



WHITE  
PAPER



RELEASE MANAGEMENT: PART 2

# What is Release Management?

- Definition of the Release Manager role.
- How to align the business and IT using Release Management.
- Creating a repeatable Release Management system that scales using modern tools.
- The role of Release Management in Agile, DevOps, and enterprise digital transformation.

Release Management is the process that handles software deployments and change initiatives.

Across an organization, it schedules the relevant tasks (internal and external), assigns the physical and human resources needed to carry them out, and oversees the execution. It starts with planning what will be contained within a release, managing the software build through different stages and environments, testing stability and finally, deployment.

It is a difficult balancing act between:

- Delivering **value and novel features** to customers
- **Prioritizing value streams** according to business needs
- Improving the **speed and quality** of release deployments
- Not compromising **product stability**

No matter the software development approach, the Release Manager has the skill set, initiative and determination to make this happen.

## What exactly does a Release Manager do?

1. Understands the business needs and their priorities, and under what circumstances those priorities can change.
2. Works with business leaders, product owners, IT project teams, and operations staff to ensure every release contains the correct features.
3. Changes the release scope or re-prioritize release features according to information from project and portfolio managers.
4. Has a clear picture of development dependencies, and how changes to one part of a product can affect the stability of the whole.
5. Understands the bandwidth and work capacity of each team involved in development.
6. Understands the availability of resources and environments for testing.
7. Schedules builds and testing according to team and resource bandwidth and availability.

8. Create release plans, including governance and approval requirements.
9. Ensure compliance of new releases with governance requirements.
10. Optimizes value creation at every step, from feature check-in to deployment.
11. Schedules seamless release deployments.

A release manager needs to cross departments and disciplines in order to effectively prioritize business needs within an application's release. They need to be both generalists and specialists – the T-shaped skillset.

Release managers need a wide range of general skills – they must be able to understand the software development process, liaison with the business, synthesize data and be well versed in both products and projects. They must also have highly specialized knowledge of management skills, strategic decision-making and how to coordinate implementation.

A lot of the day-to-day work of a release manager may look like project management, and release managers often work closely with project or portfolio managers. However, release management is distinct from the project management office (PMO), product management, or change advisory board (CAB).

Operations Manager	Product Manager	Release Manager	DevOps Manager	CIO
Develop policies & procedures	Understand business needs	Provide reporting	Improve collaboration	Navigate tech trends
Monitor IT systems	Define scope of project	Adjust release scope	Testing automation	Set strategy
Direct IT technicians	Make activity schedules	Create release plans	Implement CI/CD	Customer engagement
Resolve help desk escalations	Risk & time management	Ensure compliance	Find & address bottlenecks	Establish partnerships

Breadth / General Knowledge

Depth / Specialized Knowledge

**A release manager needs to cross departments and disciplines** in order to effectively prioritize business needs within an application's release. Often, release managers are IT professionals specializing in coordinating release activities or product owners that have upskilled their development knowledge to be able to manage releases.

## What is the role of Release Management?

Companies in the midst of digital transformation can lose sight of the importance of release management. Agile, DevOps, continuous distribution – surely we can automate away the need for release management? However, accelerating the development of software alone does not give an enterprise a competitive advantage.

An automated delivery pipeline and autonomous product teams do not remove the need for end-to-end alignment and a systems-based approach to development and delivery efficiency. Without effective release management, organizations are at risk of wasteful development pipelines, underutilized resources, and inadvertently prioritizing delivery of low-impact features. The release manager marries the outside (customer and business value) and inside (user stories) perspective, ensuring the

efficient prioritization of work in every sprint.

## Delivering value to customers

It's easy for a company to focus on the speed of software development as key to digital transformation. However, digital transformation success relies on the radical focus of increasing value delivery to customers. Release management teams understand the business needs resulting from customer feedback, and can translate that into actionable development plans.

## Scalable risk management

DevOps leads to faster time to market, but does not eliminate the business risks associated with product development. Faster deployment to production environments must be supported by rigorous risk awareness and risk management. Release managers are ideally placed to audit the process and standardize governance policies

and requirements. Only a standardized, repeatable risk management process can be scaled to the enterprise level.

