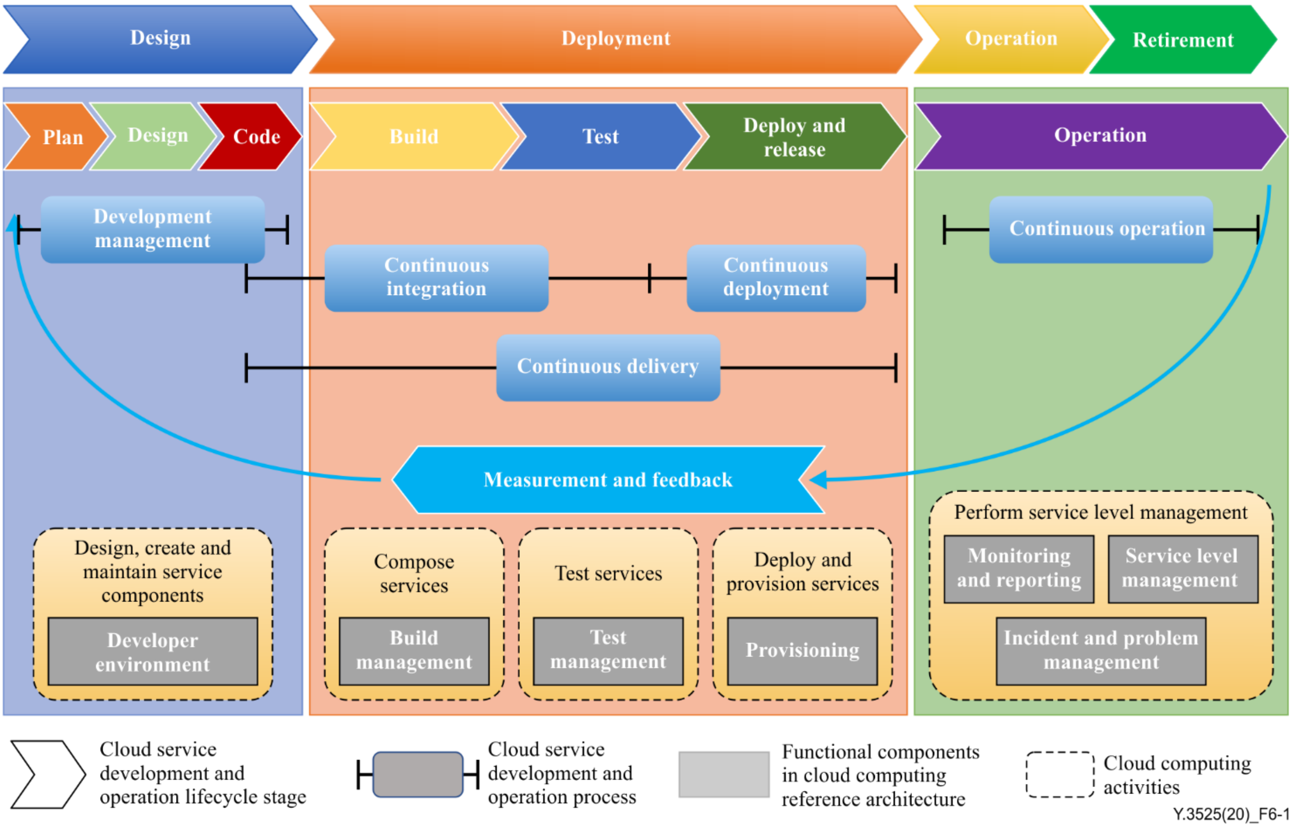


Figure 6-1 – Cloud service development and operation management framework

The cloud service development and operation management framework is divided into five processes as follows:

1) Development management

The development processes of CSN:cloud service developers, including priority of plan, architecture design, and coding, are managed according to the demands of CSC, and the CSN:cloud service developers need to adjust the content and the priority of the plan in time according to the feedback from CSC.

2) Continuous integration

CSN:cloud service developers integrate their work frequently, which is verified by an automated build process (including testing) to detect integration errors as early as possible to reduce integration problems.

3) Continuous deployment

CSN:cloud service developers deploy applications in a specific environment safely, quickly, automatically and sustainably.

4) Continuous delivery

The CSP continues to deliver the software which is built and tested through CI by CSN:cloud service developers into production and release it safely to CSC.

5) Continuous operation

Once the software is deployed in the production environment, the CSP adopts the continuous operation process and tools to monitor and operate the cloud service in an available and continuous environment.

6.2 Cloud service development and operation management stages

[ITU-T Y.3522] describes the lifecycle of the cloud service, including four stages: design, deployment, operation and retirement, which are shown in Figure 6-1. But the use cases analysis and corresponding functional requirements of cloud service development and operation management are not considered. This Recommendation not only details the cloud service development and operation management stages, but also clarifies the corresponding management requirements.

The lifecycle of development and operation management for cloud service includes eight stages, where each stage is iterative. The tasks of each stage depend on the cloud service characteristics and software development demands. Each stage defined in Figure 6-1 is described as follows:

1) Stage 1: Plan

At this stage, CSN:cloud service developers determine the functions of cloud services according to the demands of CSC and define the delivery plan.

2) Stage 2: Design

At this stage, CSN:cloud service developers design the functions of cloud services, which are scalable, testable and observable.

