

Release management ITIL®4 Practice Guide

[AXELOS.com](https://www.axelos.com)

Contents

1	About this document	3
2	General information	4
3	Value Streams and Processes	12
4	Organizations and people	19
5	Information and technology	24
6	Partners and suppliers	27
7	Important reminder	28
8	Acknowledgments	29

1 About this document

This document provides practical guidance for the release management practice. It is split into five main sections, which cover:

- general information about the practice
- the practice's processes and activities and their roles in the service value chain
- the organizations and people involved in the practice
- the information and technology supporting the practice
- considerations for partners and suppliers for the practice

1.1 ITIL 4 QUALIFICATION SCHEME

Selected content of this document is examinable as a part of the following syllabus:

- **ITIL Specialist** Create, deliver and support
- **ITIL Specialist** High Velocity IT

Please refer to the syllabus document for details.

2 General information

2.1 PURPOSE AND DESCRIPTION

Key message

The purpose of the release management practice is to make new and changed services and features available for use.

Release management practice ensures that services are available to use in line with organization's policies and agreements between the organization and its service consumers.

Traditionally, service components are visible and accessible for users including infrastructure, software, and documentation. As infrastructure and documentation are increasingly digitized, software management methods and approaches become more applicable to these types of service components. This affects the release management practice and other practices with significant focus on software such as service validation and testing, deployment management, and software development and management.

From the customer and user journey perspective, release management supports onboarding and offboarding. For users, this practice may support the very first touchpoints and interactions with the service provider. After initial onboarding is complete, this practice supports the delivery of service updates, which is important for the success of the practice.

2.2 KEY TERMS AND CONCEPTS

2.2.1 Release management and deployment management

Release

A version of a service or any other configuration item, or a collection of configuration items that is made available for use

Organizations should define a high-level approach to release and deployment management practices and their role in organization's value streams and service relationships.

One approach is to combine release and deployment activities; once moved to the operational environment, service components become available to users. Co-existence of different versions of one component in live environment is rare and does not last long. There is no clear border between release and deployment activities (and steps of a product lifecycle). This approach can often be applied to hardware service components, and large monolithic software systems.

Another approach is applicable to the Agile digital environment, modern architecture, and cloud-based digital solutions. In this approach, new versions of software can be deployed to the live environment before release activities start, and then released to all or some of the users. In this case, release management activities are focused on enabling service usage and can be very simple and technical (such as changing application's status in a repository so it is available for download

by a selected audience), or complex and human-focused (such as training of users to reduce risks and increase effectiveness of the version changes).

CI/CD and release management

The key concepts for deployment in agile and DevOps are continuous integration, continuous delivery, and continuous deployment. Martin Fowler[1] defines them as:

- Continuous integration usually refers to integrating, building, and testing codes within the software development environment.
- Continuous delivery extends this integration, covering the final stages for production deployment. Continuous delivery means that built software can be released to production at any time.
- Continuous deployment refers to the changes that go through the process and are automatically put into production. This enables multiple production deployments a day. Continuous delivery means that frequent deployments are possible, but deployment decisions are taken case by case, usually due to businesses preferring a slower rate of deployment. Continuous deployment requires that continuous delivery is being done.

In organizations, using continuous deployment management for releases as a separate practice is common and effective; new versions of software, documents, and digital infrastructure are deployed to live environments as soon as they are ready, and then release management practice is used to 'switch them on' for users.

If continuous delivery is used without continuous deployment, deployment and new and changed release components may be synchronized and managed as a single step in respective value streams.

Finally, if an organization does not use continuous delivery or continuous deployment, release management activities are more likely to be combined with deployment management.

Organizations define the approach to release and deployment management practices for all products and services, or per product. This is usually defined by organization's product architecture (and its consistency across products), and by organization's approaches to management of software lifecycle.

2.2.2 Release management approaches, models and plans

If an organization manages different architecture products, it is likely that several approaches for release management will be defined. A product-specific release management model can be developed once an approach is agreed for a specific product. This model includes, but is not limited to:

- agreed high-level approach
- target user audience of releases and rules for user enablement
- release units and packaging rules
- push/pull conditions
- verification and acceptance criteria
- terms and conditions of release usage for hypothesis verification and experimentation.

It is possible to have more than one release management model for a product. For example, when a product is used to provide services on different markets or to business and individual service consumers.

One of the factors that is affecting the development of the release management model and the practice, is the organization's scope of control of the product. When organization's control the full product lifecycle, including development and deployment, it has more freedom in defining release management models. In contrast, if the organization's services are based on third-party components, or the development and deployment are managed by a supplier, it usually introduces constraints that the organization should consider. It still may be able to decide whether to include updated components in its services, but only to a certain extent (until components' vendor allows to keep using previous versions).

