



Real-Time Visualization of Value Streams with CloudBees DevOptics



Contents

03	EXECUTIVE SUMMARY
04	THE COMPLEXITY OF SOFTWARE VALUE STREAMS
07	CONTINUOUS OPTIMIZATION OF CONTINUOUS DELIVERY USING CLOUDBEES DEVOPTICS
11	CONCLUSION



EXECUTIVE SUMMARY

DevOps culture and continuous delivery as a software development and delivery approach continue to prove critical in delivering value to users quickly, safely and in a sustainable manner. However, when these practices are scaled beyond small independent teams, the visibility, coordination and orchestration required become significantly more difficult to achieve. As scale increases, it becomes more and more challenging to discover and maintain the visibility and insights that allow for effective optimization of the process, tooling and automation required for continuous delivery.

The technique of software value stream mapping offers an elegant solution to these challenges. The technique has its roots in lean manufacturing, where it was created and used to visualize and analyze how information and raw materials flow through an organization and its various processes that turn inputs into outputs on the journey to deliver value. A value stream map presents a big picture view of this flow, and documents the duration of all the various processes, as well as highlighting when and where the flow is paused or interrupted and where handoffs occur, such that waste can be easily identified and subsequently discussed and addressed.

With software value stream mapping, the raw materials or inputs include ideas, user stories, application source code, documentation source, infrastructure source code and tests. The processing includes planning, development, software builds, multiple phases of deployment and testing and maintenance. For software that is deployed on-premise, there is also processing for packaging and distribution. The output is software that delivers value to its users. This whitepaper takes a short look at why mapping and management of software value streams is so important and how CloudBees DevOptics can help organizations create living maps of their software value streams that provide a basis for continuous optimization of continuous delivery processes and automation.



The Complexity of Software Value Streams

Software value streams are typically complex (often more complex than their manufacturing counterparts) for a number of reasons. This increased complexity introduces additional opportunities for various forms of waste – often referred to as ‘Muda’ (Japanese for “futility; uselessness; wastefulness”) in the lean movement – and emphasizes the need for mapping and live management of value streams in order to maximize value flow on a continual basis. Let’s take a look at the five reasons behind this inherent complexity.

NON-LINEAR FLOW

Firstly, the flow across a software value stream is typically non-linear with strong incremental and iterative elements, unlike typical factory production and assembly lines. With continuous delivery and its emphasis on amplified and automated feedback loops, change is expected and embraced as an important enabler in ensuring a focus on delivering valuable outcomes with high quality as opposed to simply maximizing output.

Hurwitz & Associates also asked survey participants [how implementing continuous delivery](#) has impacted their business.

