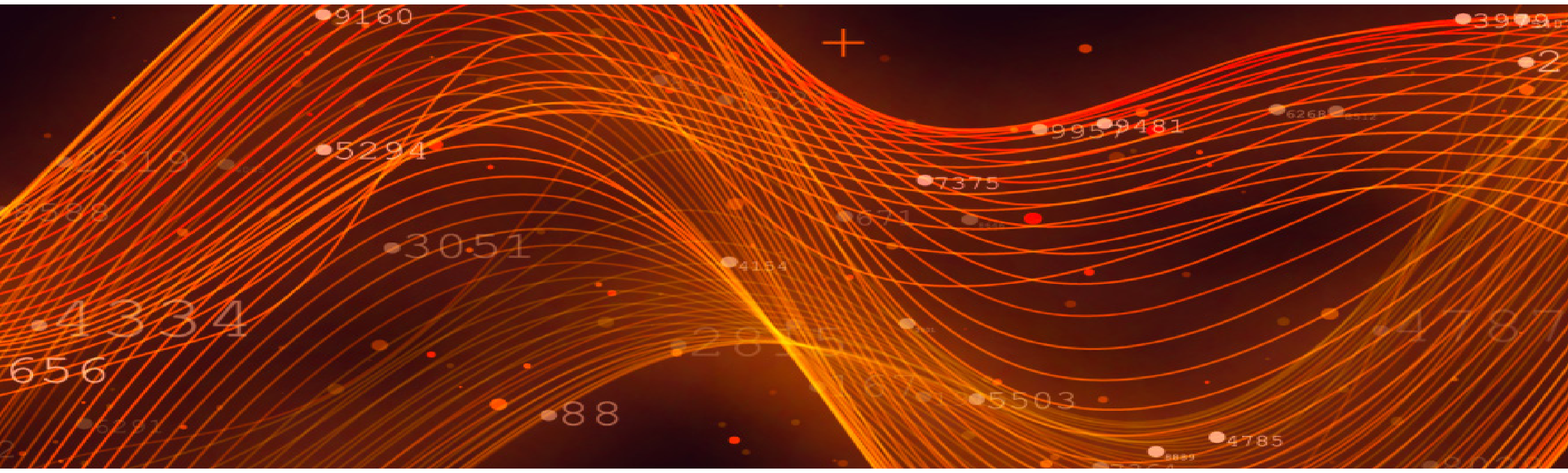
Plandek



The role of value stream metrics in effective DevOps Value Stream Management

Whitepaper

Contents

About Plandek

Plandek is an intelligent analytics platform that helps software engineering teams deliver value faster and more predictably.

Celebrated by Gartner and Forrester as a 'leading global vendor', Plandek mines data from delivery teams' toolsets and gives them the opportunity to optimise their delivery process using both intelligent insights and predictive analytics.

Co-founded in 2017 by Dan Lee (founder of Globrix) and Charlie Ponsonby (founder of Simplifydigital), Plandek is based in London and currently services the UK, Europe, the Middle East and North America.

Contact: Senior Account Executive Sam Kange, skange@plandek.com

1. Introduction

This short whitepaper considers the central role of end-to-end value stream metrics in effective DevOps Value Stream Management (VSM). The preamble of Gartner's DevOps Value Stream Management Platforms Market Guide in Sept 2020 summarises it as follows:

> 'Organisations lack end-to-end visibility into product delivery and struggle to improve their flow of value.
>
> I&O leaders, together with application leaders, must implement a DevOps value stream management platform and analyze value stream metrics to optimize the overall health of product delivery.'
>
> **Gartner**
> DevOps Value Stream Management Platforms Market Guide, Sept 2020

But what are value stream metrics and how should they be applied to 'optimise the overall health of product delivery'?

1.1 A brief introduction to Value Stream Management

DevOps Value Stream Management is an extension of the Agile DevOps philosophy. It borrows from broader lean management concepts of treating businesses as 'value generators' (for customers). And hence organisations as best viewed as a series of end-to-end 'value streams' – in existence to deliver value to the customer.