3) Stage 3: Code

At this stage, CSN:cloud service developers write the codes based on the design in the design stage.

4) Stage 4: Build

At this stage, CSN:cloud service developers submit the codes to the version control system, in which the codes are automatically compiled and analysed.

5) Stage 5: Test

At this stage, CSN:cloud service developers write testing cases and perform the testing, and then conduct acceptance of the system functions. The acceptance conclusions include pass or fail, and the acceptance methods include manual and automatic approaches.

6) Stage 6: Deployment and release

At this stage, CSN:cloud service developers install the corresponding software package into the production environment and configure it, and then CSP releases it to CSC.

7) Stage 7: Operation

At this stage, CSP monitors the events and changes of the cloud service so as to ensure its continuity and better user experience.

8) Stage 8: Measurement and feedback

At this stage, the CSC provides the feedback continuously to CSP. Meanwhile, CSP measures and analyses feedback of CSC and provides new demands to the Plan Stage.

6.3 Relationship with cloud computing reference architecture

[ITU-T Y.3502] describes the roles and activities of CCRA in detail. Although CCRA does not describe any specific cloud service development and operation management method, some of its functional components can be used in different cloud service development and operation management stages. For example, based on the build management functional component of CCRA, handling the building of a ready-to-deploy software package which can be passed to the CSP for deployment into the cloud service environment, the build stage defined in clause 6.2 deals with the building from source code to package for the test environment, including continual code compilation and analysis.

7 Functional requirements of cloud service development and operation management

This clause provides the functional requirements of cloud service development and operation derived from the corresponding use cases presented in Appendix I.

7.1 Requirements analysis

It is recommended that CSP collects and analyses CSC's demands and translates them into the system requirements of cloud service.

7.2 Development plan

It is recommended that CSN:cloud service developers schedule the development tasks into iterative plans.

7.3 Function design

It is recommended that CSN:cloud service developers design the function of cloud service based on the development plan.

7.4 Code development

It is recommended that CSN:cloud service developers develop the software source code based on the function design.

7.5 Automated testing

It is recommended that CSN:cloud service developers perform the automated tests, which is an automatic method to test the availability of delivered functions, in order to verify as early as possible whether the delivered functions satisfy the requirements of cloud service.

7.6 Continuous integration

It is recommended that CSN:cloud service developers perform the integration earlier in the development period, by which the codes can be built, tested and integrated more regularly to ensure that the software is always in the working state.

It is recommended that CSP builds a unified pre-integrated resource repository.

NOTE – The resource repository includes software repository, supporting resource repository, testing case repository, configuration repository, etc., and performs repository control continuously in the development and integration process.

It is recommended that CSP creates an open platform to integrate CSN:cloud service developers' components, such as software images, testing suites, etc., and define standard interfaces.

It is recommended CSN:cloud service developers provide the related developing configurations, testing suites and parameters to the resource repository and dock to the open platform in integration process.

7.7 Continuous delivery

It is recommended that CSP eliminates the service downtime without redundant resources and human intervention during the software change process, including upgrading and downgrading.

It is recommended that CSP creates a process engine for delivery.

NOTE – The process engine integrates planning, configuration, deployment, and test verification to support the flow of CD.

7.8 Monitoring state

It is recommended that CSP monitors the status of cloud service in order to identify risks in a timely manner.

It is recommended that CSP converts the monitored data into usable and required forms.
It is recommended that CSP processes the monitored data.

NOTE 1 – The methods of data processing include data pre-processing, streaming data processing, batch data processing, etc.

It is recommended that CSP stores the monitored data for future use.

It is recommended that CSP supports to process the monitored data for a variety of application scenarios.

NOTE 2 – The monitored data for application scenario is used for making decisions, warning of risks, alarming in case of accident, etc.

7.9 Quality assurance

It is recommended that CSP implements quality assurance to guarantee high quality of software.

It is recommended that CSN:cloud service developers schedule a plan for testing.

It is recommended that CSN:cloud service developers execute testing according to the testing plan.

It is recommended that CSN:cloud service developers report defects to CSP.

7.10 Deployment and release management

It is recommended that CSN:cloud service developers deploy a set of software code, application, configuration, and data changes to the development environment, testing environment and production environment.

It is recommended that CSP releases the completed cloud services to CSC.

7.11 Measurement and feedback

It is recommended that CSP measures the situations of all stages and processes in cloud service development and operation framework and collects feedbacks from CSC for solving the problems.

8 Security consideration

Security aspects for consideration within the cloud computing environment, which are addressed by security challenges for CSPs, are described in [ITU-T X.1601]. [ITU-T X.1601] analyses security threats and challenges, and describes security capabilities that could mitigate these threats and satisfy the security challenges.

Appendix I

Use cases of cloud service development and operation management

(This appendix does not form an integral part of this Recommendation.)

I.1 Use case template

The use cases developed in this appendix should adopt the following unified format for consistent readability and convenient material organization.

