

PLUTORA



WHITE
PAPER



Disciplined DevOps and Value Stream Management

Delivering value outcome faster, better, safer

- The Origins of Disciplined DevOps
- Disciplined DevOps Accelerating Value Safely
- Disciplined DevOps, Flow and CI/CD
- Continuous Compliance and Value Stream Management
- How Plutora enables Disciplined DevOps in the enterprise

The technology industry for much of its history assumed there needed to be a trade-off between throughput and stability. The advent of agile software development, although initially evolved to assist with adaptability and manage risk in uncertain working environments, soon was viewed as an accelerator for the throughput of change.

The Origins of Disciplined DevOps

But IT Operations weren't ready or able to integrate into agile ways of working, instead staying focused (and frequently solely rewarded) on stability.

Change and stability came to be in contention and the mismatched cadence between the two manifested itself in conflict around admin access to production systems, war-rooms and blame-games when releases failed and instructions had been tossed over the 'wall of confusion.' This created general discontent around the speed and quality of activities like testing and environment provisioning.

Software developers were often seen as the rule breakers, creating chaos and unpredictability, while IT operations teams were the rule-makers, creating unnecessary bureaucratic burdens and generally getting in the way. DevOps was born to resolve this conflict and finally bring balance to throughput and stability in software development.

The State of DevOps in the Enterprise

Using four key metrics, two for stability and two for throughput, **The Annual Accelerate State of DevOps** reports showed year on year that high-performing IT organizations perform better using DevOps principles:

- They deploy code **30 times more frequently** than disjointed DevOps organizations.
- They experience **60% fewer failures** than mismanaged DevOps teams. Their release cycles are up to
- **200 times faster** than those IT departments that haven't mastered DevOps collaboration.
- In cases of failure, high-performing DevOps teams recover **168 times faster** than low performing peers.

Many people felt though that it was difficult to universally define DevOps and that a lack of coherent manifestos, available patterns, frameworks and reference architectures meant that DevOps was frequently led by development and wasn't as inclusive of security and risk considerations as it should be; it still wasn't safe enough. Ever increasing regulations and governance requirements

compounded the sense that people needed more discipline in their DevOps implementations to ensure that they did not sacrifice stability and compliance to speed - that they didn't succumb to the trade-off and lose the balance of throughput and stability that DevOps seeks. The advent of Disciplined Agile provides an approach that can be extended to Disciplined DevOps.

Disciplined Agile and Disciplined DevOps

Developed by the Project Management Institute (PMI), Disciplined Agile (DA) originated as a way to move beyond simple Scrum, which often struggles to scale in large, well established enterprises, and provides mechanisms to streamline and scale the software delivery process safely.

Disciplined Agile Delivery (DAD) is a people-first, learning-oriented hybrid agile approach to IT solution delivery. It has a risk-value delivery lifecycle, is goal-driven, enterprise aware and designed to scale. It's the foundation to the DA framework. Governance strategies are built in with light-weight milestone reviews, 'standard' opportunities for increased visibility and to steer the team, enterprise awareness and robust stakeholder definition. It's an

agnostic hybrid that leverages strategies from a variety of sources.

There are Seven Principles of Disciplined Agile (DA):

1. Delight customers
2. Be awesome
3. Pragmatism
4. Context counts
5. Choice is good
6. Optimize flow
7. Enterprise awareness

The sixth principle is of particular interest to us in **value stream management** since it focuses on the optimization of flow.

Value stream management practitioners are focused on the flow of value through the value stream (product or service) with the ultimate goal of delighting customers and being awesome (a high performing organization).

Flow isn't just about throughput; it is also about ensuring that what's delivered is excellent and valuable - if it's not excellent,

flow problems will be created as a result as quality needs to be improved, problems remediated and rework interrupts planned work. If it's not valuable, then it's worthless.