2.2.3 Release units

Release unit

A pre-defined set of configuration items or parts of configuration items that is the basic size to be included into a release

Release units may include different types of software components, user equipment, and other hardware, documents. Release unit for the initial release of a service to new users, can be different from release unit for updates of the same service. However, some combinations of components may be recommended or even mandated. For example, every update should include published release notes for users; however, in some cases, user equipment should be updated after its initial release to users.

Some release instances may include incomplete release units, but should be an exception: either a release is urgent (emergency update), or too complex and an impractical release unit has been defined.

It is important to remember that a release unit may be different from a deployment unit, which defines components that are normally deployed together. Releases are user-facing, and the definition of a release unit depends on which components of service affect users' ability to use the service and user experience in general.

2.2.4 Push/pull conditions

One of the decisions made during the development of the release management model is whether new versions of service components will be pushed to users, pulled by users, or there will be a mix of the approaches.

A ‘push’ approach implies that new or changed components of services are enabled for users without their specific consent, and users are obliged to use these versions. In contrast, the ‘pull’ approach makes new components and services available to users, but users can decide whether they prefer using these new versions, stick to older ones, or not using the service at all.

Typically, organizations do not apply a single approach; instead, they define conditions where the ‘pull’ or ‘push’ approach would work better. Considerations are common for internal and external service provision. This includes:

- the benefits of having a single version across the user base (maintainability, compatibility)
- the benefits of allowing users to have more freedom (better image, flexible pricing options)
- technical and organizational ability to manage multiple versions in a live environment
- critical changes (an update removing critical security vulnerability is likely to be ‘pushed’)
- functional and other customer’s requirements (if a required new functionality is implemented, customers may mandate the update for all users)
- regulatory requirements.

2.2.5 Hypothesis testing and experimentation

Release management may be used to validate a hypothesis and an experiment. When an organization needs to test a hypothesis with a sample user audience, testable services may be released to sample groups (sometimes called treatment groups). This approach is widely used by providers of mass services, such as social networks, but also applied to small user groups. Related techniques include blue/green releases, canary releases, and A/B testing.

These experiments require the involvement of other practices. This includes, but not limited to:

- infrastructure and platform management
- software development and management
- deployment management
- architecture management
- service desk
- incident management
- release management.

2.3 SCOPE OF THE PRACTICE

The scope of the release management practice includes the following:

- Development and maintenance of the organization’s approach to release new and changed services¹ and components.
- Management and coordination of all release instances in line with the defined approach, from planning, to implementation, and review.

¹ Removal of services and components from users is included in ‘new and changed’ here.

There are a number of activities and areas of responsibility that are closely associated with release management, but not part of the scope of the practice.

Some of those key areas are listed in Table 2.1, and includes references to the practices in which they can be found. It is important to remember that ITIL practices are merely collections of tools to use in the context of value streams, and they should be combined as necessary depending on the situation.

Table 2.1 Release related activities described in other practice guides

Activity	Practice guide
Authorization of changes/releases	Change enablement
Deployment of new and changed components and services in live environment	Deployment management
Development of software	Software development and management
Development and building of infrastructure components	Infrastructure and platform management
User training support and operational staff training	Workforce and talent management
Testing and validating the services and service components	Service validation and testing
Naming, versioning and control of the service components	Service configuration management
Management of organizational changes related to large-scale releases	Organizational change management
Management of projects	Project management

2.4 PRACTICE SUCCESS FACTORS

A practice success factor (PSF) is a complex functional component of a practice that is required for the practice to fulfil its purpose.

A PSF is more than a task or activity, as it includes components of all four dimensions of service management. The nature of the activities and resources of PSFs within a practice may differ, but together they ensure that the practice is effective.

The release management practice includes the following PSFs:

- establishing and maintaining effective approaches to the release of services and service components across the organization
- ensuring an effective release of services and service components in the context of the organization's value streams and service relationships.

2.4.1 Establishing and maintaining effective approaches to the release of services and service components across the organization

Release management practice includes defining and agreeing approaches and models to follow for the release of new and changed services and service components. Organizations are likely to combine several approaches and to define several release management models for every product they manage.

Apart from organization's and product's specifics, release models are defined by service relationships between the organization and its service consumers. This includes factors such as:

- internal or external service consumers
- individual or corporate service consumption
- out-of-the-box or tailored services

See *ITIL® 4: Drive Stakeholder Value* for more details on how these factors influence service provision.

The approaches and models for release management should have some flexibility to adapt to changing circumstances, such as scale, urgency, or complexity. A plan for every release instance may be developed based on one of the agreed models to reflect the specifics of the release instance.

Release approaches, models, and the practice in general, should be subject to continual improvement, constantly looking for ways to eliminate waste and increase effectiveness and efficiency.

2.4.2 Ensuring an effective release of services and service components in the context of the organization's value streams and service relationships

Ensuring an effective release may require organizing resources in all four dimensions of service management.

