# ITIL – A guide to release and deployment management

## The goal of release and deployment management

Release and deployment management aims to build, test and deliver services to the customers specified by service design.

The goal of release and deployment management is to deploy releases into operation and establish effective use of the service in order to deliver value to the customer.

Release and deployment management also ensures handover to service operations takes place and that suitable training and documentation exists to ensure ongoing support of the new service.

## The purpose and objectives of release and deployment management

The purpose of release and deployment management is to:

- Define and agree release and deployment plans with customers/stakeholders
- Ensure that each release package consists of a set of related assets and service components that are compatible with each other
- Ensure that integrity of a release package and its constituent components is maintained throughout the transition activities and recorded accurately in the configuration management system
- Ensure that all release and deployment packages can be tracked, installed, tested, verified, and/or uninstalled or backed out, if appropriate
- Ensure that change is managed during the release and deployment activities
- Record and manage deviations, risks, issues related to the new or changed service, and take necessary corrective action
- Ensure that there is knowledge transfer to enable the customers and users to optimise their use of the service to support their business activities
- Ensure that skills and knowledge are transferred to operations and support staff to enable them to effectively and efficiently deliver, support and maintain the service, according to required warranties and service levels

## The scope of release and deployment management

The scope of release and deployment management includes the processes, systems and functions to package, build, test and deploy a release into operation.

The release process commences with receipt of an approved request for change to deploy a production ready release package. Deployment commences with receipt of an approved request for change to deploy a release package.

Value to business of release and deployment management

Effective release and deployment management adds value to the business/organisation by:

- Delivering change, faster and at optimum cost and minimized risk
- Assuring that customers and users can use the new or changed service in a way that supports the business and demonstrates the value outcome has been achieved
- Improving consistency in implementation approach
- Contributing to meeting auditable requirements for traceability through service transition
- Well planned and implemented release and deployment will make a significant difference to a business/organisation's service costs

# Release and deployment management concepts

## The release policy

The release policy covers release numbering, frequency and the level in the IT infrastructure that will be controlled by definable releases. The organisation should decide the most appropriate approach, depending on the size and nature of the systems, the number and frequency of releases required, and any special needs of the users – for example, if a phased rollout is required over an extended period of time. All releases should have a unique identifier that can be used by configuration management.

A release policy may say, for example, that only strict emergency fixes will be issued in between formally planned releases of enhancements and non-urgent corrections.

The term release is used to describe a collection of authorised changes to an IT service. A release is defined by the RFCs that it implements. The release will typically consist of a number of problem fixes and enhancements to the service. A release consists of the new or changed software required and any new or changed hardware needed to implement the approved changes. Releases are often divided into:

- **Major software releases** and hardware upgrades, normally containing large areas of new functionality, some of which may make intervening fixes to problems redundant. A major upgrade or release usually supersedes all preceding minor upgrades, releases and emergency fixes.
- **Minor software releases** and hardware upgrades, normally containing small enhancements and fixes, some of which may have already been issued as emergency fixes. A minor upgrade or release usually supersedes all preceding emergency fixes.
- **Emergency** software and hardware fixes, normally containing the corrections to a small number of known problems.

## Release unit

A *release unit* describes the portion of a service or IT infrastructure that is normally released together according to the organisation's release policy. The unit may vary, depending on the type(s) or item(s) of service asset or service component such as software and hardware.

The general aim is to decide the most appropriate release unit level for each service asset or component.

The following factors should be taken into account when deciding the appropriate level for release units:

- The ease and amount of change necessary to release and deploy a release unit
- The amount of resources and time needed to build, test, distribute and implement a release unit
- The complexity of interfaces between the proposed release unit and the rest of the services and IT infrastructure
- The storage available in the build, test, distribution and live environments

## Release identification

Releases should be uniquely identified according to a scheme defined in the release policy.

Releases should be uniquely identified according to a scheme defined in the release policy. The release identification should include a reference to the CI that it represents and a version number that will often have two or three parts.

Example release names are as follows:

- major releases: HR_System v.1, v2, v3 etc.
- minor releases: HR_System v.1.1, v.1.2, v.1.3 etc.
- emergency fix releases: HR_System v.1.1.1, v.1.1.2, v.1.1.3 etc.

## Release design options and considerations

Service design will define the approach to transitioning from the current service to the new or changed service.

Common options for release and deployment that are considered in service design are document below. The selected option will have a significant impact on the release and deployment resources as well as the business outcomes. It is important to understand the patterns of business activity (PBA) and types of users (user profiles) when planning and designing the releases.

## *Big bang* option

The new or changed service is deployed to all user areas in one operation.

## Phased approach

The service is deployed to a part of the user base initially, and then this operation is repeated for subsequent parts of the user base via a scheduled rollout plan.

## Push and pull

A push approach is used where the service component is deployed from the centre and pushed out to the target locations. In terms of service deployment, delivering updated service components to all users – either in big bang or phased form – constitutes *push*, since the new or changed service is delivered into the users' environment at a time not of their choosing.

A pull approach is used for software releases where the software is made available in a central location but users are free to pull the software down to their own location at a time of their choosing.