The Disciplined Agile (DA) toolkit is architected into four views:

- 1. Mindset:** This builds on the foundations of agile and lean to address enterprise realities
- 2. People:** Considerations around roles, responsibilities, and team topologies
- 3. Flow:** This captures the dynamic aspects of processes via lifecycle diagrams and workflow diagrams
- 4. Practices:** This includes small strategies and techniques and uses goal diagrams as a high-level pick list of practices

Flow is of particular interest to value stream management practitioners since it also focuses on the flow of value through the value streams and starts to explain the practices of their management.

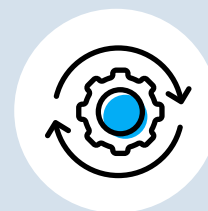
What's being described here can be considered to be the lean approach, **Value Stream Mapping**, where teams collaborate to map their value streams (a lifecycle diagram) and sometimes also create process maps (workflow diagrams). Customer journey mapping may also be included here.

Disciplined Agile uses process blades to define its framework. A process blade encompasses a cohesive collection of process options, such as practices and strategies, that should be chosen and then applied in a context sensitive manner.

The Disciplined DevOps Process Blade Includes:



Data Management



IT Operations



Release Management



Security



Support

The Practice of Value Stream Mapping and the Emergence of Value Stream Management (VSM)

Value stream centric thinking and Value Stream Mapping techniques have long been established (see chapters five and six of **'The DevOps Handbook'**) as critical to the success of organizations seeking to benefit from the implementation of DevOps practices.

In August 2018, industry analyst Forrester published **'The Forrester New Wave™: Value Stream Management Tools'** in which they stated:



“VSM is an emerging tool category that connects an organization’s business to its software delivery capability. VSM tools provide multiple roles — product managers, developers, QA, and release managers — a view into planning, health indicators, and analytics, helping them collaborate more effectively to reduce waste and focus on work that delivers value to the customer and the business. Applying the concepts of value stream and lean to software development is not new, but having a single software tool that facilitates VSM is.”

Value stream management can govern all of the areas included in the Disciplined DevOps Process Blade, each of which are explored in further detail in sections to follow.

Disciplined DevOps Accelerating Value Delivery Safely

It may seem relatively straightforward for some individuals within a large enterprise to adopt a DevOps mindset and understand the principles of working in a DevOps manner. However, learning to practice them on a day to day basis often involves breaking or adjusting existing processes, habits and behaviors that are likely deeply embedded. The traditional ways of working are frequently established in order to govern complex environments where systems and organizational structures are tightly coupled and work is executed in large batches (monoliths and projects). Disciplined DevOps seeks to accelerate throughput without compromising stability and ensure that governance around data and production systems is fully understood and as solid as ever.

Disciplined DevOps and Data Management

Data is a corporate asset and can be leveraged for improved decision making and the use and protection of it is highly regulated. In traditional working environments, accessing data in a timely

manner is frequently difficult due to onerous governance. Disciplined DevOps seeks to treat data as part of the flow and manage and govern it as an integral part of the value stream. As PMI themselves say about Data Management in the context of Disciplined DevOps:

“...we need to optimize the entire workflow, not sub-optimize our data management strategy. We need to support the overall needs of our organization, producing real value for our stakeholders.”

Data must be available for inspection and testing when it's needed so as to not introduce delays into the flow of the value stream. This requires lightweight governance and documentation using 'just enough' principles. It also may require some organizational redesign, cross-skilling or specific effort to collaborate rather than wait to ensure teams aren't waiting unduly for other teams.

Data also needs to be included in the CI/CD processes in order to improve the deployment frequency and lead time as well as reduce the change fail rate and MTTR.

Disciplined DevOps and IT Operations

It can be said that the goal of IT Operations is to be invisible: metrics such as downtime and MTTR become irrelevant when systems are stable and businesses are free to focus on time to realize value outcome as a key metric. However, many traditional enterprises suffer from high levels of technical debt which makes achieving the levels of resilience and reliability DevOps seeks a long and arduous journey.

