

WHITE
PAPER



RELEASE MANAGEMENT: PART 1

What is a Release in Software?

- How the way that customers access software has evolved.
- New distribution channels are leading to dramatic innovation and industry disruption.
- The role of Releases and clearing up common misconceptions about it.
- The future of the Release process.

A release is the distribution of software to the consumer. From floppy disks to CDs, one-off downloads to continuous delivery, software distribution speed has accelerated exponentially.

How customers access software: then and now

Distribution method	New release timeline	Type	Characteristics
Physical item (CDs, floppy disks, hardware model)	Years	Waterfall	<ul style="list-style-type: none"> • Could not be update, modify, or monitor. • Major updates in every release to justify pricing. • Must work perfectly first time. • Planned far ahead of time.
Software downloads (Internet)	Months - Years	Waterfall - Agile	<ul style="list-style-type: none"> • Increasing internet speeds allowed for software downloads. • Problems and bugs can be patched. • Focus on incremental improvement. • Focus moves from project to value stream. • Customer feedback becomes basis for next iteration.
SaaS (Cloud storage)	Months - Weeks	Agile - DevOps	<ul style="list-style-type: none"> • Constant responsiveness to customer feedback. • Monthly subscription income funds further development. • Cloud resources mean that companies can take advantage of new methods without investing in their own hardware. • Increased use of automation. • Instant software releases. • Continuous Improvement as a foundational element of transformation

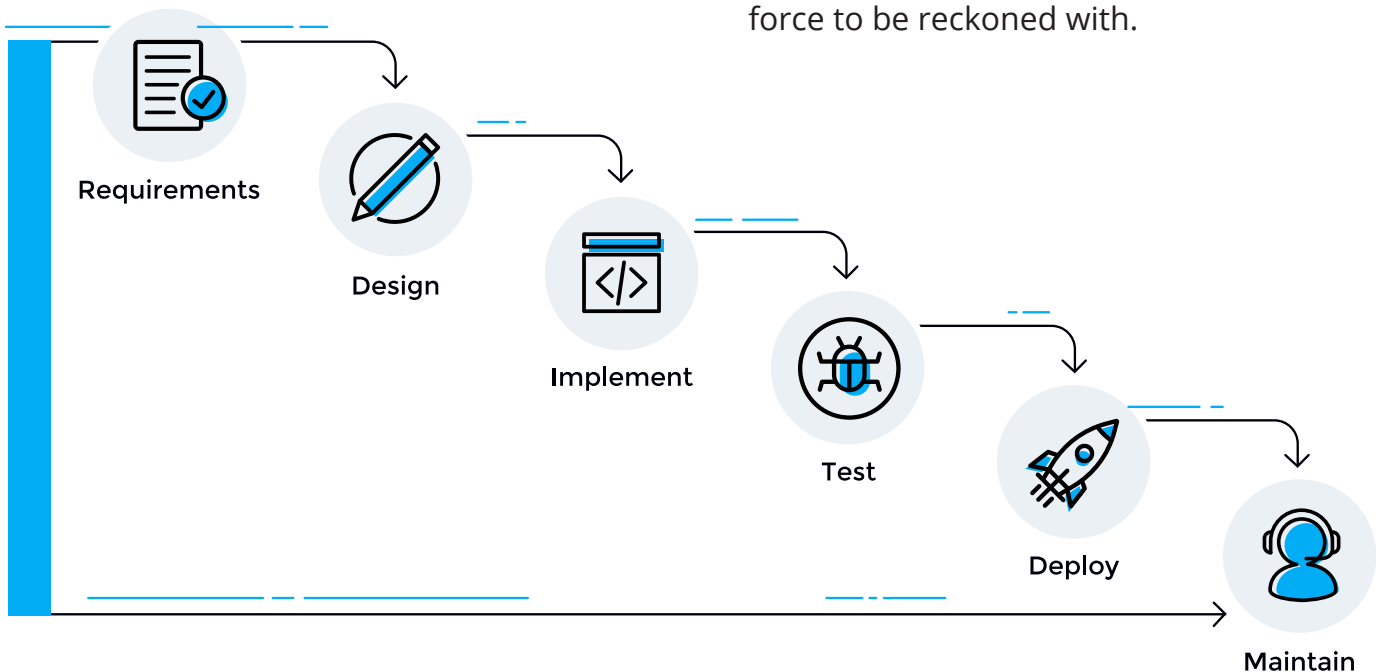
New distribution channels enable development innovations