Depending on the release management model, activities and resources that are required to implement a release instance, vary significantly:

- A release of a new version of mobile application for all users in a certain country or region, may be performed by a changing status of the previously deployed version of the software, related release notes, and user documentation; and informing relevant stakeholders within the service provider organization. No further actions may be required.
- A release of a new custom-made ERP system with on premises installation and a need for user equipment upgrade may be managed as a large-scale project, involving many teams and practices across and from outside of the organization.

In any case, effective coordination, use of automation, and good planning of the release model from the early steps of product lifecycle are crucial for the success of release.

This practice is focused on identifying the tasks and coordinating the participants. It also provides recommendations on procedures and techniques to use during release implementation. Therefore, an effective combination of practices and cooperation from teams is necessary during the implementation.

Effective coordination of software development and management, infrastructure and platform management, deployment management, service validation, and testing and release management is especially important.

2.5 KEY METRICS

The effectiveness and performance of the ITIL practices should be assessed within the context of the value streams to which each practice contributes. As with the performance of any tool, the practice's performance can only be assessed within the context of its application. However, tools can differ greatly in design and quality, and these differences define a tool's potential or capability to be effective when used according to its purpose.

The same applies to practices, their performance should be assessed in the context of value streams but their potential is defined by their design and the quality of the resources. Further guidance on metrics, key performance indicators (KPIs), and other techniques that can help with this can be found in the measurement and reporting practice guide.

Key metrics for release management practice are mapped to its PSFs. They can be used as KPIs in the context of value streams to assess the contribution of release management to the effectiveness and efficiency of those value streams. Some examples of key metrics are given in Table 2.2.

Table 2.2 Example of key metrics for the practice success factors

Practice success factors	Key metrics
Establishing and maintaining effective approaches to the release of services and service components across the organization	<ul style="list-style-type: none"> ● Stakeholders' satisfaction with the way new and changed services are introduced to users ● Adoption of the agreed approach to release management across the organization ● Key partners and service consumers' alignment with release management approaches and models ● Audit findings and external compliance issues caused by releases

<p>Ensuring an effective release of the services and service components in the context of the organization's value streams and service relationships</p>	<ul style="list-style-type: none">● Stakeholders' satisfaction with release instances● Percentage of successful release instances/number of release errors/failures● Number and percentage of incidents related to release● Timeliness/adherence to release schedule● Release backlog throughput
--	--

The correct selection and aggregation/segregation of metrics into composite/hierarchical indicators will make it easier to use them for the ongoing management of value streams and for the periodic assessment and continual improvement of the deployment management practice.

There is no single best solution; metrics will be based on the overall context, service strategy and priorities of an organization, as well as on the goals of the value streams to which the practice contributes.

3 Value Streams and Processes

3.1 VALUE STREAM CONTRIBUTION

Each ITIL management practice, contributes to multiple value streams in the service value chain (SVC). It is important to remember that no value stream is made up of a single practice, and each practice may contribute to more than one value streams.