Whilst early DevOps models frequently stopped at the moment features were deployed, as the movement has evolved, it's now well-established that the end-to-end lifecycle of changes to a value stream (product or service) is included - that is the maintenance and support of that product and the realization of value.

This means that focus should be put on paying down technical debt which frequently causes delays in the value stream and can put good governance at risk if short-cuts are taken. Beyond this, a focus on reliability and resilience using

limited blast techniques for deployment and chaos engineering helps strengthen the systems and supports governance requirements. Where existing delays occur, for example, around environment provisioning, should be a key focus for optimization.

Disciplined DevOps and Release Management

Though the “you build it, you own it” mantra is increasingly popular, the reality is that many enterprises are too challenged organizationally to allow autonomous teams to perform their own release and deployment activities.

Regulations such as segregation of duties, or long-standing organizational models where IT operations and the management of production systems is outsourced to a systems integrator mean that there are several steps teams must go through until they reach their target state.

There may also be technical debt in the form of shared and tightly coupled systems that must be paid down, but cannot happen overnight. However, this doesn't mean that the release management process in a value stream cannot be

optimized, even when the releases themselves are performed by a separate, dedicated release management team.

Teams should seek common understanding of the release governance models, allowing them to be lightweight where possible, and coordinate releases and blackout periods when they do not have the autonomy to release without potentially impacting another team's value stream. They need to collaborate to make cross value-stream improvements and release managers must focus on where they can remove delays and automate onerous bureaucracy in the form of checklists with the goal of improving both lead time and change fail rate.

Disciplined DevOps and Security

Security breaches can be devastatingly disruptive and result in long term reputational damage. A subset of DevOps, DevSecOps has emerged to promote cross-skilling and knowledge sharing between security and engineering professionals and promote the automation of security testing early in the software development and delivery lifecycle.

Disciplined DevOps and Support

The DevOps mindset goes beyond customer satisfaction to aim for customer delight. It seeks to minimize the need for customer support where possible and relies on a high level of self-service and automation. This is also a key driver in the trend to increase activity and focus on customer journey mapping and user experience (UX) design.

The DevOps mantra of 'telemetry everywhere' seeks to increase observability and pre-empt problems and failures before they become customer impacting.

The "You build it, you own it" mantra also applies here: teams will frequently start owning support issues, perhaps along with a swarming strategy.

Disciplined DevOps, Flow and CI/CD

Core to DevOps principles is the improvement of the flow of work and therefore the creation of value, without compromising the quality or performance of the product or service. Key DevOps practices are Continuous Integration and Continuous Delivery (CI/CD). Continuous Integration requires that all developers

commit code at least daily to trunk and that when the code is committed, a number of automated tests (unit, integration and user acceptance) are run and passed. The intent is to test early (as it's more time consuming and costly to remedy defects later in the software delivery cycle) and avoid having long-lived feature branches where the risk of conflicts in dependencies increases.

Practicing Continuous Integration means that software is always in a releasable state - the basis of Continuous Delivery where teams can choose when they release updates, enhancements and fixes (Product Backlog Items or PBIs) without necessarily being tied to release calendars or sprint cycles.

Continuous Delivery is different to Continuous Deployment where changes are automatically released to production when the Product Backlog Item (PBI) passes the tests. Continuous Delivery requires a manual push button to production. Traditional ways of working are challenged by these approaches since value stream flow often involves change approval or advisory boards (CABs), release teams that

manage release calendars and checklists; the purpose of all of these processes is to manage the high level of dependencies in complex systems that make it difficult to ensure governance in all the moving parts - the discipline in DevOps.