Table I.1 – Use case template

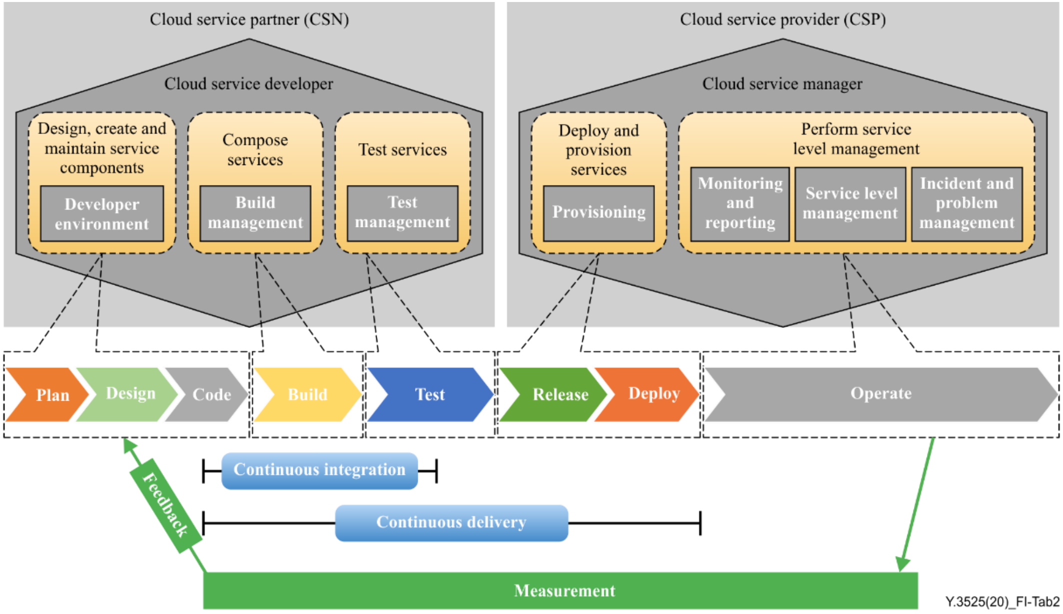
Title	Title of the use case
Description	Scenario description of the use case
Roles	Roles involved in the use case
Figure (optional)	Figure to explain the use case, but is not mandatory
Pre-conditions (optional)	The necessary pre-conditions that should be achieved before starting the use case
Post-conditions (optional)	The post-condition that will be carried out after the termination of current use case
Derived requirements	Requirements derived from the use cases, whose detailed descriptions are presented in the dedicated clauses

I.2 Use case of continuous integration/continuous delivery for cloud service

This use case illustrates the process of development and operation management for the cloud service.

Table I.2 – Continuous integration/continuous delivery for cloud service

Title	Continuous integration/continuous delivery for cloud service
Description	First, CSN:cloud service developers analyse the CSC's demands and define particular tasks. Based on this, the CSN:cloud service developers provide a development plan. This plan and corresponding tasks are created on the development platform. Then, a series of development practices based on CI is utilized to ensure that codes of software can be integrated, tested and deployed. After that, codes pass a rigorous automated testing to verify whether the requirements of cloud services are satisfied. Only the codes that positively pass the automated testing are deployed to the real environment through a deployment pipeline defined in CD. After delivery codes to production environment, the CSP monitors automatically its states to receive feedback for quality assurance. CSP is responsible for cloud service continuity according to the service level agreement (SLA) between CSP and CSC. Finally, the CSP delivers the cloud service to CSC.
Roles/sub roles	CSN, CSP, CSC

Figure	 Cloud service partner (CSN) Cloud service developer Design, create and maintain service components Developer environment Compose services Build management Test services Test management Cloud service provider (CSP) Cloud service manager Deploy and provision services Provisioning Perform service level management Monitoring and reporting Service level management Incident and problem management Plan Design Code Build Test Release Deploy Operate Continuous integration Feedback Continuous delivery Measurement Y.3525(20)_FI-Tab2
Pre-conditions (optional)	
Post-Conditions (optional)	
Derived requirements	– development plan (refer to clause 7.2) – automated testing (refer to clause 7.5) – continuous integration (refer to clause 7.6) – continuous delivery (refer to clause 7.7) – monitoring state (refer to clause 7.8) – quality assurance (refer to clause 7.9)