DevOps Value Stream Management applies this powerful idea to the end-to-end software delivery process. It encourages (Agile) technology teams to consider themselves as value streams (often arranged around the Agile concept of 'products'). As such, it encourages technology teams to map these value streams, to fully understand the myriad of interconnecting processes and participants across the software delivery process – from ideation, to development, testing, integration, deployment and live management.

Once these value streams are mapped and understood, they can be continuously optimised in order to drive increased value to the customer – the ultimate objective always being the delivery of value (rather than creating software that may not ultimately deliver the value intended).

As with all complex and interrelated processes, metrics become critical in tracking and driving improvement across the process. These metrics may be 'descriptive' to track performance and outcomes, but also importantly can be 'deterministic' to be much more predictive in nature.

2. DevOps VSM – metrics to embrace teams and managers

A lot has been written about Value Stream Management as a concept to help managers remain focused on delivering real value for the customer by better understanding the process of value delivery (and optimising that process).

However, Value Stream Management is most powerful when everyone is involved – so that engineering teams are involved in the Value Stream mapping process and have a set of real-time value stream metrics at the team level which they are focused on improving – as well as Product Managers, Delivery Managers, PMOs and technology leadership, who themselves look at value stream metrics at a more aggregated level to track and accelerate overall value delivery.

Indeed, Plandek is a value stream metrics tool that is designed to engage teams and managers alike. If it is only managers that buy into the value stream concept, it is likely to fail, joining the ranks of management buzz-word initiatives that fail to make a demonstrable difference.

2.1 DevOps Value Stream metrics for teams

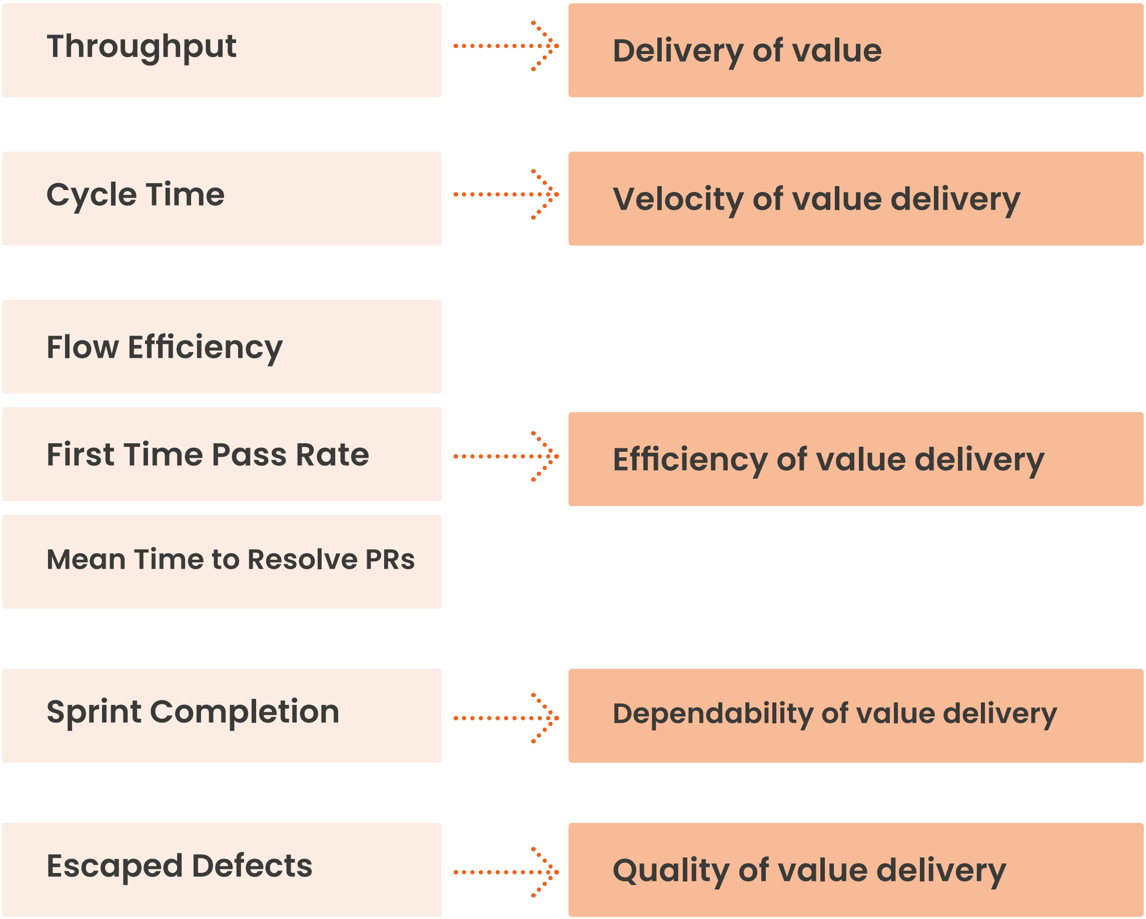
In essence, the delivery of value is in the hands of the development teams themselves. So, in our view, value stream metrics start at the team level. They should be set, configured, tracked and owned by the teams themselves.