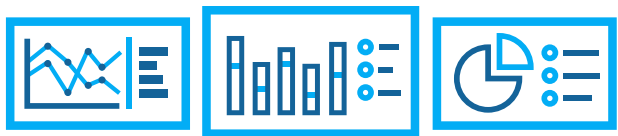
Continuous Compliance and Value Stream Management

DevOps practices such as CICD offer opportunities to automate checklists and compliance into the value stream flow. Rather than having separate steps, like CABs, continuous compliance aims to make the necessary governance checks at any and every possible point integrated into workflows so as to not allow delays, wait time or handoffs between teams during the software development and delivery lifecycle including support in production.

How Plutora Enables Disciplined DevOps in the Enterprise

Plutora is a value stream management platform (VSMP) designed to help improve software quality and increase delivery frequency while managing risk and embedding governance checklists into

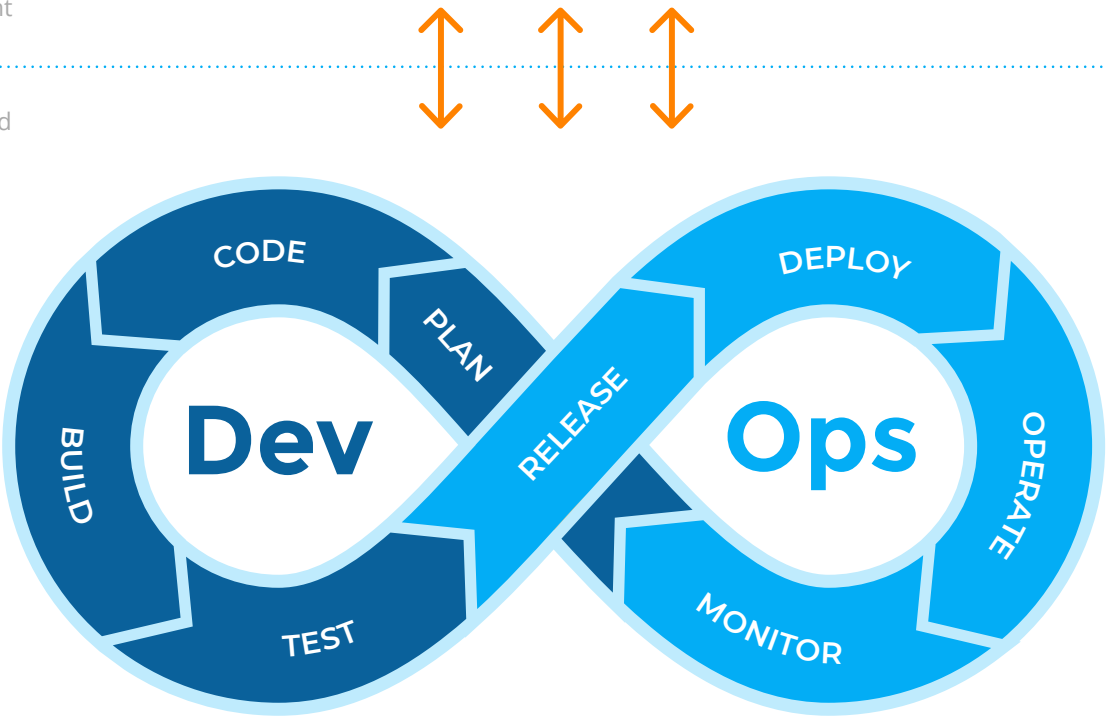
the software development and delivery lifecycle. Having visibility into the flow of work across multiple value streams with dependencies upon each other enables decision making around how to coordinate releases and to streamline collaboration amongst all parties involved in the delivery process.