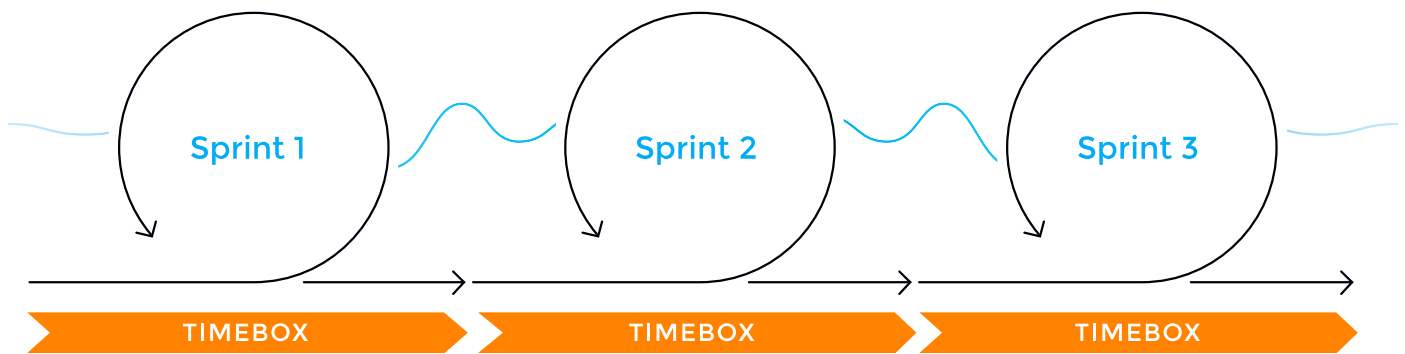
Distribution is fundamental to every market and its constraints determine viable production methods. Changing these constraints creates production opportunities that lead to dramatic innovation and industry disruption.

Consider how the shift from local cottage industries to the mass production line changed how we manufacture products. Software development is no different. Changes in distribution have changed how we approach the software development lifecycle.

Waterfall, based on the traditional project management methods that work for construction and manufacturing, was naturally the first method used for software development. A project is completed step-by-step in distinct stages. Each stage must be completed before the next stage can begin. Progress cascades down the stages, giving the methodology its name.

Waterfall methodologies work best in situations where software product requirements can be well-defined upfront with a high degree of certainty. This stage by stage approach works well for a building or a bridge – you can't go back and optimize the foundation once the windows are in. But developers in the 90s were already chafing at the constraints of the methodology as the internet became a force to be reckoned with.

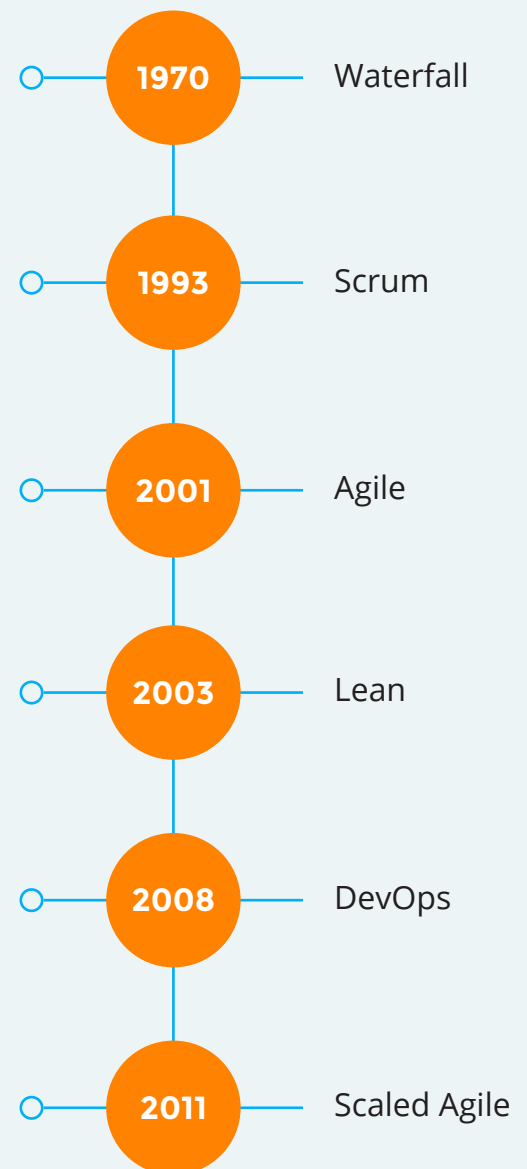




Agile emerged symbiotically with the large-scale adoption of the internet. The world wide web fundamentally changed how software was delivered and how customer feedback was received. Customers no longer had to wait a year or more for a software CD – they could download it.

Users could give their feedback to developers as soon as they tried out the software. Patches, fixes and updates could be sent out in a matter of weeks or months. It made more sense to code and test incrementally, where bugs and issues could be spotted well before deployment.

The speed of software distribution took a huge leap forward, and waterfall methods found it hard to compete with this emerging nimble, agile delivery method.