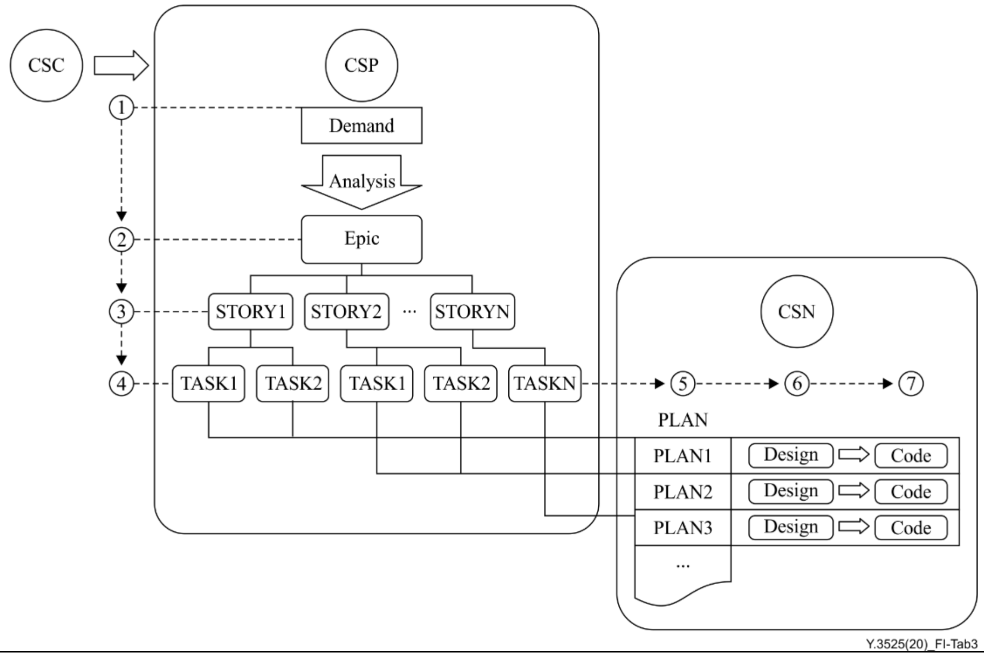
I.3 Use case of development management

This use case illustrates the process of cloud service development management.

Table I.3 –Use case of cloud service development management

Title	Use case of cloud service development management
Description	The CSP collects the demands of CSC and derives the requirement of cloud service, and then performs the development management process to implement the requirements. 1 CSP clarifies and analyses CSC's demands and takes it as an epic. 2 CSP disassembles the epic into user stories and translates CSC demands into system requirements of cloud service. 3 CSP disassembles the CSC user story into small tasks, so that the CSP could implement the tasks. 4 CSP schedules tasks into iterative plan to implement and deliver tasks frequently. 5 After plan 1 is finished, CSN:cloud service developers confirm the task of plan 1 and evaluate the workload, and the start to do plan 2, while each plan is an iteration. 6 CSN:cloud service developers design the system architecture and database schema according to the tasks. 7 CSN:cloud service developers write the codes to implement the tasks.
Roles	CSC, CSP, CSN

Table I.3 –Use case of cloud service development management

Figure (optional)	 CSC CSP Demand Analysis Epic STORY1 STORY2 ... STORYN TASK1 TASK2 TASK1 TASK2 TASKN CSN PLAN PLAN1 PLAN2 PLAN3 ... Design Code 1 2 3 4 5 6 7 Y.3525(20)_FI-Tab3
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– requirements analysis (refer to clause 7.1) – development plan (refer to clause 7.2) – function design (refer to clause 7.3) – code development (refer to clause 7.4)

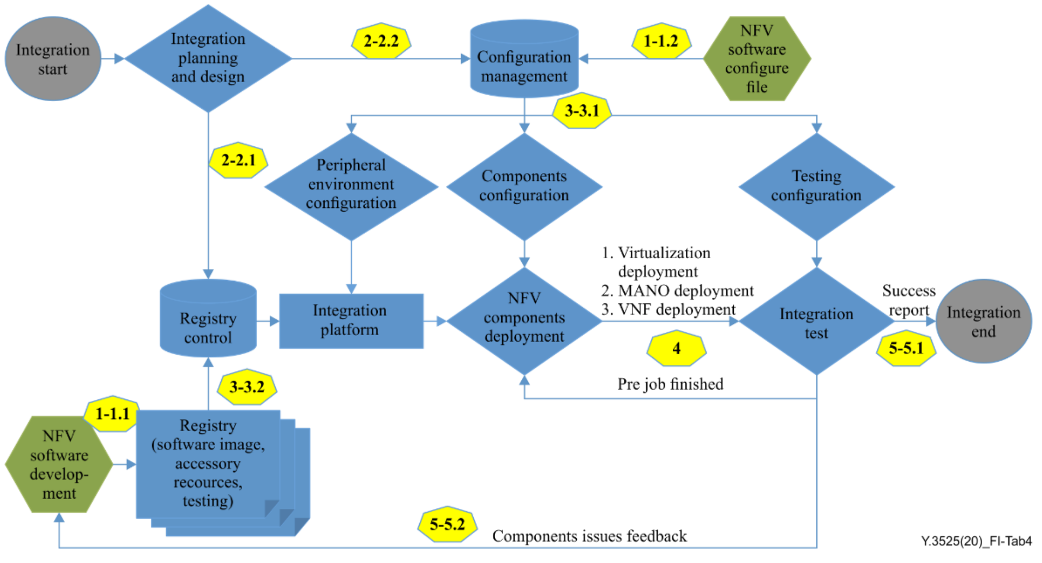
I.4 NFV development/integration/verification closed loop system

This use case illustrates network function virtualization (NFV) software component development, integration and verification to form a closed loop process.

Table I.4 – NFV development/integration/verification closed loop system

Title	NFV development/integration/verification closed loop system
Description	NFV software component development/integration/verification are combined to form a closed loop process. The whole system is divided into several steps as follows: 1 NFV related components development and configuration (performed by CSN) CSN:cloud service developers take the software development responsibility and upload the software images and configuration files into the repository of integration platform which is provided by CSP. 1.1 NFV components development – CSN:cloud service developers develop the NFV components including software images, physical devices and testing suites. 1.2 NFV components related configuration file development – CSN:cloud service developers develop component configuration files, scripts or plug-ins for software installation.