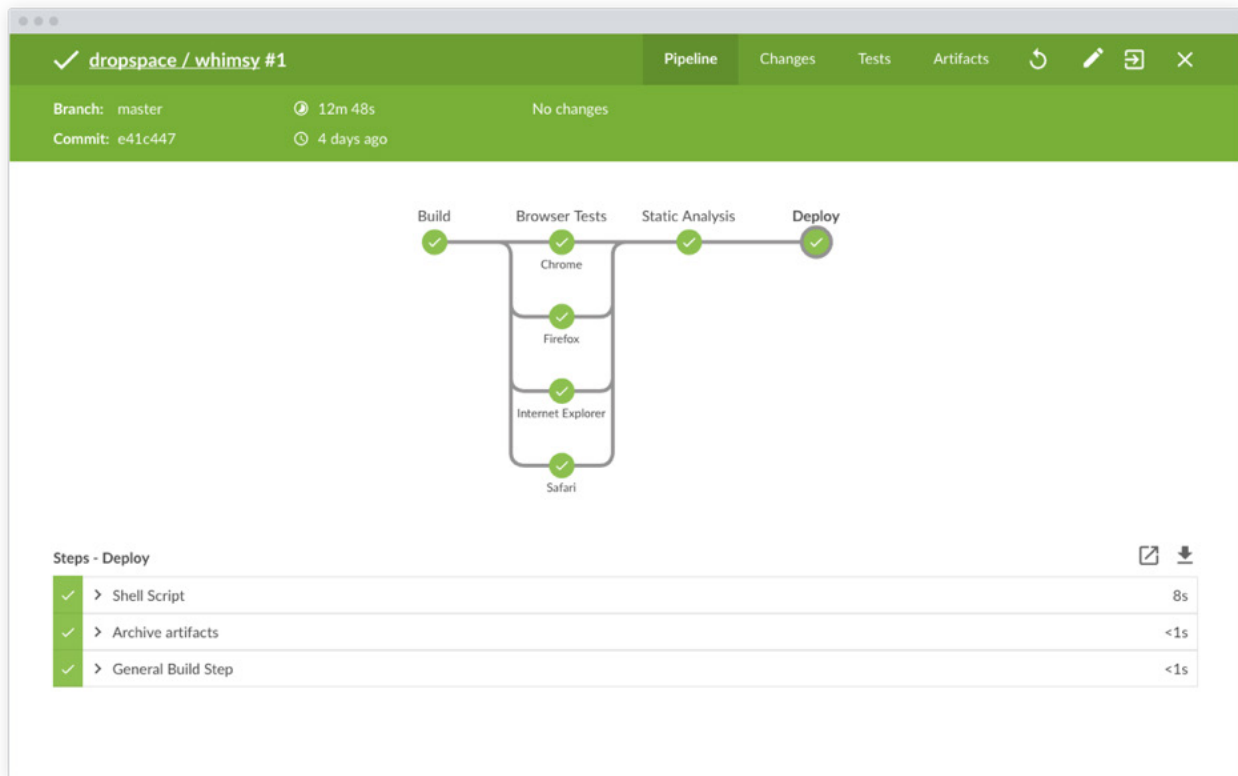
DYNAMIC RAW MATERIALS

Secondly, unlike a traditional plant assembly line where the raw materials are the same every time, with software the issues being worked on, the code being committed and environments are continually evolving, even when the automation and process remain unchanged.



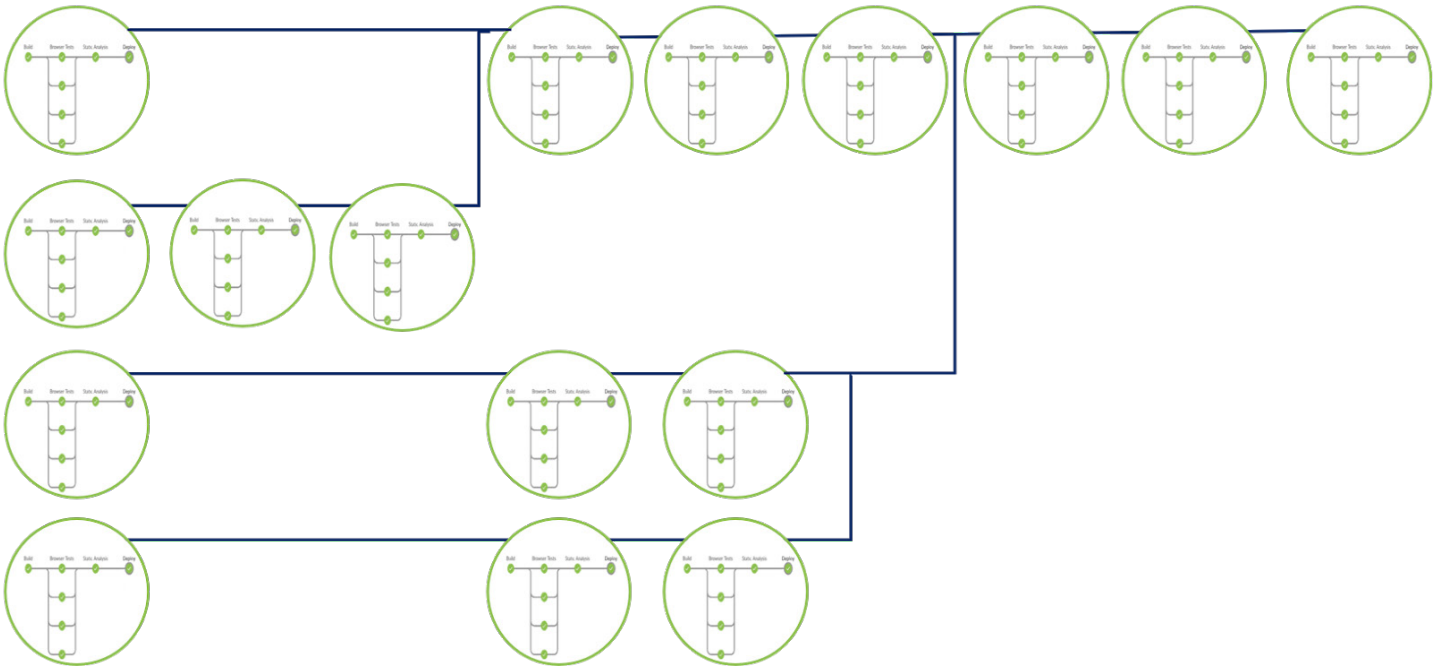
SCALE

Thirdly, software is becoming inherently larger; not through traditional growth of a monolith, but through loosely-coupled microservices architectures allowing for applications built as systems of systems. Small, independent and autonomous agile teams have clarity of purpose and individual ownership of components. Their work is typically focused around single software pipelines, and each team has good visibility into their specific inputs, pipelines and dependencies.