Insights, Analytics and Governance

Business and Management

Engineer and Operations



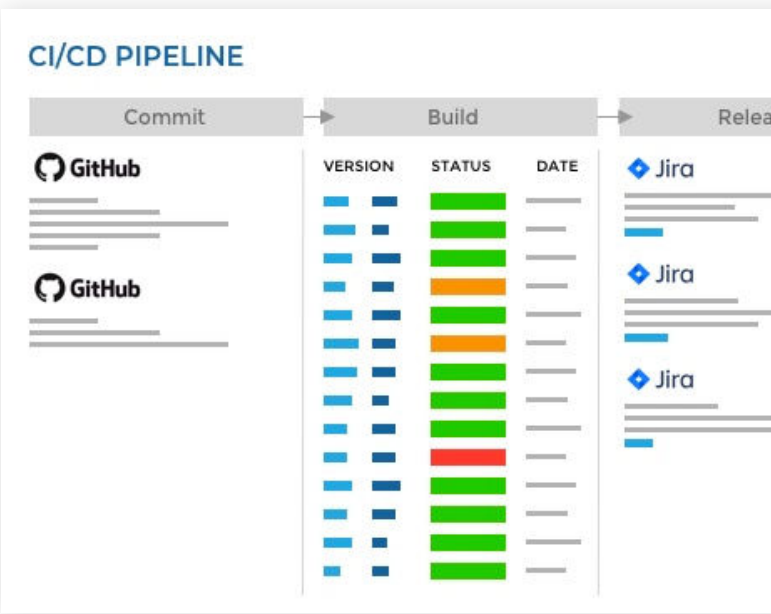
Plutora and Data Management

At a granular level, Plutora normalizes all of the data from tool sets used by different teams to provide a single platform (a converged DevOps toolchain) for cross-team members to collaborate, no matter their role. Updates are sent across the toolchain to do things like ensure Jira defects are aligned with ServiceNow incidents, and defects from a test plan can be correlated. A common language (e.g. standardized status) can be set across tools allowing for reporting that can easily be understood by everyone. Insights can then be accessed into anomalous workflows or constraints that weren't obvious before teams could inspect flow across the entirety of their DevOps toolchain. This is **DataOps**.

This sets the foundation for a common data model, and it's necessary to have a standard model of looking at things in order to have a standard method for managing them. Once these models are in place, it facilitates cross-skilling as it's easier to understand what others are doing when we have the same foundational elements.

One of the challenges of working with software is that it often seems 'invisible' (particular when compared with the

manufacturing industry where lean and value stream principles originated). But Plutora creates transparency and makes data visible for inspection by the teams who can then adapt their goals according to what they have learned. It helps make (production-like) test environments and test data available, addressing a common flow constraint in traditional ways of working.

