WHITE PAPER

RELEASE MANAGEMENT: PART 7

Release Management, DevOps, and Agile

- Examine how release management fits into DevOps, Agile, ITSM, Waterfall, SAFe, and other software development methodologies.
- Shorten the time between planning and delivering a release using DevOps practices
- Identify waste and measure improvements as you continually streamlining your release processes.

PLUTORA

DevOps (an extension of agile) promises to balance throughput and stability so we can deliver more value faster and more safely. Along with cloud, it's the key enabler for digital transformation.

Start with two key principles:

1. Release weekends are bad
2. Releases should be 'like breathing'

Much of the pain associated with releases in traditionally operating environments has been caused by the separation of the Development and IT Operations teams.

What is often referred to as a 'wall of confusion' exists between these two silos which particularly becomes apparent when the development team 'throws new code over the wall' where it is caught by the operations team who now have to perform the release with little knowledge of its provenance.

When things go wrong:

- Failure can be catastrophic
- It's extremely stressful for everyone involved
- It can be very expensive – not just to fix but in terms of fines and reputational damage
- It can cost people their jobs and progress in their careers
- It creates immense bad feelings between teams in contention as a 'blame-game' ensues

That's why release management tools, like Plutora, are so critical to supporting teams practicing DevOps. They make the releases visible, automate the workflow between teams and mitigate the risk of failure.

Before diving into the relationship between release management, DevOps, and Agile, let's also examine how it fits into ITSM, Waterfall, SAFe, and other software development methodologies.

Release Management and ITSM

ITSM defines release and deployment management as the process of managing planning and scheduling the rollout of IT services, updates, and releases to the production environment.

The idea of a new software release was created when a new business case or a project was created by, for example, by the marketing team. Traditional ITSM processes create release packages; large bundles of features.

The hands-on work of Release Management has traditionally been the

domain of those managing the IT services, the IT Operations department. It was the release process where much of the conflict happened that triggered the DevOps movement. This was because the development team created the release packages which they would then "throw over the wall (of confusion)" to the IT operations team to deploy.

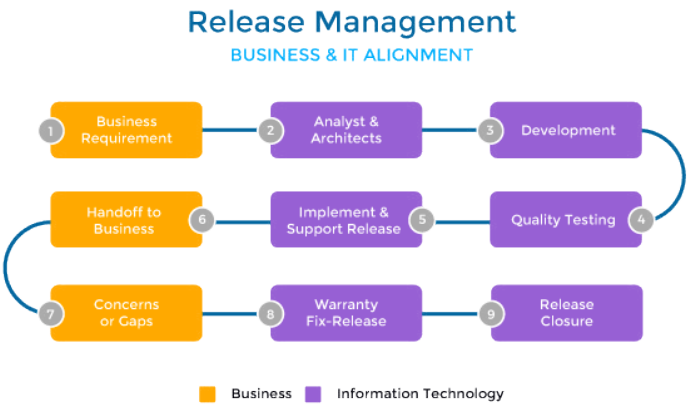
Problems would then occur as a result of a number of factors:

- The IT operations team did not understand the release package
- Inconsistency between pre-production and production environments
- Poor quality code, with so many parts it was difficult to pinpoint the cause
- Integration issues with complex dependencies in tightly coupled systems
- The development team increased release cadence when IT operations were not ready
- The conflict that arose from these painful release events manifested in several ways:
- Outages that led to 'war-rooms' where individuals and teams blamed each other without evidence
- Development teams complaining IT operations took too long to provision environments, or make test environments available
- IT operations becoming frustrated with developers asking for administrator access to production systems which was in contravention of compliance requirements (segregation of duties)

This conflict continues to prevail in many organizations, but effective release management along with DevOps and cloud practices supported by tools like Plutora can resolve these problems.

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. 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.