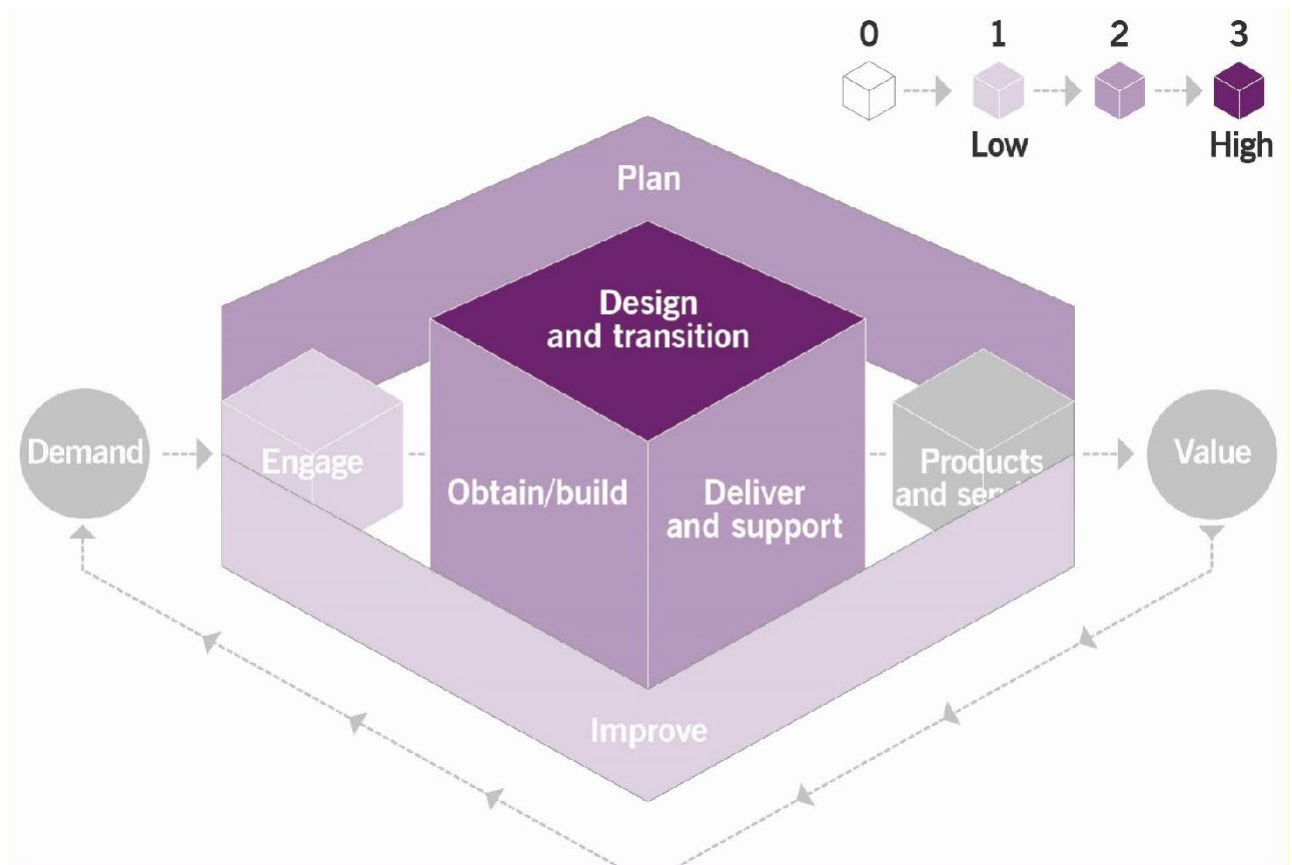


Figure 3.1. Heat map of the contribution of release management to value chain activities

The main value chain activities to which the practice contributes are:

- plan
- improve
- design and transition
- obtain/build
- deliver and support.

3.2 PROCESSES

Each practice may include one or more processes and activities that may be necessary to fulfil the purpose of that practice.

Process is a set of interrelated or interacting activities that transform inputs into outputs. A process takes one or more defined inputs and turns them into defined outputs. Processes define the sequence of actions and their dependencies.

The release management practice activities form two processes:

- release planning
- release coordination.

3.2.1 Release planning process

This process is focused on the continual improvement of the release management practice, release approaches and models, and the development of plans for complex release instances. It is performed regularly and triggered by events or requests. Regular reviews may take place every two to three months or more frequently, depending on the effectiveness of the existing models and procedures. This process includes the following activities and transforms the following inputs into outputs shown in Table 3.1.

Table 3.1 Inputs, activities, and outputs of the release planning process

Key inputs	Activities	Key outputs
<ul style="list-style-type: none"> ● Current release management approaches and models ● Release records ● Release review reports ● Policies and regulatory requirements ● Product architecture ● Service catalogue ● Service level agreements ● Incident records and reports ● IT asset information ● Agreements and contracts with suppliers and partners ● Relevant policies and plans (information security, continuity, capacity, and so on.) 	Product architecture and <ul style="list-style-type: none"> ● service relationship analysis ● Release management approach review and development ● Release management model review and development ● Release instance planning ● Release plan communication 	<ul style="list-style-type: none"> ● Updated release management approaches and models ● Release plans ● Release schedule ● Improvement initiatives ● Change requests ● Updated knowledge management articles ● Lessons learnt

Figure 3.2 shows a workflow diagram of the process.