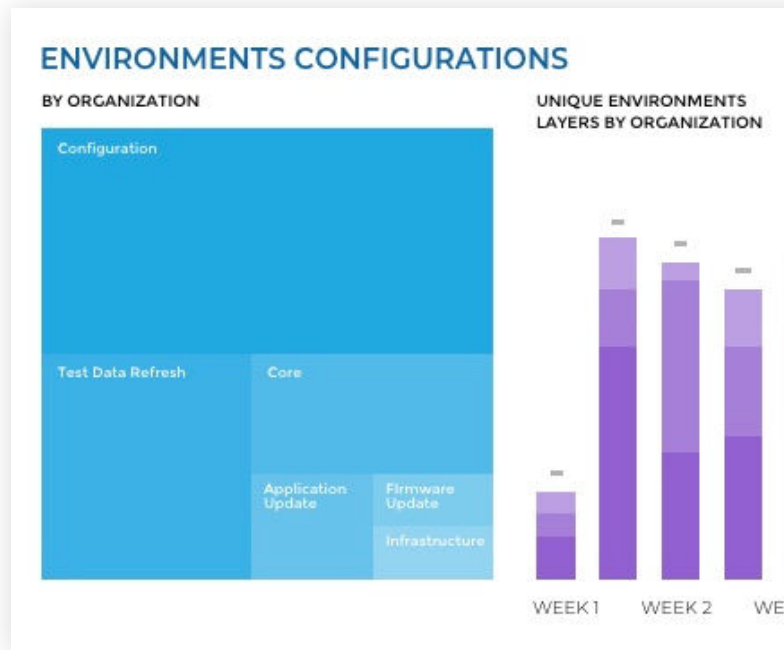


Plutora and IT Operations

A key DevOps metric for IT Operations is MTTR or Mean Time to Recovery, which is commonly held in the service desk tool, but may not be visible to all in the value stream. Plutora connects the dots between what happened earlier on in the flow and brings IT Operations and the service desk tool into the value cycle. Additionally, a multi-functional team can see end-to-end how the value of their efforts is realized. A common complaint about IT Operations in traditional enterprises is around their speed of environment provisioning. Plutora manages and thereby accelerates the environment provisioning process and, because it understands the flow of the value stream, it provides insights into constraints that can offer further improvements.

Environment management, particularly for pre-production stages can be provided as self-service to development and test teams, or value stream teams. But there's a problem; while IT Operations or the service desk aren't managing the non-prod environments, they do need to know what's in the service catalogue since there are considerable risks associated with the first known deployment being in production. Plutora provides transparency into all

environments improving collaboration and trust.



Many enterprises look to the [Site Reliability Engineering model that originated at Google](#) for inspirational practices to improve the performance of IT Operations; the 'wisdom of production' can then be used to manage risk earlier in the value stream cycle. Having performance monitoring available across all instances of the product or service from development, throughout the route to live and into production helps teams at all stages understand root causes of any problems they may be experiencing.

A common constraint in the traditional enterprise is change management,

particularly where there are heavyweight (change approval/advisory boards) processes. The Accelerate State of DevOps Report research found a correlation between higher organizational performance and lightweight, peer-reviewed, change processes.

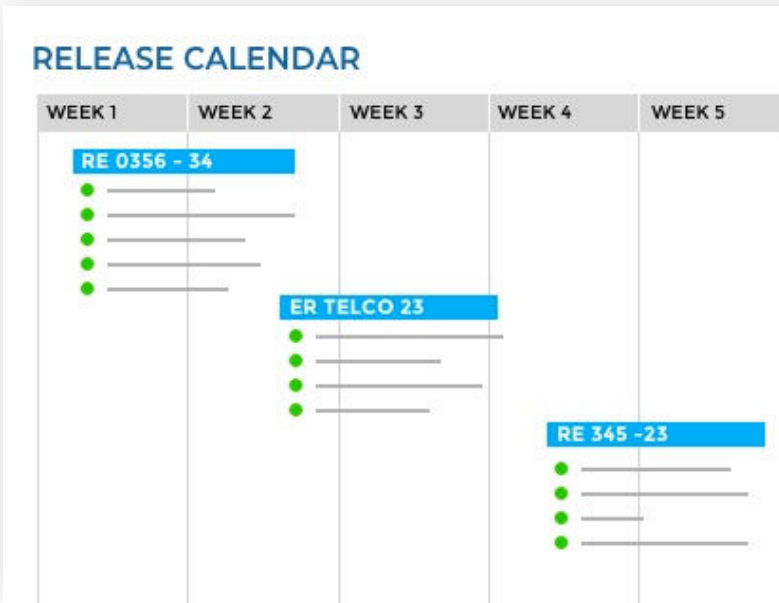
Using Plutora to automate the governance the change teams seek into the value stream’s lifecycle provides evidence and confidence that risk is not being compromised in order to optimize flow.

Using Plutora reduces administrative overhead in IT Operations as approval processes can be automated and non-production deployments can be made available via self-service. This helps teams who are moving towards a ‘you build it you own it’ way of working and checklists can be increasingly automated, building confidence through the transition from siloed to autonomous teams. Where test teams remain siloed (i.e. are not embedded into the multi-functional, autonomous teams), they are the ones that truly own the non-production environments and it is they that approve deployments into that

world; Plutora handles that approval cycle, lessening the burden on IT Operations.