Table I.4 – NFV development/integration/verification closed loop system

	2 NFV integration system planning and design (performed by CSN) 2.1 NFV components selection – CSN:cloud service developers select integration components images from the integrated resource repository, including virtualized infrastructure software, MANO, VNFs, etc. 2.2 NFV components configuration selection – CSN:cloud service developers implement relevant components configuration. 3 System configuration and components images preparation (performed by CSP) 3.1 System configuration – CSP keeps the software and hardware configuration of integrated components configuration, integrated peripheral environment configuration and test environment configuration in resource repository. 3.2 components images preparation – CSP builds the related images into integration environments in pre-integrated open platform according to the design. 4 NFV integration deployment/verification (performed by CSN) – CSN drives the joint debugging of each layer of the integrated system, including the virtualized infrastructure, MANO, VNFs and other components in end-to-end integrated systems to implement functional verification. 5 Integration verification results (performed by CSN) 5.1 If functional verification is successful, CSN:cloud service developers get the NFV integration verification report and promote to CD process. 5.2 If functional verification fails with some issues, the issues are fed back to the image development process, and the loop process is continued.
Roles	CSP, CSN
Figure (optional)	 Integration start Integration planning and design 2-2.2 Configuration management 1-1.2 NFV software configure file 3-3.1 2-2.1 Peripheral environment configuration Components configuration Testing configuration 1. Virtualization deployment 2. MANO deployment 3. VNF deployment Registry control Integration platform NFV components deployment Integration test Success report Integration end 4 5-5.1 Pre job finished 3-3.2 1-1.1 NFV software develop-ment Registry (software image, accessory recources, testing) 5-5.2 Components issues feedback Y.3525(20)_FI-Tab4
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– continuous integration (refer to clause 7.6) – continuous delivery (refer to clause 7.7)

I.5 Use case of VNF change management without interruption

This use case illustrates the scenario for VNF change management without interruption.

Table I.5 – VNF change management without interruption

Title	VNF change management without interruption
Description	With the maturity of NFV, more and more network functions have been realized by software. These are known as virtualized network functions (VNFs), and network as a service (NaaS) CSP has to face the problem of large software management. Traditionally, NaaS CSP needs to provision an entire replica of the production VNF on a new cluster of virtual machines (VMs)/containers, install the new software on the replica VMs/containers, switch all the traffic from existing VMs/container of the VNF to the replica resources, and then release the old VMs/container back to the resource pool. The whole change process requires the double VMs/containers and is typically done at time when there is less traffic due to the short time period interruption. The above traditional software upgrading approach cannot satisfy the frequent request change and service continuation. It is needed to figure out the challenge by changing the software delivery manner in order to reduce human intervention and redundant resources during the VNF upgrading.
Roles	CSP, CSN, CSC
Figure (optional)	
Pre-conditions (optional)	CSN:cloud service developers provide the implementation of VNFs and are responsible for their continuous integration. CSP is responsible for the lifecycle management of VNFs during the execution time.
Post-conditions (optional)	VNFs are changed (upgrading/downgrading) without interruption and less resources are required during the delivery process, compared with the traditional upgrading approach.
Derived requirements	– continuous delivery (refer to clause 7.7)

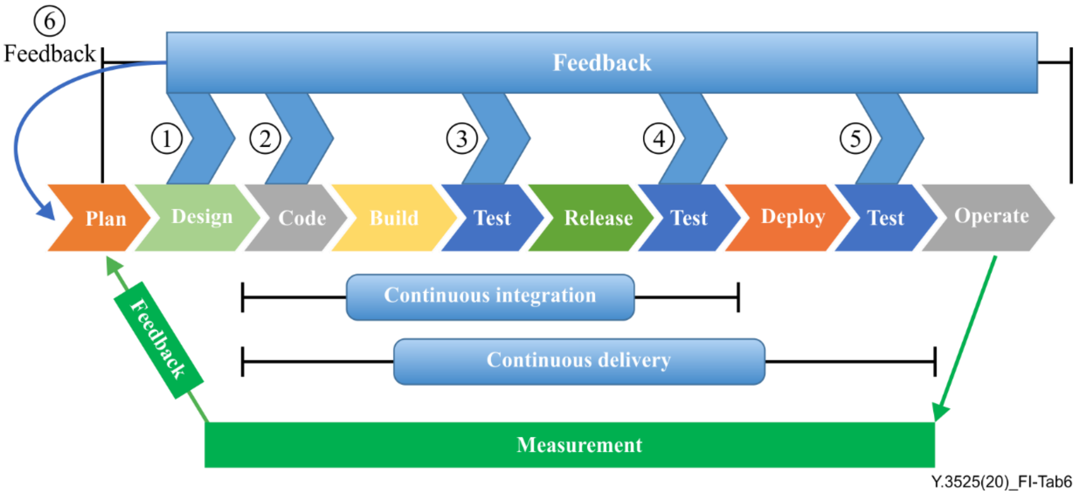
I.6 Use case of automated testing

This use case illustrates which steps in the cloud service development and operations management process need to execute automated testing.

Table I.6 – Use case of automated testing

Title	Use case of automated testing
Description	The testing designed by CSN:cloud service developers should include test methods, test strategies, test plans, test cases, etc. Due to accelerated releasing speed, it is necessary to increase the proportion of automated testing to shorten the testing duration and improve the test efficiency. Automated testing needs to be performed at the stage of requirements analysis, integration and delivery in order to detect and fix problems immediately. The CSN:cloud service developers develop code and write test cases simultaneously and begin to execute test cases continuously. In order to achieve the target of test-driven development early, the automated testing are performed much earlier than that in the traditional development process. The steps for automated testing are as follows: 1 The CSN:cloud service developers designed the test strategies and test cases. 2 CSN:cloud service developers write test cases.