## Improving deployment efficiency

The efficiency of a deployment can be graded with three metrics. With the same resources:

- The customer must receive more positive value (novel features and fixes).
- The customer must receive less negative value (bugs and downtime).
- The customer must receive the net value faster.

Effective release management aims to improve all three of these metrics.

## Release Management Process Components

**Software Delivery Value Stream:** The processes that add or create software value across the release pipeline.

**Release Pipeline:** A specific release process from feature planning to delivery.

**Release Policy:** The definition of release types, standards, governance requirements for an organization.

**Release Template:** A single, repeatable workflow process for release pipeline that includes human and automated activities and follows an organization's release policies.

**Release Plan:** An instance of a release template developed for a specific release.

**Deployment Plan:** Activities to deploy a release to the production environment.

**Release unit:** The set of artifacts released together to implement a specific feature.

**Release package:** A combination of one or more release units deployed together as a single release due to interdependencies, scheduling, or business priorities.

**Major Releases:** Infrequent release packages that often include many release units that have a high or critical business impact.

**Minor Releases:** More frequent release packages with fewer release units that do not include mission critical components.

## The Release Management Process

The steps of the release management process will be similar regardless of company size, but there will be many differences in the details.

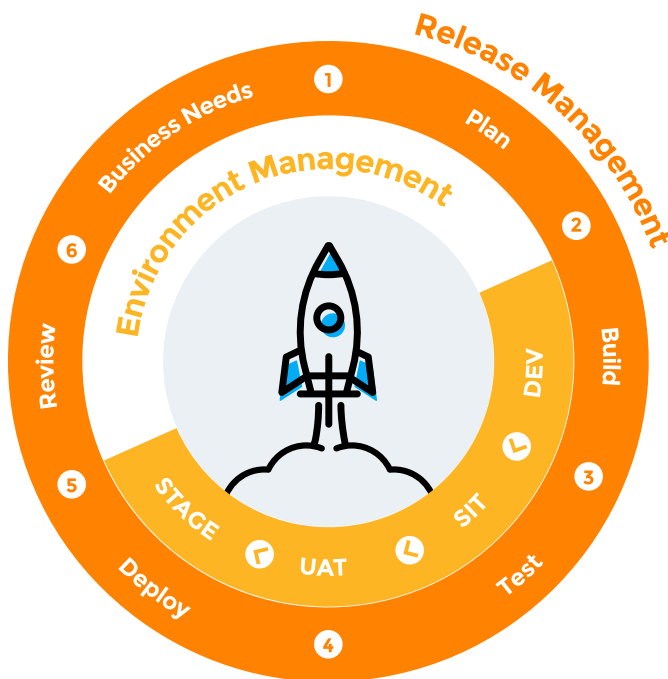
A small company's releases may consist of single applications and value stream. This is relatively simple to plan, build, test, deploy and review. Release managers can keep track of the release status and any scheduling using simple tools and applications.

As company complexity grows, its releases grow commensurably in size and scope. A large enterprise release can have multiple interdependent value streams. Scheduling must be coordinated between dozens of people, resources and testing environments without impacting other concurrent releases. Governance requirements increase and are more complicated to comply with. Coordination is essential and simple tools like Excel and Sharepoint can't keep up with the new pace.

## Creating a Release Management System

There are two components of the release management process. One is setting up a release management system, and the other is following through with that system. The most basic way to manage releases relies on a checklist. This can be enough for very small teams and simple applications but does not scale with increased release complexity.

Effective release management relies on a repeatable system that is difficult or impossible to circumvent. It simply can't be done manually at the enterprise level. As a business scales, release management must scale too. Enterprise-class software development requires enterprise-class tools.



Once the right tools and systems are in place, the release management process should be second-nature, and integrated into the application lifecycle.

Scaling the release management process means using tooling to:

- Enforce your checklist
- Standardize procedures across the organization
- Get the visibility needed to understand dependencies.

As you create a system, here are the steps to include:

**Clarify business needs.** The very first step in any release should be a clear user story that outlines the benefits to end-users. If this is not clear, it's impossible to tell if the feature meets its stated goals or if it needs further iteration.