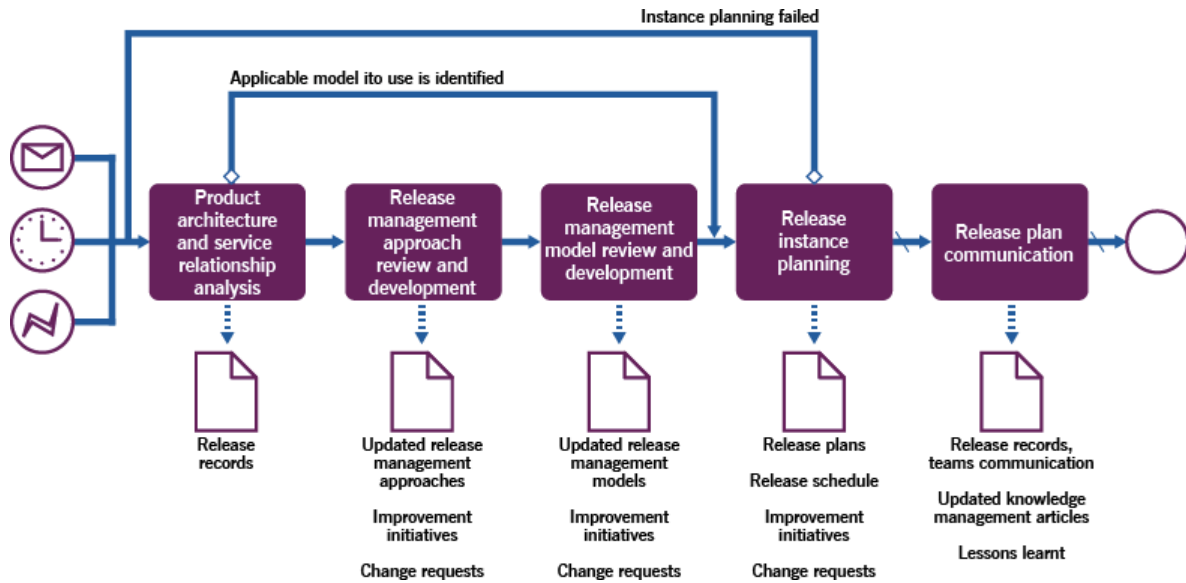


Figure 3.2 Workflow of the release planning process

Table 3.2 provides an example of the process activities.

Table 3.2 Examples of the release planning process activities

Activity	Regular review	Planning of a complex release instance
Product architecture and service relationship analysis	<p>Release manager, together with product/service owners, architects, and other teams, analyse and discuss new or changed conditions affecting release approaches:</p> <ul style="list-style-type: none"> ● preferred approach to the creation/modification of a group of products and services ● nature of the group products or services ● organization’s architecture approaches and decisions ● main release audiences and relationship with them, existing service level agreements ● organization’s risk management approach and risk appetite ● compliance, policies and technology opportunities and constraints that are present ● market position and financial conditions 	<p>Release manager, together with product/service owners, architects and other teams analyse and discuss the factors influencing the release instance.</p>

- level of control over the components of products or services

Based on the analysis and discussion, a new release approach is defined, or changes are proposed to the existing approach.

Release management approach review and development

The team discusses the new release approach or changes to the existing release approach and agree on the approach. Release approach is developed or updated.

Release manager, together with product/service owners, architects and other teams run fit/gap analysis of the existing release approaches and choose an approach suitable for the complex release instance in question.

Release management model review and development

Based on the new or changed approach, release models are defined or updated. Examples of this includes:

- release procedures
- release authorities
- template plan
- schedules templates
- templates for communications plans
- knowledge articles.

Automation scripts developed for multiple release instances.

The team should assess the risks for the release instance, while considering previous knowledge, architecture, technical debt, service level agreement, and user relationship, as well as security, availability, continuity, capacity, and financial constraints.

Based on the initial backlog assessment, the team decides on using a new or existing release model.

Release instance planning

The team plans the following for the release instance:

- target audience
 - set of components or features included into the release instance
 - sequence and methods for enabling the components/features (for example, using feature toggles), including plan for any hypothesis verification and experimentation
 - verification and acceptance criteria and user enablement
-

- (trainings, knowledge sharing, accounts preparation, and so on)
- release units and packaging rules
 - push/pull conditions.

Release plan communication	Communications for the new or updated release plans, schedules and procedures prepared, reviewed by stakeholders and fed into service desk and knowledge management.	Communications for the release plan and prepared schedule are reviewed by stakeholders and fed into service desk and knowledge management.
----------------------------	--	--

3.2.2 Release coordination process

This process includes the following activities shown in Table 3.2 and transforms the following inputs into outputs.

Table 3.2 Inputs, activities, and outputs of the release coordination process

Key inputs	Activities	Key outputs
Release management models	Identification of applicable model or plan	Released service components/services
Release plans	Verification of the service components	Release records
Release schedule	Verification of the release procedures	Release communications
Environment details	Release execution	Feedback from users, customers, and involved team members
Service component/ release components deployed to live environment or prepared for deployment	Release verification	Release review report
Acceptance and verification criteria	Release review	

Figure 3.3 shows a workflow diagram of the process.