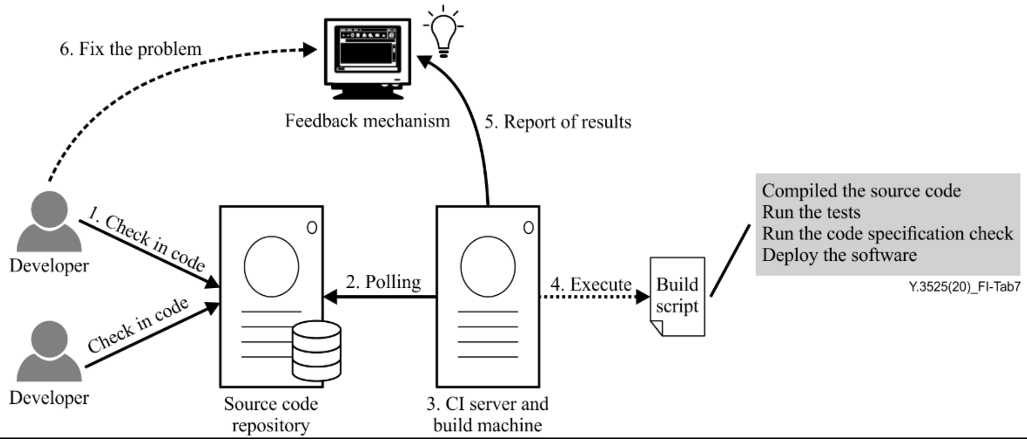
Table I.6 – Use case of automated testing

Title	Use case of automated testing
	3 When the CSN:cloud service developers is in the stage of commit building, the unit test after committing the source code is triggered simultaneously. 4 When in the stage of CI, the CSN:cloud service developers execute acceptance testing, including functional integration testing, non-functional integration testing, regression testing, etc., to verify whether the requirements of the cloud service are satisfied and provide the problem as the feedback. 5 In the stage of release, the verified software package in CI process is deployed into the production environment for joint testing before formal deployment. 6 CSP deploys the software package into the production environment and performs continuous monitoring, and the problems from the CSC are collected to CSP.
Roles	CSP, CSC, CSN
Figure (optional)	 ⑥ Feedback Feedback ① ② ③ ④ ⑤ Plan Design Code Build Test Release Test Deploy Test Operate Continuous integration Continuous delivery Feedback Measurement Y.3525(20)_FI-Tab6
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– Automated testing (refer to clause 7.5)

I.7 Use case of continuous integration

This use case illustrates a typical continuous integration process.

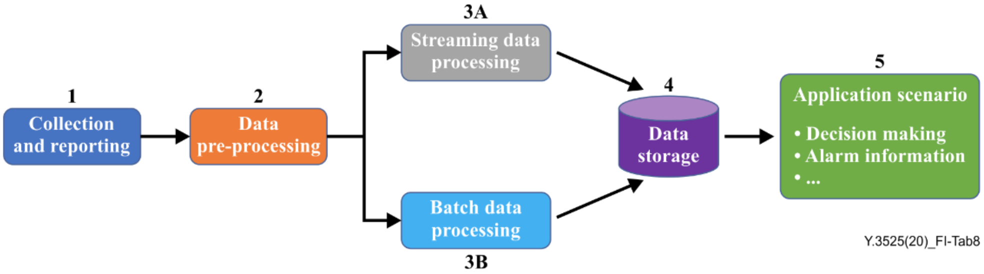
Table I.7 – Use case of continuous integration

Title	Use case of continuous integration
Description	The CSP requests the CSN:cloud service developers to provide CI. When the CSN: cloud service developers submit codes, the automated code integration is automatically triggered. These newly submitted codes need to be compiled and automatically tested before they are finally merged into the trunk. In the process of CI, the results of automated testing verification are quite important to ensure that all the submitted codes have no problems after merged into the trunk. The procedures of one execution of CI are as follows: 1 CSN:cloud service developers check in codes to the source code repository. 2 CI server is triggered regularly (such as a fixed time interval or event triggered by some special event) to poll the source code repository and find whether the code is changed or not, and the polling results are fed back to CI server. 3 The CI server checks out the latest code automatically to dedicated server (if the application scale is small, the CI server can be utilized repeatedly). 4 The building scripts or commands specified by the CSN:cloud service developers run to check the latest codes (such as dynamic and static code scans, compile and package, run unit tests, deployment and functional tests, etc.) 5 Finally, the report of results (success or failure) is fed back. 6 If CSN:cloud service developers receive an exception notification, the problem is immediately fixed to ensure the availability of CI.
Roles	CSP, CSN
Figure (optional)	 6. Fix the problem Feedback mechanism 5. Report of results Compiled the source code Run the tests Run the code specification check Deploy the software Y.3525(20)_FI-Tab7 Developer 1. Check in code Check in code Developer 2. Polling 4. Execute Build script Source code repository 3. CI server and build machine
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– continuous integration (refer to clause 7.6)

I.8 Use case of monitoring state

This use case illustrates a monitoring system to monitor the software layer metrics and infrastructure layer metrics.