**Release planning.** Each feature or application should go through a release planning phase before development begins. This involves finalizing the correct scope for the release and identifying potential problems before they can cause delays or issues.

At this stage, there should be a clear understanding of exactly which features / user stories will be included in the release. The release manager must also understand the upstream and downstream dependencies related to each feature, and how the changes will interact with other components of the application suite. The release risk profile, from compliance/ security risk to downtime risk, should also be evaluated at this time.

**Build/test/deploy.** As software is being built, tested and ultimately deployed, developers and operators need access to the dependencies and potential risks identified during the release planning phase. This allows them to test accordingly, and to know where to look for potential problems before promoting the release to full production.

**Plan for the future.** A mature release management system is circular, and involves constant iterations for continual feedback and improvement. As soon as a release is finished and determined to be a success, it's time to think about how it could be improved on in future releases. Is it meeting the business needs identified at the beginning? Does user adoption meet

expectations? For this stage, having access to data collected throughout the release process makes continual improvement possible.

**Tools make the release management process transparent.**

The software development lifecycle starts with the business need and ends with end-user feedback and adoption. Effective release management at scale relies on tooling to provide end-to-end visibility for the entire lifecycle. Release managers need a way to get up to the 10,000 foot view to see the end-to-end process and identify bottlenecks, see how seemingly unrelated teams and applications work together and head-off potential collisions.

Release data is often trapped in data silos and disparate applications throughout the enterprise. Release managers and team members struggle with updating spreadsheets and the administrative burden of collating this data.

Tools pull together this information for whole-enterprise visibility – dependencies, sources of potential risk and the interaction

information needed to manage releases effectively. Without this visibility, creating the release plan and using that information throughout the build/test/deploy phase is impossible.

Tools can also be used to enforce release management policies—for example, to make it impossible to deploy an application if the system detects an unresolved dependency or a violated governance policy. At scale, this is essential for ensuring that the established release management process is followed.





## Release checklist concepts

Foundationally, a release aims to realize a target state for an application or set of applications from feature acceptance to delivery. The release strategy starts with determining the success metrics that matter most to the business.

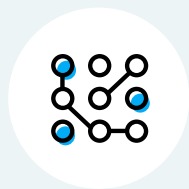
**Next you need to determine how will you measure these metrics throughout the release process, this includes considering:**

- What type of release cadence have you committed to and what can you truly support?
- Can you measure and report on plan versus actual?
- What about measuring deployment cycle times?
- Establish a common vocabulary of terms and concepts.
- Establish a release scheduling process
- Establish guideline for release scoping
- Establish governance processes and compliance auditing procedures

## What is the role of Release Management?



Pre-defined metrics for a successful release



Leading indicators of release risks including using a release impact matrix and environment impact matrix