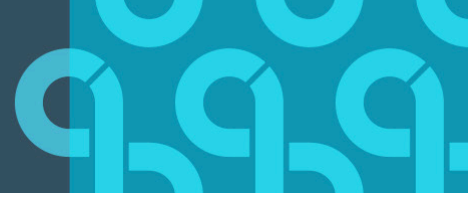


Everybody on the small agile team has a good handle on their roles, responsibilities and individual expectations. But when we move beyond the single team and start viewing it from the perspective of entire applications and systems, the picture looks very different.



We have many teams working on parts of something much bigger. We not only have cross-functional development teams, but also other domain-specific teams that play key roles in the flow of value through the software delivery lifecycle; for example, release, security, operations and performance teams. Often, components are handed between such teams to support specific processes along the way to releasing. We also have inherent dependencies between teams due to the application architecture with components or frameworks that are used by others downstream. For embedded software and cyber-physical systems, which we increasingly rely on in our day-to-day lives, complex interactions between hardware and software must also be considered.

As this scale increases, we start to lose visibility. It's hard for people upstream to see what impact their work has downstream. Projects and teams that are downstream have little-to-no-visibility on what is going on upstream.



GEOGRAPHICAL DISPERSION

Fourthly, complexity is heightened because a single application might be built by geographically distributed teams. The time zone differences create global dependencies, handoffs and waiting which make it more challenging to gain and maintain visibility into the process.

INFORMATION DISPERSION

Finally, in the course of building software, we create a lot of important information – information that is both part of the products we are building (e.g. source code), information used to orchestrate the delivery of software (e.g. work items/issues, build job and pipeline definitions), as well as information created as a by-product of the continuous delivery process (e.g. build and test results and metadata). A best of breed tooling approach has naturally risen to the fore, but leads to a situation where all of this information is located in different tools and systems.

Continuous Optimization of Continuous Delivery Using CloudBees DevOptics

Because of the complexities outlined above, mapping software value streams is unfortunately not a simple activity. It is difficult to create and visualize context between the different teams, the information in the various tools and the part they're playing in the bigger picture of application delivery. If we can't see that big picture and don't have the right context, we can't effectively optimize. In Jenkins, you can see individual jobs and pipelines in isolation. Likewise, you can look at work items and issues in Jira or at Git commit details, but there is no way to view all three in context. A one-time exercise to map value streams will certainly be valuable, but at best it represents a static view of the delivery process. It cannot provide visibility into the current state of a value stream, enable an immediate understanding of what changes are being worked and the blockages currently impacting value delivery and creating waste.