Plutora and Release Management

It’s common for organizations to have different understandings of the terms release management and deployment management between teams so it’s important to create, agree and share common definitions. For many, there is a release step, then a deployment step, then another release step. The first release step is all the activities needed to prepare for deployment: opening ports or signing off change approval for example. These tend to be manual steps that precede the deployment of the updated code to the pre-production and production environments, where finally the changes are released to the customer.



In many enterprises there are separate release teams and a defined release process aligned to a release calendar. Teams may have to book release slots and this can cause delay in achieving the intended value outcomes. This happens when systems are tightly coupled or monolithic and the organizational structure reflects the same set up.

Disciplined DevOps seeks to manage these dependencies and cross-team coordination while value stream thinking guides us to remove waste in the form of delays. With these practices in place, teams should aim to loosely couple their systems (using **microservices** for example) and teams where possible. Value stream management provides the teams additional insights into the end-to-end flow of the release, so they can see how effective their collaboration really is.

In the meantime, Plutora integrates release, environment, test and deployment management and tracks release trains from initial planning through delivery across the software development and delivery lifecycle and final deployment into production.

When using Plutora, a release becomes automatically validated during multiple customer-specified phases that each have their own entry and exit gates. At each step of the delivery process, these quality gates are assigned a checklist of requirements. The requirements must be met before the software can move to the next phase of the delivery process.

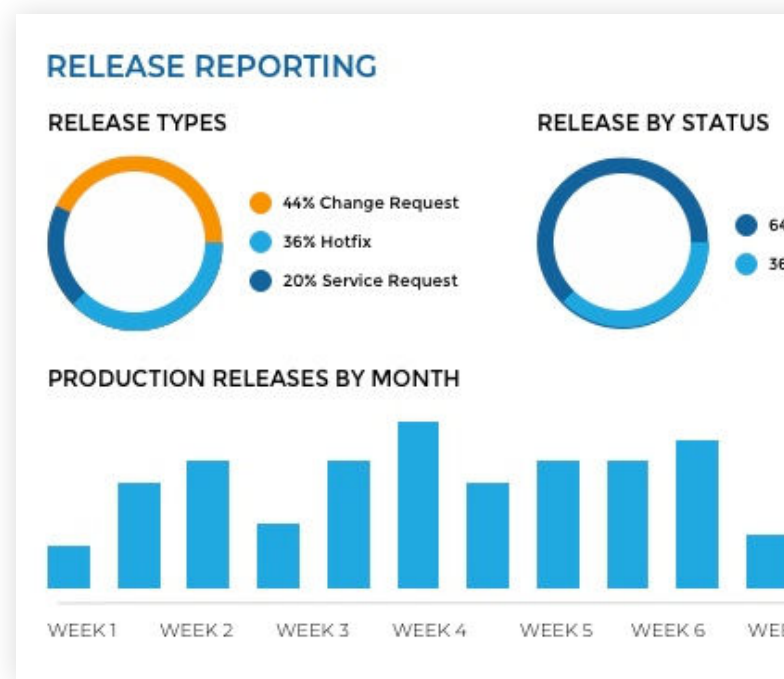
PHASES AND GATES					
E PR-3656		Organization			
PHASE NAME	START DATE	END DATE	DURATION	TYPE	
BUILD	10/08/2020	10/10/2020	2		GA
SI	10/10/2020	11/07/2020	27		PH
SIT	11/07/2020	11/11/2020	3		PH
UAT	11/11/2020	12/08/2020	28		GA
DEMO	12/08/2020	12/20/2020	12		PH

For example, each QA phase in the route to live can have its own exit gate. The exit gate could be set to consist of criteria such as “Completion of all test cases”, “No severity 1 or 2 defects outstanding” and “Signed Off Test Completion Report.”

When all exit conditions are met, the software can progress to the next delivery phase on the defined path to production or route to live, much as happens automatically in a CI/CD pipeline to the development trunk.