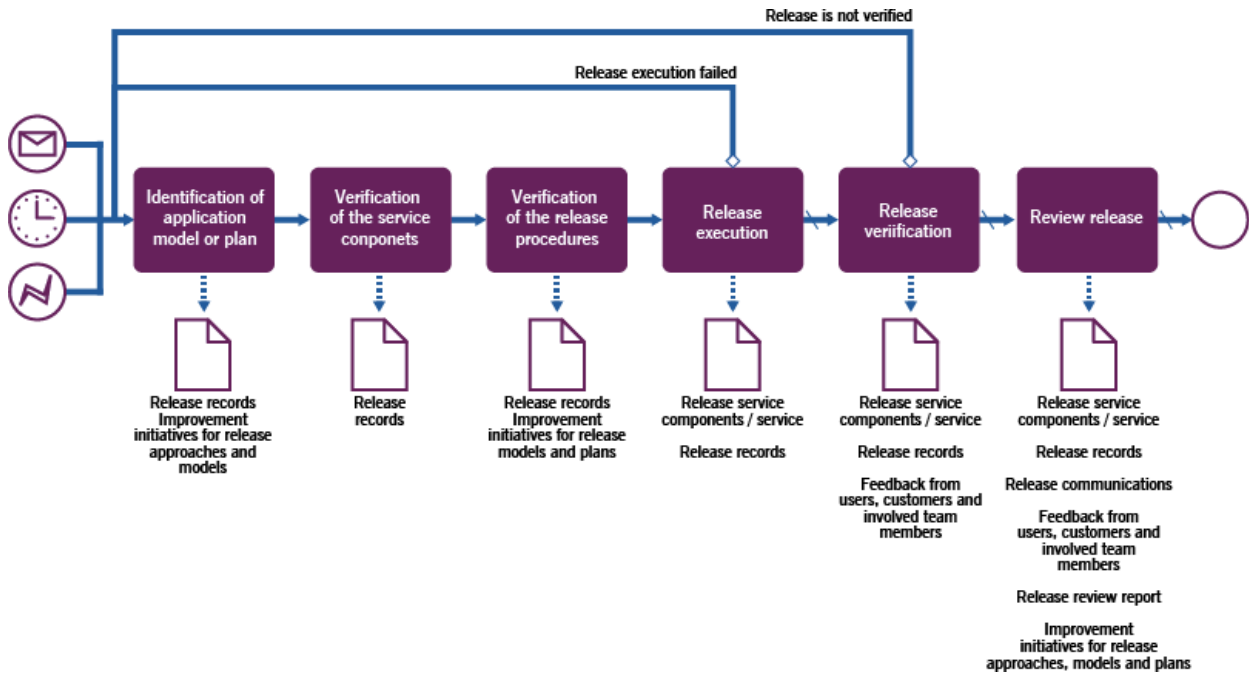


Table 3.3 shows the release coordination process activities.

Table 3.3 Release coordination process activities

Activity	Automated release of a software component	Complex release project
Identification of applicable model or plan	Release pipeline is organized so that the release model or plan is detected automatically based on the product or service, target environment, or development team.	<p>A product/service owner together with the development team assess if the product/service or sum of changed components is potentially releasable.</p> <p>The team assesses the interdependencies between the release instance and existing services, assesses risks for the release instance (including technical debt influences) and chooses an appropriate release model to use.</p> <p>The team may be updated based on the release model requirements.</p>
Verification of the service components	Release instance components run pre-defined component tests. In CI pipeline, each software change committed runs through the automated tests.	Based on the release model, component verification is administered. Some additional tests could be done based on risks assessment and technical debt.

	<p>Verification may include automatically releasing a component to the members of the development team, or a test user group for functional tests or releasing to a specialized group of users for non-functional tests, for example, experience, security, or performance tests.</p>	<p>Verification may also trigger additional building, deployment, or testing, if some of the components are not deployed according to the model.</p>
Verification of the release procedures	<p>Release procedure is chosen automatically based on the component attributes.</p>	<p>Release procedures for the chosen model are verified for the release instance. Release execution may start only when all requirements of the selected release model are met (all required resources are available and procedures are ready to run)</p>
Release execution	<p>Release executed according to pre-defined script (for example, this may be limited to granting access to an appropriate group of users or change service traffic routing to the environment containing a new feature/component) and affected user groups are informed automatically.</p>	<p>Release executed based on a trigger (for example, project team decision, product/service manager approval or consumer request) in conjunction with other relevant practices. Many internal and external teams may be involved.</p>
Release verification	<p>Automated script verifies that all features/components were released.</p>	<p>Release teams and dedicated users check that all the features/components needed were released.</p>
Release review	<p>Any exceptions and logs of the automated release process are analysed by the development team.</p> <p>The development team runs post-mortem, updates knowledge base, and records lessons learnt.</p>	<p>Feedback is gathered from the user groups.</p> <p>Release team runs release post-mortem, updates knowledge base and records lessons learnt. Resulting release review report may trigger the release planning process.</p>

4 Organizations and people

4.1 ROLES, COMPETENCES AND RESPONSIBILITIES

The practice guides do not describe the practice management roles such as practice owner, practice lead, or practice coach. The practice guides focus on specialist roles specific to each practice. The structure and naming of each role may differ from organization to organization, so any roles defined in ITIL should not be treated as mandatory, or even recommended. Remember, roles are not job titles. One person can take on multiple roles and one role can be assigned to multiple people.

Roles are described in the context of processes and activities. Each role is characterized with a competence profile based on the model shown in Table 4.1.