Indeed, analytics platforms like Plandek now enable the surfacing of accurate value stream metrics in real time at the team level, so they can be more easily embedded in a team's daily routines and broader ceremonies, eg. in stand-ups, sprint retros, etc.

The figure on page 6 shows our suggested value stream metrics for teams. They are relatively simple and reflect the core aim of delivering (quality) software as rapidly and dependably as possible. They exclude DevOps metrics relating to integration and deployment (eg. Deployment Frequency) as those are often out of the control of the engineering teams themselves due to more complex architectures and organisational design constraints.

The team value stream metrics therefore only focus on the elements of the value stream that the team can directly control.

Value Stream Metrics for Teams



Throughput → Delivery of value

Cycle Time → Velocity of value delivery

Flow Efficiency
First Time Pass Rate → Efficiency of value delivery
Mean Time to Resolve PRs

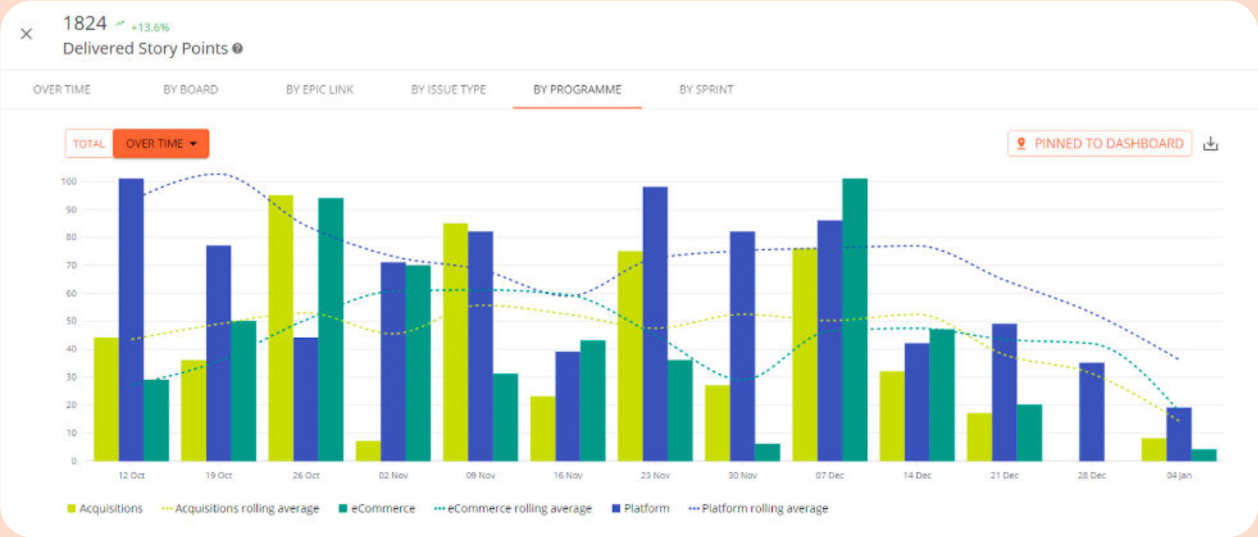
Sprint Completion → Dependability of value delivery

Escaped Defects → Quality of value delivery

Throughput metrics

Delivered Story Points metrics (eg. Velocity) are often considered problematic metrics as they may not reflect the delivery of real value – and due to the potential inconsistencies in the calculation of Story Points and how much relative complexity they represent. However, as a basic measure of throughput and how that is changing over time, they are powerful metrics around which teams can align.