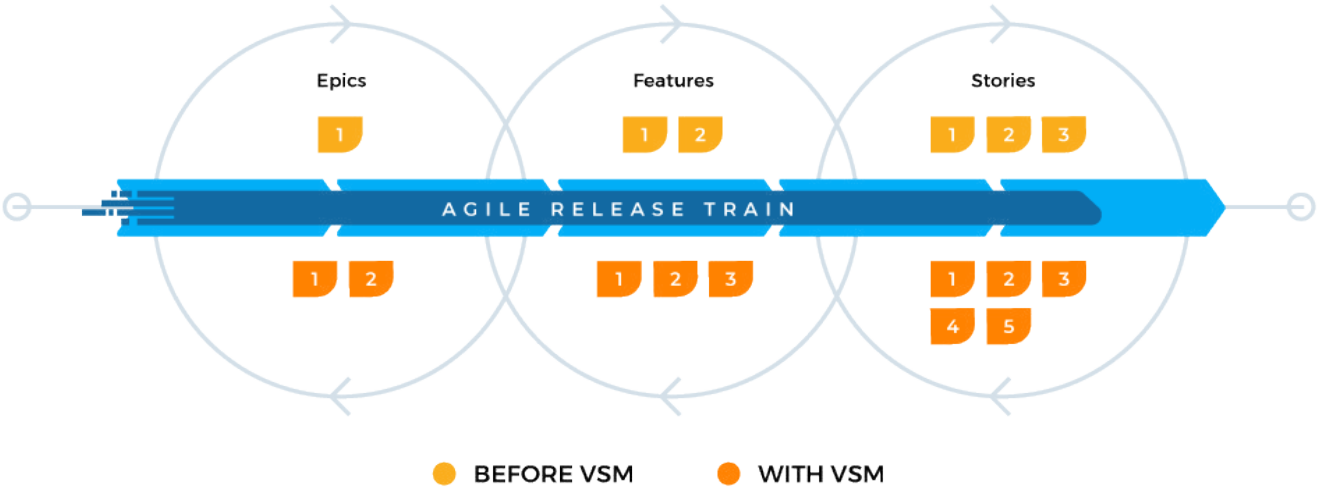
The release manager is responsible for creating and executing a release plan. The release package goes through testing, acceptance by the QA team, then any other stakeholder approvals as defined by the Release Policy. It is then ready for deployment in the Production environment where end-users or customers will be able to access the new capabilities released.

Release Management and SAFe

In Agile methodologies, Agile Release Trains (ARTs) align to value streams to deploy releases. Teams and ARTs develop in set, timeboxed iterations but are free to release when appropriate for the customer or market demand, and regulatory needs allow. When larger pieces of work, with more teams are required, a Program Increment (PI) is used to coordinate the effort and manage dependencies. PI planning is said to be the most significant cadence-based event for an enterprise, when the ART aligns to their goals.

It recognizes that in order to be able to release on demand, DevOps competencies must be embraced. A small batch, iterative approach replaces the traditional big-batch waterfall approach driving continuous feedback, delivery and compliance. Project plans are replaced with product backlogs and roadmaps. Epics are decomposed into capabilities and features when the work nears execution readiness. There is a focus on loosely coupled architectures to drive team autonomy and allow for small increment release.

SAFe recognizes that in some enterprise scenarios, true continuous integration, where every developer merges code into trunk daily (and automated tests are run) is impossible. There may be dependencies for pieces of code that simply cannot be ready or that are being worked on outside of the organization. So SAFe's 'continuish' integration approach aims for frequent partial integration with at least one full product integration per Program Increment (PI).



Release Management in Agile+DevOps

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.

That's why release management is just as essential in DevOps environments 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.