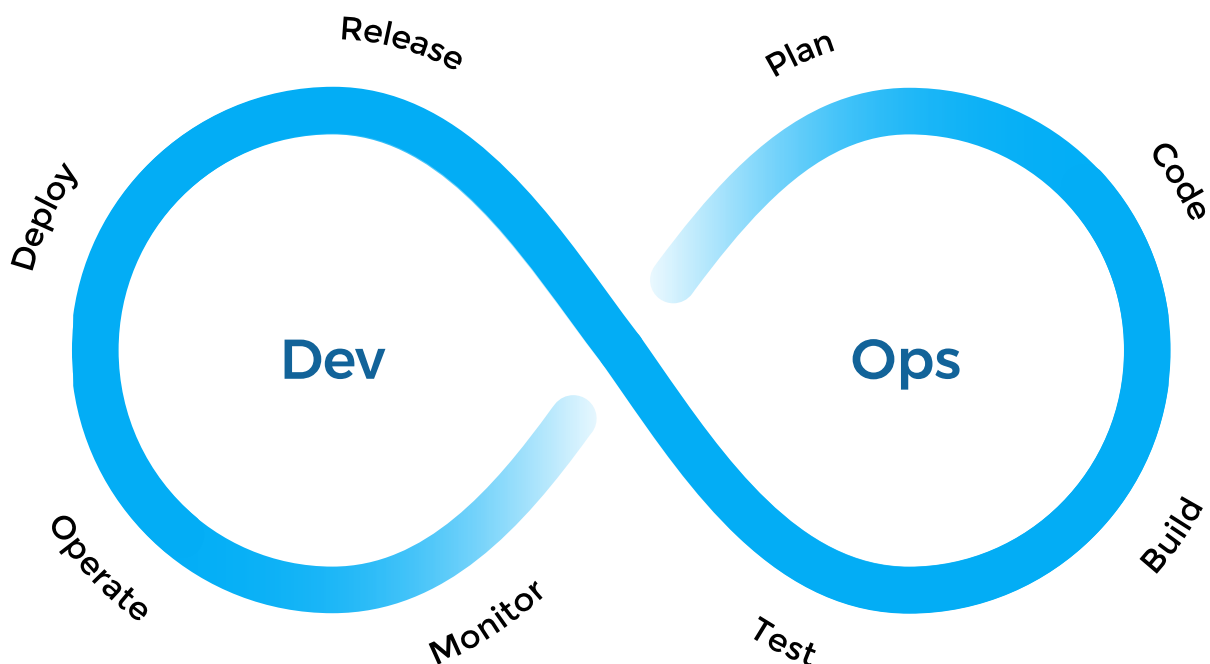
DevOps emerged in the wake of the next big step of the digital transformation – the shift towards cloud computing. Cloud tools and services made provisioning infrastructure faster than ever. Developers can quickly try new things without waiting for IT operations to provision services to them. The silos between Development and Operations began to break down as velocity between the two areas increased and close collaboration became essential. Cloud computing enabled DevOps teams to automate more and more of the process of building, managing, provisioning and deploying.

If software was still released annually via CD, DevOps would likely not be the standard development methodology. There's simply less value in implementing CI/CD pipelines if customers won't receive software updates for a year.

While how software is “released” has dramatically changed, a release remains the distribution of software to the consumer. Software distribution infrastructure will continue to change and in turn lead to more innovations in software development life cycles (SDLC).

No matter the SDLC, every organization needs a process to align the features of a release with:

- The business's priorities
- The infrastructure the software will use
- How and when customers will be able to access it.



Common release misconceptions

Software is released in multiple ways, and SDLCs cover a variety of release processes. Between all the moving parts, new technologies, and processes, it's easy for the role of release management to be misinterpreted, overlooked and minimized.

Release management is not a core IT process

- Release management is an essential component of modern software development, aligning business needs with IT work. As many day-to-day release manager activities can be automated, release management can be perceived as no longer critical and replaced by product management.
- Product managers, unlike release managers, may not have the experience in software development to effectively scope, schedule and manage the development process.
- Release management keeps your software running smoothly during updates and minimizes bugs and fixes – it is essential to continued successful deployment.

Project management, release management and product teams perform the same function

- All of these roles bring together needs of multiple departments to coordinate activity execution. That's why they can be perceived as doing the same role. And they certainly have overlap.
- Project management teams do not create strategy. They ensure a plan is executed. PMO teams ensure that work is prioritized, communicated, and tracked. However, while they can communicate the business priorities they do not work with the software development and delivery process to determine the best way to implement business priorities.
- Product managers leverage a T-shaped skillset to develop product strategy from business needs in the form of user stories. They are close to IT teams and keep an understanding of where their features are in development. However, unless they are upskilled to release management practices they may not be able to accurately manage release needs.

- Release managers create the software development strategy for product requirements/user stories. They coordinate between business, product teams and IT, scheduling and optimizing release packages to ensure successful delivery. Product managers who upskill often take on release management responsibilities.

Release management is not part of DevOps

- With DevOps, many release management activities have been automated. This leads to companies believing that a release manager is no longer necessary – incorrect. Scheduling, scoping, and planning releases is not trivial and even with automation requires a release management with deep software development knowledge.

Without a release management process, there is no way to ensure organization-wide governance compliance, efficient scheduling of shared resources, and effective use of team members resources and skills.

The future of the release process

As data silos break down and collaboration tools accelerate, the role of release manager and PMO will change. However, as new technologies and organizational structure become mainstream, there will continue to be a need for a release manager to understand both the priority of business needs, development dependencies, and work bandwidth to ensure successful deployments.





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 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.

PLUTORA®

Learn more: www.plutora.com

Email: contact@plutora.com