Table 4.1 Competency codes and profiles

Competence code	Description
L	<u>Leader</u> Decision-making, delegating, overseeing other activities, providing incentives and motivation, and evaluating outcomes
A	<u>Administrator</u> Assigning and prioritizing tasks, record-keeping, ongoing reporting, and initiating basic improvements
C	<u>Coordinator/communicator</u> Coordinating multiple parties, maintaining communication between stakeholders, and running awareness campaigns
M	<u>Methods and techniques expert</u> Designing and implementing work techniques, documenting procedures, consulting on processes, work analysis, and continual improvement
T	<u>Technical expert</u> Providing technical (IT) expertise and conducting expertise-based assignments

There is one practice-specific role in release management that may be found in the organizations: release manager. This role is often introduced in organizations where there is a significant volume of releases, especially if they need manual planning and execution. In other organizations, the responsibilities of a release manager may be taken by product or service owners.

4.1.1 Release manager role

Where a release manager role is defined, it is usually assigned to specialists that have a strong knowledge of the organization's business, products and services, technology, platforms, frameworks, and processes. The role will require strong planning and project management skills, ability, and authority to coordinate teamwork.

The competence profile for this role is AMCT. This role is usually responsible for planning, managing, and coordinating release management as a practice as well as individual release instances, including:

- reviewing and developing the release approaches and models
- promoting the adoption of the agreed release management approaches and models across the organization
- planning complex releases
- managing and communicating the release schedule
- ensuring the practice is aligned and coordinated with other practices
- reviewing and continually developing the practice.

In some complex organizations, part of the release manager's responsibilities may be delegated to the role of release coordinators.

4.2 ROLES INVOLVED IN THE RELEASE MANAGEMENT ACTIVITIES

Examples of other roles which can be involved in the release management activities are listed in Table 4.2, together with the associated competence profiles and specific skills.

Table 4.2 Example of roles involved in release management activities

Activity	Responsible roles	Competency profile	Specific skills
Release planning process			
Product architecture and service relationship analysis	Enterprise architect	ATC	Knowledge of service relationship
	Service owner		Business analysis
	Product owner		Knowledge of service architecture
	Relationships manager		Knowledge of release and deployment methods
	Development team member		Expertise in infrastructure and platform
	Account manager		Communication skills
	Delivery manager		
	Designer		
Release management approach	Service owner	AMTC	Knowledge of service relationship
	Product owner		Knowledge of release and deployment methods

review and development	Relationships manager		Expertise in infrastructure and platform Communication skills
	Development team member		
	Account manager		
	Delivery manager		
Release management model review and development	Service owner	AMTC	Knowledge of service relationship
	Product owner		Knowledge of release and deployment methods
	Development team member		Expertise in infrastructure and platform Communication skills
	Account manager		
Release instance planning		TA	Expertise in infrastructure and platform Technical knowledge of service/product Service architecture knowledge Knowledge of release and deployment methods Administrative expertise in the service/product Knowledge of service relationship
	Service owner	C	Knowledge of service relationship
	Product owner		Communication skills
	Relationships manager		Marketing knowledge
	Account manager		
	Delivery manager		
Release coordination process			

Identification of applicable model or plan	Service owner	AT	Administrative expertise in the service/product
	Product owner		Service user experience
	Development team member		Expertise in infrastructure and platform
	Account manager		Knowledge of release and deployment methods
	Delivery manager		Technical knowledge of service/product
Verification of the service components	Service owner	TA	Expertise in infrastructure and platform
	Product owner		Knowledge of release and deployment methods
	Development team member		Technical knowledge of service/product
	Account manager		Administrative expertise in the service/product
	Delivery manager		
	Customer representative		
Verification of the release procedures	Development team member	TA	Expertise in infrastructure and platform
	Systems administrator		Technical knowledge of service/product
	Information security specialist		Administrative expertise in the service/product
Release execution	Development team member	T	Expertise in infrastructure and platform
	Systems administrator		Knowledge of release and deployment methods
	Information security specialist		Technical knowledge of service/product
Release verification	Service owner	A	Administrative expertise in the service/product
	Product owner		Service user experience
	Account manager		

	Delivery manager	Development team member	Customer representative	User
Release review	Service owner	ATC	Knowledge of service relationship	
	Product owner		Business analysis	
	Relationships manager		Knowledge of service architecture	
	Account manager		Knowledge of release and deployment methods	
	Delivery manager		Expertise in infrastructure and platform	
	Development team member		Technical knowledge of service/product	
	Designer		Communication skills	
	Customer representative		Marketing knowledge	
	User			

4.3 ORGANIZATIONAL STRUCTURES AND TEAMS

A designated release management team can only be established in large organizations with significant volumes and complexity of releases. In most cases, release management does not need a dedicated team; either these activities are highly automated, or a temporary project team is built.