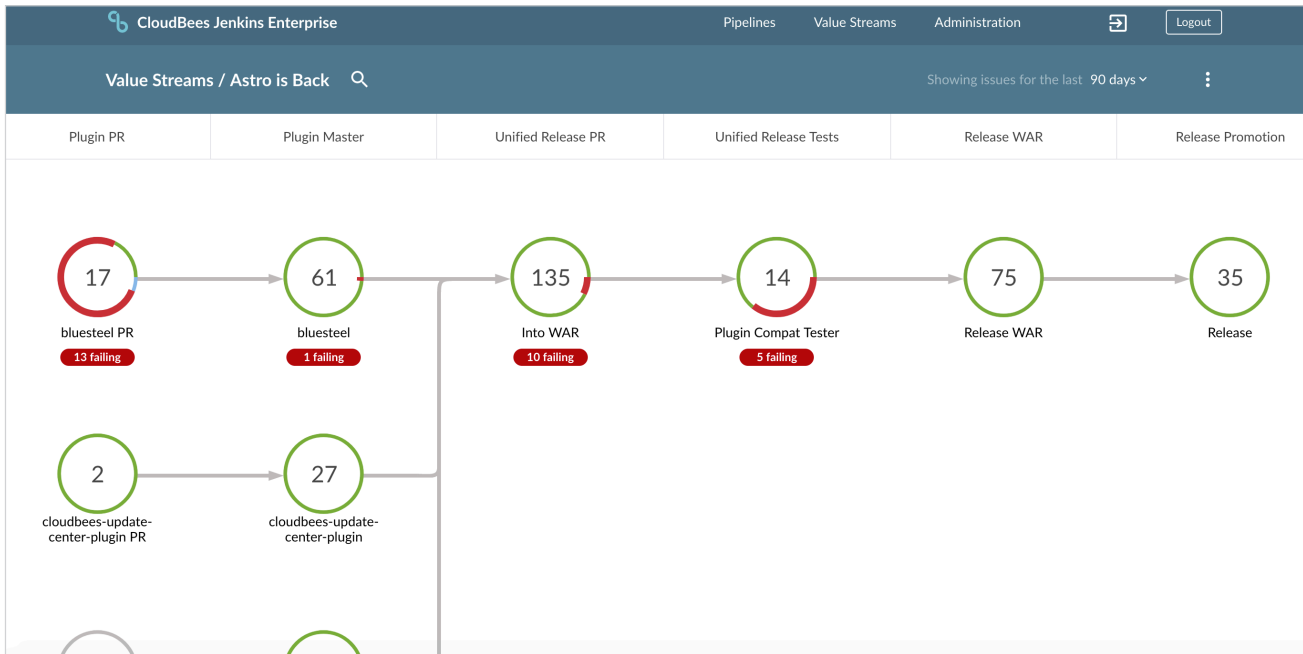


CloudBees DevOptics solves this problem by making it easy to map, then actively measure and manage your value streams, acting as a single source of truth to analyze and continually optimize the flow of value. It reduces the reliance on manual, effort-intensive and error prone data collection and reporting from multiple sources for overall reporting, and allows teams to collaborate on near real-time views of their value streams.

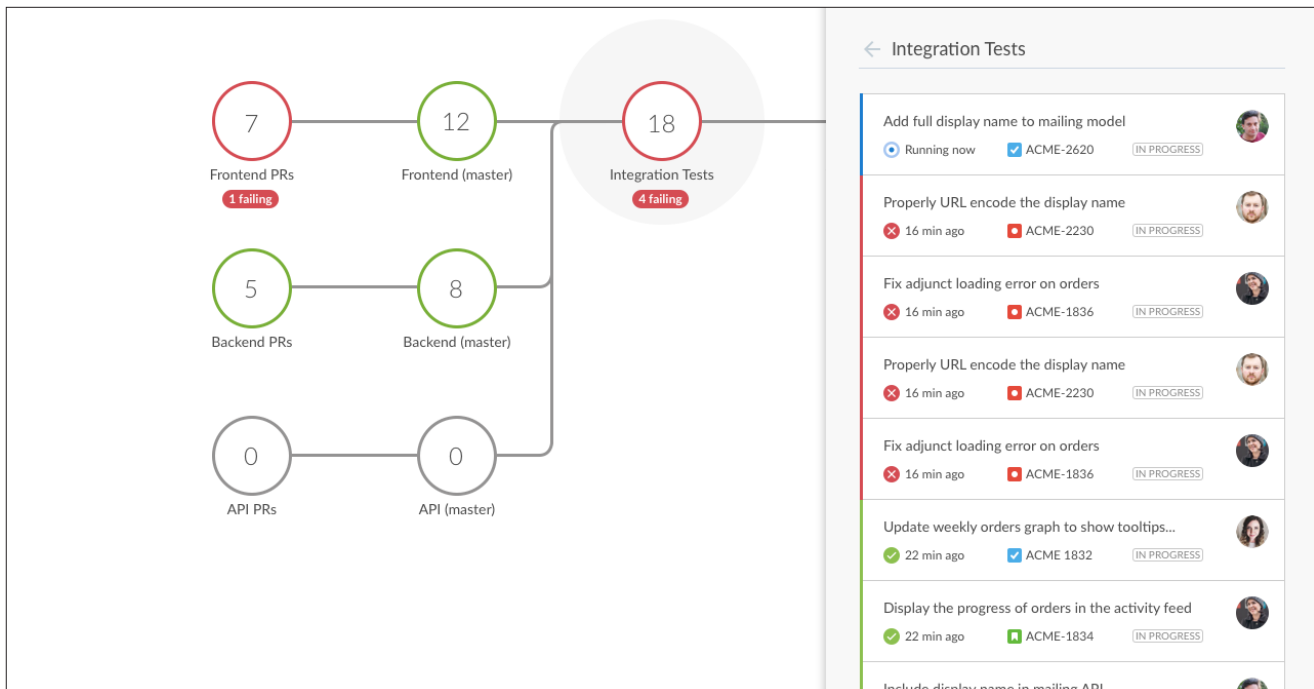
The screenshot shows the Jenkins web interface for managing Value Streams. The top navigation bar includes 'Pipelines', 'Value Streams', and 'Administration'. The current page is 'Value Streams / Acme Online SDLC'. A value stream map is displayed with stages: Build, Test, and Deploy. The map includes nodes for Store, Check Out, Credit Card, Test (with a count of 9), Release (with a count of 14), and Pet Store. A modal dialog titled 'Introducing the Value Stream Editor' is open, explaining that the editor allows building a Value Stream from Jenkins jobs by clicking a plus button on a gate. The dialog includes a diagram showing a plus sign on a gate between 'Check Out' and 'Test', with a red arrow pointing to the plus sign on the 'Test' node. A 'Close' button is at the bottom of the dialog.