Table I.8 – Use case of monitoring state

Title	Use case of monitoring state
Description	CSP builds a monitoring system to monitor the cloud services delivered to CSC. The monitoring system needs to have the functions of data collection, data pre-processing, streaming data processing, batch data processing, storage and alarms, where each step involves a lot of data processing and calculation. The specific procedures are as follows: 1 Collection and reporting: The CSP collects and reports the pre-defined event data of cloud service. 2 Data pre-processing: CSP performs the data collection and data pre-processing. 3A Streaming data processing: CSP obtains monitoring data continuously, and carries out real-time data processing and outputs to data storage. 3B Batch data processing: CSP processes the data in the batch manner, and outputs the processing results to the data storage. 4 Data storage: CSP stores the structured data into the storage system. 5 Application scenario: CSP obtains the processed data from the storage system for decision making, alarm information and other operation and maintenance scenarios.
Roles	CSC, CSP
Figure (optional)	 3A Streaming data processing 1 Collection and reporting 2 Data pre-processing 4 Data storage 5 Application scenario • Decision making • Alarm information • ... Batch data processing 3B Y.3525(20)_FI-Tab8
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– Monitoring state (refer to clause 7.8)

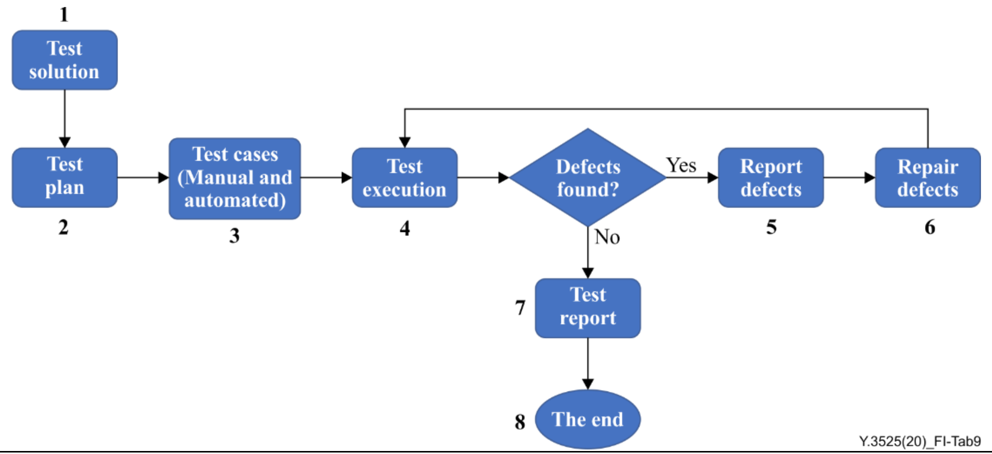
I.9 Use case of quality assurance

This use case illustrates the process of quality assurance.

Table I.9 – Use case of quality assurance

Title	Use case of quality assurance
Description	The CSP requires CSN:cloud service developers to perform tests to ensure that software quality satisfies CSC expectations. The process includes functional and non-functional testing via manual or automated testing manner. The procedures of quality assurance are as follows: 1 The test solution is created. 2 The test plan is provided.

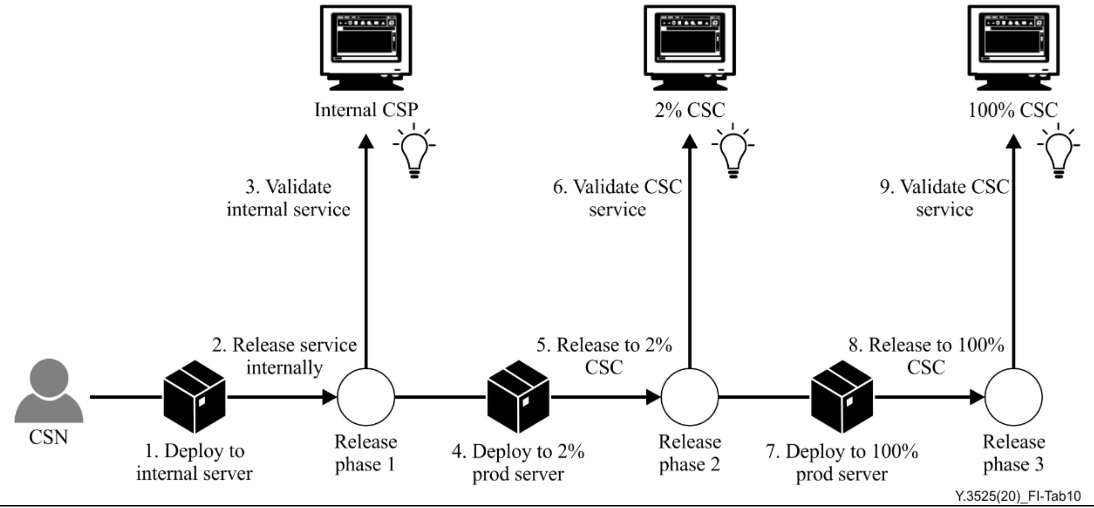
Table I.9 – Use case of quality assurance

Title	Use case of quality assurance
	3 The test cases (manual and automated) are created. 4 Test execution is performed. 5 The report of defects is generated. 6 The defects are repaired. 7 The testing report is generated. 8 Quality assurance process ends.
Roles	CSP, CSN, CSC
Figure (optional)	 1 Test solution Test plan 2 Test cases (Manual and automated) 3 Test execution 4 Defects found? Yes No Report defects 5 Repair defects 6 7 Test report 8 The end Y.3525(20)_FI-Tab9
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– quality assurance (refer to clause 7.9)

I.10 Use case of deployment and release management

This use case illustrates the process of deployment and release management.