What is the usage of the build and test environments including CI/CD pipelines



Reviewing testing results and auditing issues with traceable data



Verifying governance compliance

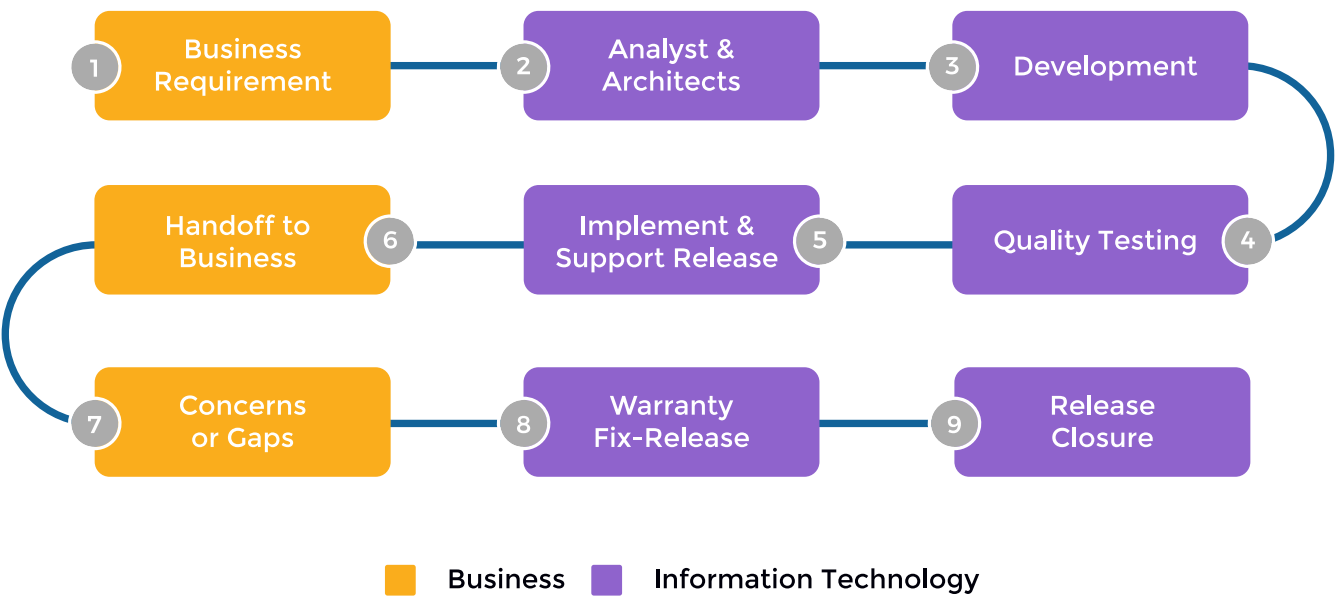
# Release Management in waterfall

Waterfall methodologies work best in situations where software product requirements can be well-defined up-front with a high degree of certainty. Here release management coordinates with the business side to define the release needs, and coordinates with IT to decide what to prioritize according to resource availability.

The release manager is responsible for creating and executing the release plan. The release package goes through testing, acceptance by the QA team, then any other stakeholder approvals as defined by the Release Policy.

It is then ready for deployment in the Production environment where end-users or customers will be able to access the new capabilities released.

## BUSINESS & IT ALIGNMENT



## Release Management in Agile and DevOps

In Agile methodologies, agile release trains align to value streams to deploy releases. Release units are delivered every 2-week sprint and release packages are deployed to production every 10-12 weeks. The scaled agile framework (SAFe) is one such agile methodology designed for large enterprises.

As an enterprise adopts DevOps, the release manager also coordinates with environment managers for testing and deployment planning for release packages. The automation and decentralization of DevOps teams may at first appear to make release management in the SDLC redundant or unnecessary.

**Release management is just as essential in DevOps teams as any other.**

New ways of building and releasing software means that new releases can be deployed multiple times a day. But software delivery still requires

the participation, collaboration, and engagement of multiple participants. Without coordination and alignment with business priorities, these faster systems risk flooding uninterested users with low-quality software.

Release management is responsible for coordinating with the DevOps manager to enact and monitor the continuous integration and delivery in the DevOps pipeline. Lastly, the release train interfaces with service management to address identified issues. These features, just like any other, must be integrated into a properly scoped release by the release manager.

As DevOps pipelines become more automated, a release managers daily coordination activities diminish. At this point in a digital transformation, the product owner often takes on the role of release manager, acting as a bridge between business, product, development and operations. Additionally, a centralized release management center of excellence can work with product owners to ensure that organization release policies are met and spread best practices across teams.



## Want to learn more about release management?

Check our series of white papers about Release Management to learn everything you need to know.

1. What is a Software Release?
2. What is Release Management
3. The Benefits of Release Management
4. Release Management Techniques for Process Improvements

Visit [www.plutora.com/software-release-management](http://www.plutora.com/software-release-management) to learn more.

---

## About Plutora

Plutora, the market leader of value stream management solutions for enterprise IT, improves the speed and quality of software creation by capturing, visualizing and analyzing critical indicators of every aspect of the delivery process. Plutora orchestrates release pipelines across a diverse ecosystem of development methodologies, manages hybrid test environments, correlates data from existing toolchains, and incorporates test metrics gathered at every step. The Plutora

Platform ensures organizational alignment of software development with business strategy and provides visibility, analytics and a system of insights into the entire value stream, guiding continuous improvement through the measured outcomes of each effort.



Learn more: [www.plutora.com](http://www.plutora.com)

Email: [contact@plutora.com](mailto:contact@plutora.com)