

ITIL® RELEASE AND DEPLOYMENT MANAGEMENT



[ITIL 4 Guide >](#)

ITIL release and deployment management

Your customers expect valuable services — and they expect them without disruption. That makes it critical that every single release be built, tested, and delivered following a rigorous process that ensures quality and minimizes risk.

ITIL's Release and Deployment Management phase is designed for exactly that: ensuring that your releases are deployed into production efficiently and effectively.

Download Now: ITIL 4 Best Practice e-Books

These all-new for 2020 ITIL e-books highlight important elements of ITIL 4 best practices. Quickly understand key changes and actionable concepts, written by ITIL 4 contributors.

[Free Download >](#)



[Free Download >](#)

According to ITIL, the purpose of the Release and Deployment Management process is:

To plan, schedule, and control the build, test, and deployment of releases, and to deliver new functionality required by the business while protecting the integrity of existing services.

But first — what's a release?

Simply put, a release (also called a release package) is a set of authorized changes to an IT service. That means a release can include hardware and software, documentation, processes, or other components that are essential to successfully implementing an approved change to your IT services.

Most ITIL-abiding organizations create **release policies**, which help to define how releases are numbered, how often they are released, and even how they are released (via a phased rollout versus a "big bang" rollout, for example.)

As part of your release policy, ITIL encourages creating a system for categorizing your releases. Categories typically include:

- **Major releases.** To qualify as a major release, it should contain new hardware or software. More often than not, a major release equates to introducing completely new functionality. Think v1.0 and v2.0-level releases — they're major milestones.
- **Minor releases** make significant improvements to existing functionality, often packaging together a number of fixes — and are often numbered v1.1, v1.2, v1.3 etc.
- **Emergency releases** are exactly what they sound like. Something bad needs attention ASAP, so you're releasing a temporary fix — and probably numbering the release something like 1.1.1, v1.1.2, v1.1.3, etc.

As you design your releases, you will have several options for how you plan to deploy, as well. Your release policies may or may not specify which approach to take, but regardless, you should pay close attention to the users you are deploying to and the business activities the release will impact to ensure you have the desired impact.

- **Big bang** releases are deployed to all users, all at once.
- **Phased** approach releases are more paced — deployed first to a subset of the broader user base, then deployed more gradually to additional users as part of a scheduled rollout plan.

Releases can also be pulled or pushed. A pull approach means the release is placed in a central location where users can download it at their own will, while a push approach implies that the release would be pushed out from a central location to each user.

Finally, ITIL suggests that you clearly specify whether the release will be deployed automatically (i.e. using installation software, etc.) or manually.

Objectives of release and deployment management

According to ITIL, the objectives of Release and Deployment Management are:

- *To create, test, verify, and deploy release packages*
- *To manage organization and stakeholder change*
- *To ensure that new or changed services are capable of delivering the agreed utility and warranty*
- *To record and manage deviations, risks, and issues related to the new or changed service and take necessary corrective action*
- *To ensure there is knowledge transfer to enable customers and users to optimize use of services that support their business activities*

Release and deployment management scope

The scope of Release and Deployment Management includes all Configuration Items (CIs) that are required to implement a release, including:

- Virtual and physical assets
- Applications and software
- Training for staff and users
- All related contracts and agreements

Testing that is carried out as part of the Service Validation Process is not considered in scope, and neither is authorizing changes.

The key phases to release and deployment management

Release and Deployment Management is divided into four distinct phases, beginning with thoroughly planning what you will release and how, and concluding after the release is deployed and a post-mortem review is conducted.

Phase 1: Release and deployment planning

A well-thought-out release and deployment plan is just one component of your overall Service Transition plan — but the point is to clearly define a set of guidelines for both what a release will include and how you will deploy it into production. The release and deployment plan is then approved as part of the Change Management process.

At the beginning of the release and deployment-planning phase, change management typically authorizes the planning process to begin for a release. The plan typically addresses:

1. What changes the release will include
2. Who will be affected or impacted by the release
3. What risk the release may introduce, if any
4. The audience for the release (i.e. what users, customers, organizations will be impacted)
5. A clear chain of approval, clarifying which stakeholders may authorize the change request at

every stage of the release

6. Ownership, defining the team who is ultimately responsible for the release.
7. Deployment schedule and strategy

It's quite common for building and test plans to also be built in this stage, clearly outlining critical logistics like design specifications, testing procedures, timelines for building and testing, and even pass / fail criteria at each phase of deployment. A pilot rollout may also optionally be planned at this time.

Phase 2: Release building and testing

Once a plan is in place and approved by change management, the responsible teams have to build and test the release, including both the software, documentation, and any other elements the release plan specifies.

At the beginning of this process, documentation is typically created to ensure that developers will be able to build the release package as accurately and efficiently as possible — and throughout the build process, accurate records should be kept so the build process can be repeated, if it becomes necessary.

Most organizations typically follow stringent procedures, or even provide standard templates for building a complete release package. Ensure you are utilizing and following these at every step of the way.

Testing happens throughout the process — from testing any and all input CIs, to testing and rehearsing the services before they are deployed live.

A few things to remember along the way:

- Pilots are a great way to identify and correct any issues with a service before they are deployed to the entire intended audience. This can dramatically reduce risk.
- Some teams also choose to stage a rehearsal, effectively "practicing" as much of a service rollout as possible shortly before a deployment is scheduled.

Phase 3: Deployment

In this phase, the release package is deployed to the live environment, beginning when change management authorizes the release package to be deployed to the target environments. The deployment phase ends with handoff to service operations and early-life support.

Before deploying, ITIL encourages quite a bit of advanced planning and preparation — confirming the target group is ready for the deployment, identifying and attempting to mitigate any potential risks or disruptions, and specifying the order of how each component of the release will be deployed (like financial assets, processes and materials, the actual service release, etc.)

Once a release is deployed, it's critical that you verify that it is operating properly for all stakeholders — and remediate or back out the release as needed should serious problems arise.

After verifying that the release is functioning as planned, ITIL calls for you to transition the new or changed service over to service operations in two stages. First, a formal notification should be issued that the service is now live (at the beginning of early life support or ELS), followed eventually by a formal notification that the service is fully operational and SLA's are being fully enforced.

Phase 4: Reviewing and closing a deployment

Once the release is deployed, it's time to review and learn from the entire process. Feedback is gathered, and evaluations are performed against performance goals — with the results being reviewed and discussed by all involved.

Reviews should be careful and thorough, confirming that all quality requirements have been met, that sufficient knowledge transfer and training were performed, and that any known errors, fixes, and changes have been adequately documented. Additionally, change management should conduct a full Post Implementation Review, or PIR.

A deployment isn't considered "closed" or completed until support has been formally transitioned to Operations.