Mapping a value stream in CloudBees DevOptics allows pipelines to be related as a series of gates that visually depicts how issues and commits flow from idea to production.

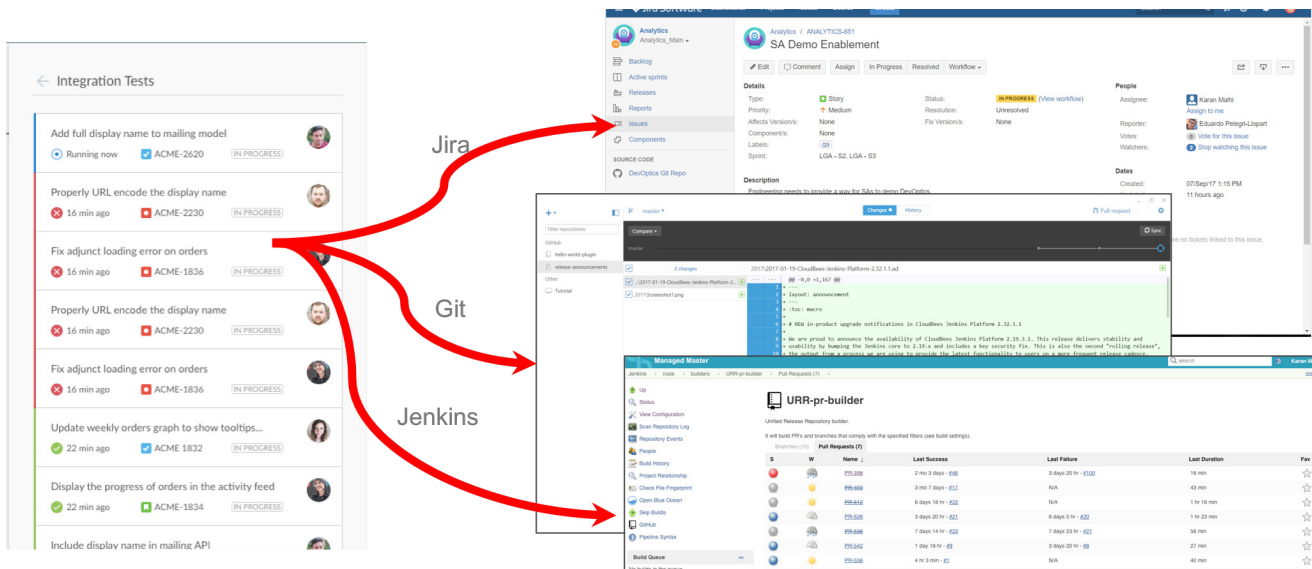


A live view of the value stream then allows bottlenecks and dependencies to be identified.





The root causes of failures are easily diagnosed. Accurate, up-to-date information, connected across the software delivery lifecycle, provides details that help individual teams work productively by searching across the lifecycle and drilling down to pipelines, repos and tickets, while also providing the big picture that helps us visualize, manage and optimize entire value streams end-to-end.





Conclusion

With CloudBees DevOptics, related information from across the software delivery lifecycle is displayed in context with easy navigation to source.


It is very easy to find issues by name, ID or assignee and see where they are in the continuous delivery process. Questions about the current state of delivery can be answered, such as:

- » Where is a given feature in the value stream?
- » What changed since yesterday? Did the fix make it to production?
- » What is choking our continuous delivery process?
- » What issues are parked in a given phase?
- » What is blocking features from progressing past a given phase?

Traditionally continuous delivery tools surface only developer-centric views and data which are typically neither relevant nor interesting to non-engineering audiences, including CxOs, VPs and other senior managers. Value streams in CloudBees DevOptics bring transparency and insight into the continuous delivery process in a manner easily consumable by everybody. This helps to align the whole organization around value-based delivery while still satisfying the needs of engineering teams.

99999999

Learn More



Read the whitepaper:
Digital Darwinism: Driving Digital Transformation with DevOps and Continuous Delivery

Download the Forrester whitepaper:
Open Source is Essential for Modern Application Delivery

Get the whitepaper:
The Business Value of Continuous Delivery