However, the role of a release manager may still be relevant in many cases. This role acts as a coach to ensure the practice is adopted across the organization. Depending on the organization's approach to release management, this role may be combined with the role of deployment manager.

5 Information and technology

5.1 INFORMATION EXCHANGE

The effectiveness of release management is dependent on the quality of information used. This information includes, but is not limited to, information about:

- product architecture
- service consumer organizations and users
- software development and management practice
- planned and ongoing deployments
- ongoing and past incidents
- emerging release management techniques.

This information may take various forms. The detailed list of inputs and outputs of the practice are listed in section 3.

5.2 AUTOMATION AND TOOLING

Release management in a digital environment is highly automated. But even in legacy environment the work of the release management practice can significantly benefit from automation. Where this is possible and effective, it may involve the solutions outlined in Table 5.1.

Table 5.1 Automation solutions for release management activities

Activity	Means of Automation	Key Functionality	Impact on Practice
Release planning process			
Product architecture and service relationship analysis	Architecture tools	Visualization of the product/service architecture and relationships, connections, and constraints	Medium
	Business analysis and modelling tools		
	Product/service modelling tools		
Release management approach review and development	Process modelling tools	Modelling, visualization and assessment of the processes and procedures	Low
	Product/service modelling tools		
	Business analysis tools		
Release management model review and development	Process modelling tools	Modelling and visualization of the processes and procedures	Medium

Release instance planning	Process modelling tools	Automated release management	High
	Release and deployment management tools		
	Pipeline management tools		
	Software delivery and integration tools		
	Development environments		
Release plan communication	Social networks	Automated communications, messaging, status updates	High
	Portals		
	Knowledge base tools		
Release coordination process			
Identification of applicable model or plan	Process modelling tools	Modelling and visualization of the processes and procedures	Low
	Release and deployment management tools		
	Pipeline management tools		
Verification of the service components	Release and deployment management tools	Automated release management based on pre-planned, developed scripts	High
	Pipeline management tools		
	Software delivery and integration tools		
	Development environments		
Verification of the release procedures	Release and deployment management tools	Automated release management based on pre-planned, developed scripts	High
	Pipeline management tools		
	Software delivery and integration tools		
	Development environments		
Release execution	Release and deployment management tools	Automated release management based on pre-planned, developed scripts	High
	Pipeline management tools		
	Software delivery and integration tools		

Development environments			
Release verification	Release and deployment management tools	Automated release management based on pre-planned, developed scripts	High
	Pipeline management tools		
	Software delivery and integration tools		
Release review	Monitoring tools	Providing information and alerts	Medium
	Collaboration tools	Knowledge sharing	
	Communication tools	Communicating issues	

6 Partners and suppliers

Very few services are delivered using only an organization's own resources. Most, if not all, depend on other services. These are often provided by third parties outside the organization (see section 2.4 of *ITIL Foundation: ITIL 4 Edition* for a model of a service relationship).

As previously mentioned, the role of partners and suppliers is connected to the level of control an organization has over its product and services, or their components. When an organization controls a full product or service lifecycle, including development and deployment, it has more freedom in making a full range of decisions about release management. In contrast, if an organization's products or services are based on third-party components, or development and deployment are managed by a supplier, it usually introduces constraints that an organization must consider. It still may be able to decide whether to include updated components in its services, but only to a certain extent.

Organizations can sometimes outsource some aspects of the release management. For example, user communications, marketing the releases, user training, gathering feedback in hypothesis testing, and so on. Organizations' should properly manage those partners and suppliers activities, as they have direct impact on the users satisfaction and financial viability of the products and services.

Relationships between organizations may involve various levels of integration and formality. (see Table 3.1 of *ITIL® 4: Foundation edition* for more information about relationships between organizations). The level of integration with partners in the release management depends on forms of cooperation, which should be decided and managed through release management, supplier management, relationships management, service level management, and other related practices.

7 Important reminder

Most of the content of the practice guides should be taken as a suggestion of areas that an organization might consider when establishing and nurturing their own practices. The practice guides are catalogues of things that organizations might think about, and not a list of answers. When using the content of the ITIL Practice guides, organizations should always follow the ITIL guiding principles:

- focus on value
- start where you are
- progress iteratively with feedback
- collaborate and promote visibility
- think and work holistically
- keep it simple and practical
- optimize and automate.

More information on the guiding principles and their application can be found in section 4.3 of the *ITIL® 4: Foundation edition* publication.

8 Acknowledgments

AXELOS Ltd is grateful to everyone who has contributed to the development of this guidance. These practice guides incorporate an unprecedented level of enthusiasm and feedback from across the ITIL community. In particular, AXELOS would like to thank the following people.

8.1 AUTHORS

Anil Bahl
Martin Cross

8.2 REVIEWERS

Roman Jouravlev