Confirmation of compliance can be configured to be either manual or automated, depending upon an organization’s requirements and the level of their delivery capabilities.

As teams move towards more autonomous ways of working, the Plutora release management capabilities can be used to manage and mitigate the risk of change or deployment failure as learning about the new ways of working and trust in them builds to support the adoption of DevOps principles.



Where organizations have release checklists, they can look to Plutora to automate these checks as described, removing this onerous, unpopular approach that frequently burdens the team and causes additional delays. Other DevOps practices such as limited blast radius approaches like canary and blue/green deployments can also be built into the release process to further reduce risk.



Plutora and Security

Security teams are also famous for providing checklists for the teams to complete which can be equally unpopular, as can be waiting for a decision that more often than not turns out to be a 'no'. **DevSecOps** suggests improving collaboration and learning between security and engineering teams and automating security tests wherever possible rather than waiting for a penetration test in production. This process may be difficult for teams to know where to start and if the relationship is dysfunctional, the conversation is unlikely to happen naturally.

Since Plutora provides insights into the end-to-end value stream, it can provide

insights that help spot when security is a bottleneck and interrupting flow. This data can be used as a starting point to open up collaboration. The teams can work together to automate security checklists and testing and gate security practices thereby ensuring best practice in every pipeline independent of any methodology. Engineering teams can invite their peers in security to help ensure non-functional requirements, like security, are pushed left into their Jira backlog and form part of the release requirements - like doing a threat model for example.

Plutora will interrogate the pipeline to ensure the automated tests have been done and block the release if not. This can also be done manually, but the steps still need to be acknowledged.

Plutora and Support

The support desk are typically the most overworked, underappreciated people in the IT Operations team, which puts them at the bottom of the technology pile, and bottom of the overall organizational hierarchy in many cases. This hardly seems fair or sensible when they are the frontline to feedback from the customer.

Their ticketing or service desk tools are frequently segregated from the rest of the value stream, and are rarely considered part of the CI/CD pipeline although they may be architected as part of the DevOps toolchain. When teams act like a value stream, they begin to operate in a way that supports a long lived product or service; they aren't delivering a batch of requirements to production as a project to walk away from (a wedding) but they are continually providing enhancements to customers for as long as it's sustainable (a marriage). Feedback from customers becomes elevated and a vital part of the backlog definition, prioritization and refinement task. When the feedback is coming into the support team though, who are separate and using a separate tool from the development teams, making

the connection between the service desk and the backlog becomes vital; defects can become user stories.

When teams see this continuous enhancement to their value stream, they become interested in cycle time, or flow, and the identification of bottlenecks (waste and delay) become a key concern.

Plutora allows the value stream to see the support team's role in identifying and fixing problems and providing useful data and feedback back into the whole value stream team; insights are available that have previously been difficult to see.

It becomes possible to identify correlations between deployment rate, change fail rate and MTTR ultimately reducing ticket rates, increasing productivity and quality and reducing MTTR reduction - resources become available to generate more value for the product, not just keep it running.

Summary and Conclusion

By breaking down the barriers to collaboration, Plutora unblocks many commonly-experienced DevOps constraints. At its core, Plutora is a purpose-built, end-to-end release and delivery assurance solution that improves DevOps collaboration, removes delivery bottlenecks, and ensures high quality in software releases; together these capabilities optimize flow in the value stream.

These value stream management capabilities combine to provide a Disciplined DevOps enablement platform that helps IT organizations marry software development speed with reliability in

production to fulfil their goal to be a high performing organization.

For large enterprises that are working towards minimizing the risk of software failure by transitioning from manual releases to automated processes for business critical applications, Plutora mitigates the risk by enabling end-to-end value stream visibility and driving increased collaboration across all involved in the software development and delivery process. As the organizations' and teams' DevOps capabilities improve, Plutora emerges as an end-to-end value stream management platform that enables the priority to move to the flow of value, over the coordination of processes.

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