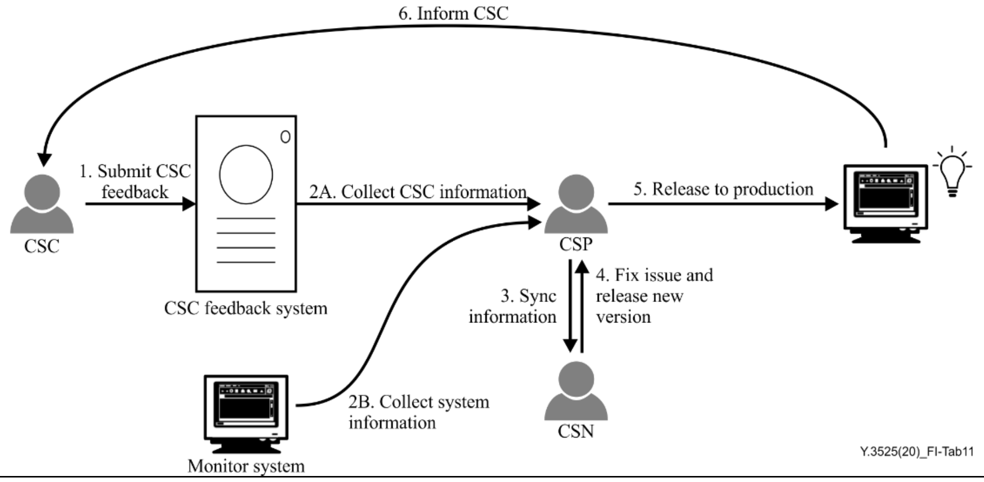
Table I.10 – Use case of deployment and release management

Title	Use case of deployment and release management
Description	After the CSN:cloud service developers complete the development of the new version of cloud services, they utilize the testing model (i.e., Canary model) to verify the reliability of the new functions. The deployment and release management steps are as follows: 1 The CSN:cloud service developers finish the development of a new version of cloud service and deploy it in the internal servers of CSP; 2 The CSP releases new version of cloud service internally; 3 The CSP validates the availability of cloud services in the internal environment; 4 After passing the validation in the internal testing environment, the CSN:cloud service developers deploy the new version of cloud service to a limited number of production servers, (e.g., 2% servers of CSC). 5 The CSP releases the new version of cloud services to a limited number of CSCs. 6 The CSP uses online monitoring as well as the other methods to validate the availability of the new version of cloud services; 7 After the new version of cloud services pass the production environment verification, the CSN:cloud service developers deploy the new version of cloud services to more production servers, (e.g., 100% production servers of CSC); 8 The CSP releases the new version of cloud services to all CSCs (e.g., 100% of CSCs); 9 The CSP validates the new version of cloud services.
Roles	CSP, CSC, CSN
Figure (optional)	 Internal CSP 2% CSC 100% CSC 3. Validate internal service 6. Validate CSC service 9. Validate CSC service 2. Release service internally 5. Release to 2% CSC 8. Release to 100% CSC CSN 1. Deploy to internal server Release phase 1 4. Deploy to 2% prod server Release phase 2 7. Deploy to 100% prod server Release phase 3 Y.3525(20)_FI-Tab10
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– deployment and release management (refer to clause 7.10)

I.11 Use case of measurement and feedback

This use case illustrates the closed loop process of measurement and feedback.

Table I.11 – Use case of measurement and feedback

Title	Use case of measurement and feedback
Description	After CSP's officially release the new cloud service, the CSCs use the service and provide their feedbacks to CSP through a dedicated feedback system. The CSP combines the feedbacks and related information together to complete the issue description. The CSP fixes the issue and provides new releases of the cloud service. 1 The CSC submit their feedback problems of new cloud services through the CSC feedback system; 2A The CSC feedback system collects CSC's information to the CSP; 2B The monitoring system collects system information to the CSP; 3 The CSP locates the issue based on CSC's information and system information, and synchronizes the analysed information to the CSN:cloud service developers; 4 The CSN:cloud service developers fix the issue, and release a new version to CSP after verification; 5 The CSP releases the new version of the repaired application to the production environment; 6 The CSP informs the CSC after the new version is released;
Roles	CSP, CSC, CSN
Figure (optional)	 6. Inform CSC 1. Submit CSC feedback CSC CSC feedback system 2A. Collect CSC information CSP 5. Release to production 3. Sync information 4. Fix issue and release new version CSN 2B. Collect system information Monitor system Y.3525(20)_FI-Tab11
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– measurement and feedback (refer to clause 7.11)

Bibliography

[b-ITU-T Y.3500] Recommendation ITU-T Y.3500 (2014) | ISO/IEC 17788:2014, *Information technology – Cloud computing – Overview and vocabulary*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2020