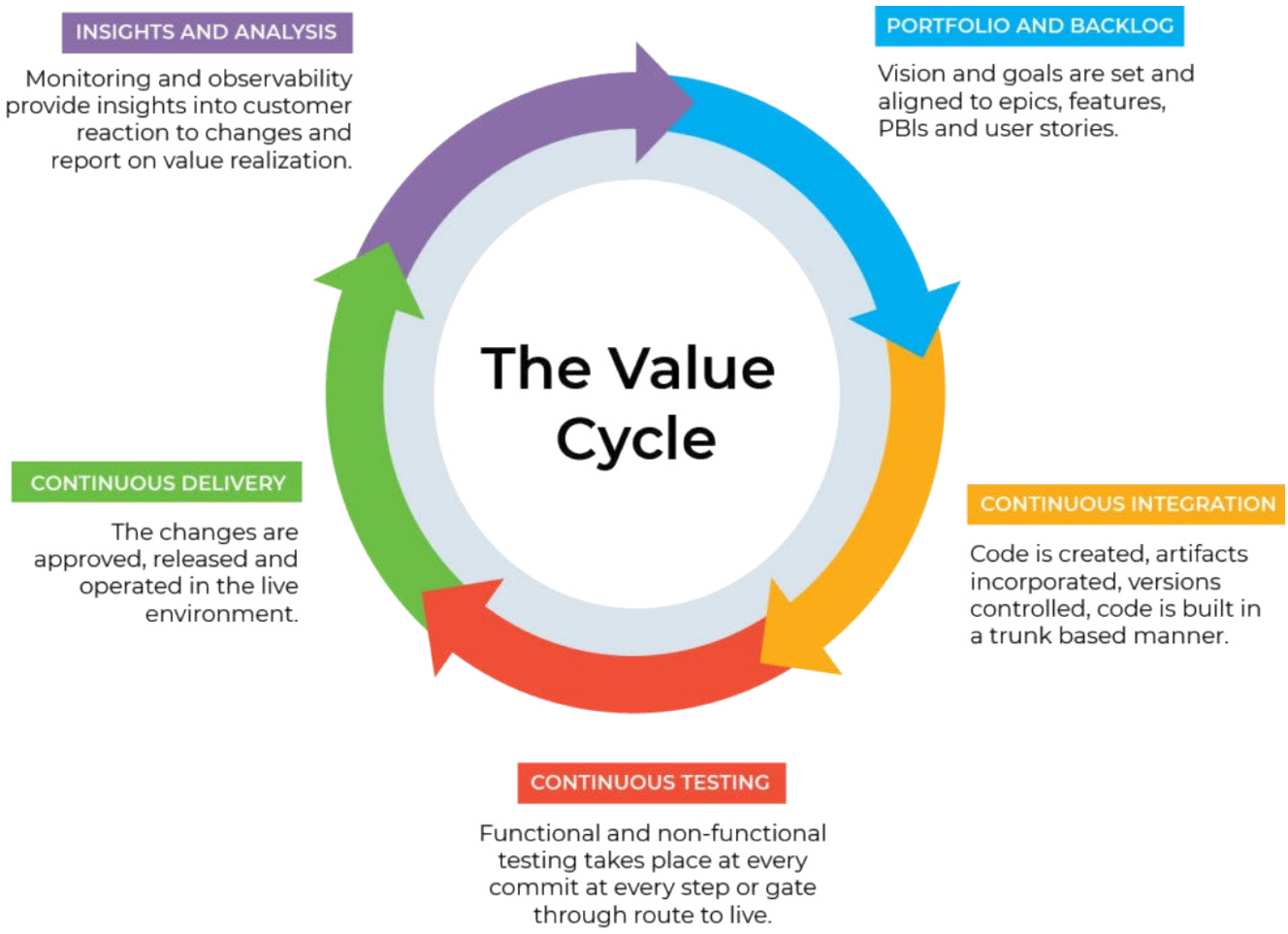
As DevOps pipelines become more automated, a release manager's daily coordination activities diminish. At this point in a digital transformation, the product owner often takes on the role of release manager, acting as a bridge between business, product, development and operations.

Additionally, a centralized release management team can work with product owners to ensure that organization release policies are met and spread best practices across teams.

Release Management, CICD, and the DevOps Toolchain

The release process is a step in the value stream. The value stream is the end-to-end lifecycle of work from idea conception to realization of the intended value in the hands of the customer.

The DevOps toolchain automates each step of the value stream. It's an extension of Continuous Integration and Continuous Delivery (CICD) which focus on keeping software always in a releasable state by encouraging developers to commit code at least daily to trunk where an automated build and tests occur.



This practice reduces the risk of release as it ensures quality upfront and works in much smaller increments.

CI can also be performed in waterfall projects and it does still have benefits in terms of avoiding “merge hell” but teams won’t benefit from the fast feedback loops in small batch, agile ways of working.

What is Progressive Release Management?

Progressive release management helps organizations scale and de-risk their release processes as they transition from traditional, project-based ways of working to current agile and DevOps approaches.

This is also part of Value Stream Management, a practice that improves the flow of work from idea to realization by showing the progress of work across an organization to identify impediments to its flow, surfacing insights on how to improve it, and providing the control to guide continuous improvement.

Progressive release management is an essential component to improving the flow

of work in an organization. It increases the number of features organizations deliver while reducing the risk of implementing changes by adopting some key practices:

- Creating small, multifunctional teams who release autonomously
- Shifting functional and non-functional testing left to fail and remediate early
- Automating the release process to reduce the steps, delays and errors
- Including governance checks in the workflow to save on errors and time spend in CAB
- Shortening the Route-to-Live to remove errors and delays between unalike stages
- Automating environment provisioning to avoid delays waiting for shared infrastructure

Streamlining Release Management

Release management is a step in a value stream, and not a value adding step. Value stream management requires that time spent on non value adding steps is reduced as much as possible or that the step is removed entirely. We can't remove this step, since it has to happen in order for the changes to reach the production environment and therefore the customer. There are non value adding steps that can be removed entirely; a good example is a Change Advisory/Approval Board meeting which can be replaced with a proven combination of automation and peer review.

The goal is to shorten the time between planning and delivering a release and there are a number of ways, using DevOps practices, to achieve this:

- Make releases as small as possible
- Make releases visible and manage exceptions and conflicts
- Loosely couple teams and systems to reduce integration issues
- Use CICD to build quality, stability and reliability in early
- Use VSMPs to continually identify and reduce waste
- Automate checklists for continuous compliance
- Use dark launch techniques to reduce release risk

Release management and value stream management tools, like Plutora, support these practices and help teams identify waste and measure improvements as they continually streamlining their release processes.

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
5. How Release Management Works: An Overview
6. Release Management, DevOps, and Agile

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