As some users will never *pull* a release it may be appropriate to allow a *pull* within a specified time limit and if this is exceeded a push will be forced, e.g. for an antivirus update.

## Automation vs manual

The mechanisms to release and deploy the correctly configured service components should be established in the release design phase and tested in the build and test stages.

Automation will help to ensure repeatability and consistency. The time required to provide a automated mechanism may not always be viable. If a manual mechanism is used it is important to monitor and measure the impact of many repeated manual activities as they are likely to be inefficient. Too many manual activities may slow down the release team and create resource or capacity issues that affect the service levels.

Many of the release and deployment activities are capable of a degree of automation. For example:

- Discovery tools aid release planning
- Discovery and installation software can check whether the required prerequisites and co-requisites are in place before installation of new or changed software components
- Automated builds can significantly reduce build and recovery times that in turn can resolve scheduling conflicts and delays
- Automated processes to load and update data to the configuration management system help to ensure the records are accurate and complete
- Installation procedures automatically update user and licence information in the configuration management system

# The release and deployment process activities

## Release and deployment plans

Plans for release and deployment will be linked into the overall service transition plan. The approach is to ensure an acceptable set of guidelines is in place for the release into production/operation.

Release and deployment plans should be authorised as part of the change management process.

The plan should define the:

- Scope and content of the release
- Risk assessment and risk profile for the release
- Customers/users affected by the release
- CAB members that approved the change request for the release and/or deployment
- Team who will be responsible for the release
- Delivery and deployment strategy
- Resources for the release and deployment

## Build and test prior to deployment into production/live environment

Build and test planning establishes the approach to building, testing and maintaining the controlled environments prior to production. The activities include:

- Developing build plans from the service design package, design specifications and environment configuration requirements
- Establishing the logistics, lead times and build times to set up the environments
- Testing the build and related procedures
- Scheduling the build and test activities
- Assigning resources, roles and responsibilities to perform key activities
- Preparing build and test environments
- Managing test databases and test data
- Software licence management

## Planning pilots

Pilots are useful for testing the service with a small part of the user base before rolling it out to the whole business/ organisation. It is important to determining the appropriate scope of a pilot (how much of the service is to be included in the pilot, size of department or user base).

The pilot should include steps to collect feedback on the effectiveness of the deployment plan. This can include:

Surveying views and satisfaction from:

- End users
- Customers
- Suppliers
- Service desk and other support staff
- Network management
- Data and knowledge management – statistics on use and effectiveness
- Analysing statistics from service desk calls, suppliers, capacity and availability

## Logistics and delivery planning

Once the overall deployment approach is understood, develop the logistics and delivery plans. These plans deal with aspects such as:

- How and when release units and service components will be delivered
- What the typical lead times are; what happens if there is a delay
- How to track progress of the delivery and obtain confirmation of delivery
- Metrics for monitoring and determining success of the release deployment effort

## Build and test of releases

Key aspects that need to be managed during the activities to build and test a service are:

- Usage of the build and test environments
- Recording the complete record of the build so that it can be rebuilt if required
- Maintaining evidence of testing, e.g. test results and test report
- Checking that security requirements are met
- Verification activities, e.g. prerequisites are met before a build or test begins

## Release and build documentation

Procedures, templates and guidance should be used to enable the release team to build an integrated release package efficiently and effectively.

Procedures and documents will be required for purchasing, distributing, installing, moving and controlling assets and components that are relevant to acquiring, building and testing a release.

## Release packaging

Build management procedures, methodologies, tools and checklists should be applied to ensure that the release package is built in a standard and controlled way in line with the solution design defined in the service design package. As a release package progresses towards production it may need to be rebuilt. For example, if a newer version needs to be incorporated quickly to fix errors; or if the documentation needs to be updated.

The key activities to build a release package are:

- Assemble and integrate the release components in a controlled manner
- Create the build and release documentation including: build, installation and test plans, procedures and scripts
- Monitor and check the quality of the release and how to recognise and react to problems
- The automated or manual processes and procedures required to distribute, deploy and install the release into the target environment (or remove it as necessary)
- Procedures to back out release units or remediate a change should a release fail
- Procedures for tracking and managing software licences
- Install and verify the release package
- Baseline the contents of the release package
- Send a notification to relevant parties that the release package is available for installation and use

If testing of a release package is successful, the release and the contents of the release package are placed under the control of configuration management, baselined and verified against the release design and release package definition.

From this point all changes to the release package are managed through change management, e.g. to fix an error in testing.

If at any step the testing of a release package does not complete successfully, reassessment and rescheduling of the release is managed through change management.

## Review and close a deployment

When reviewing a deployment the following activities should be included:

- Capture experiences and feedback on customer, user and service provider satisfaction with the deployment, e.g. through feedback surveys
- Review quality criteria that were not met
- Check that any actions, necessary fixes and changes are complete
- Review performance targets and achievements, including resource use and capacity such as user accesses, transactions and data volumes
- Make sure there are no capability, resource, capacity or performance issues at the end of the deployment
- Check that any problems, known errors and workarounds are documented and accepted by the customers/ business and/or suppliers
- Incident and problems caused by deployment
- Deployment is completed with a handover of the support for the deployment group or target environment to service operations
- A post implementation review of a deployment is conducted through change management