Delivered Value Points can be adopted for those organisations using Value Points in addition to Story Points, as an abstract representation of the business value being delivered to end users. This is not common, but will clearly provide a more accurate view of the throughput of value (assuming that they have been agreed upon broadly and estimated consistently).

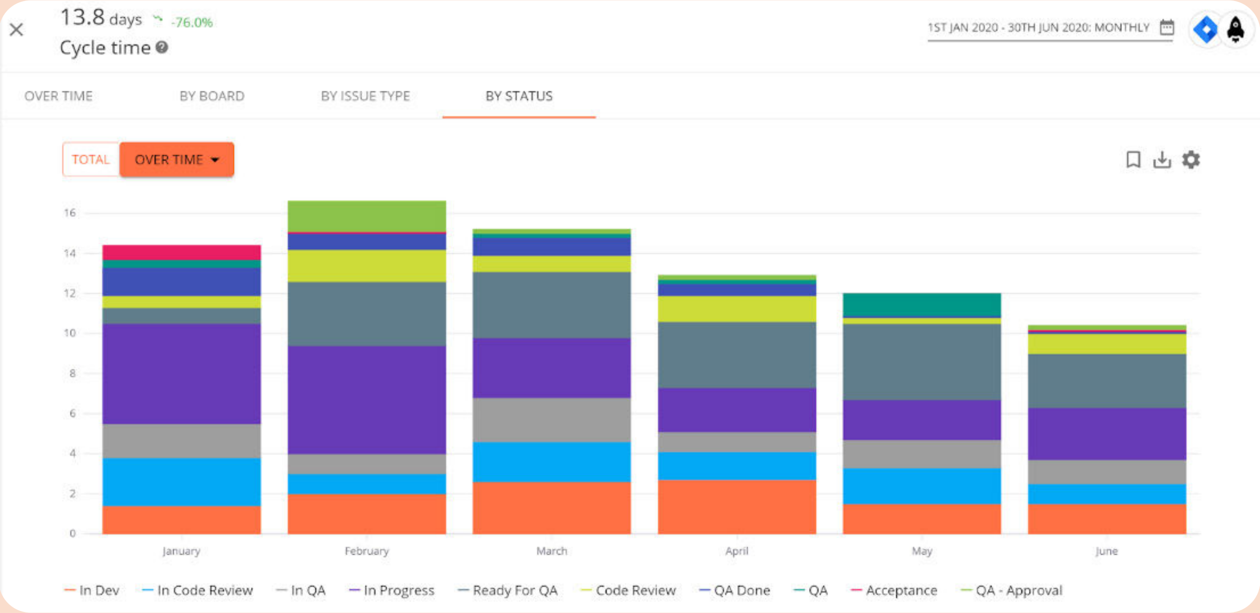


Delivered Story Points
Plandek Delivery Dashboard

Cycle Time

Cycle Time is an ideal team-level value stream metric. It simply measures the time taken to develop an increment of software – as such it is a core measure of delivery throughput. Unlike the more comprehensive measure of Lead Time, which measures the length of the entire end-to-end delivery process, Cycle Time is more often adopted by engineering teams as it reflects the steps in the process that they influence directly, ie. the time it takes from when development starts until the software is in the user's hands.

As per the image below, the Cycle Time metric view allows team to understand time spent in each Ticket status within the development cycle. Plandek has flexible analytics capability and powerful filtering to allow analysis by Status, Issue Type, Epic and any other standard or custom Ticket field. This is all plotted over any time range required.



Cycle Time
Plandek Delivery Dashboard

Flow Efficiency, First Time Pass Rate and Mean Time to Resolve Pull Requests

Flow Efficiency is a great measure of (value) delivery efficiency at team level. Its analysis enables teams to isolate and analyse each 'inactive' status (or waste) in the workflow and consider if there is scope to reduce or eliminate it. The analysis shows the relative size of each inactive status opportunity in terms of time spent in the inactive state and the volume of Tickets affected.

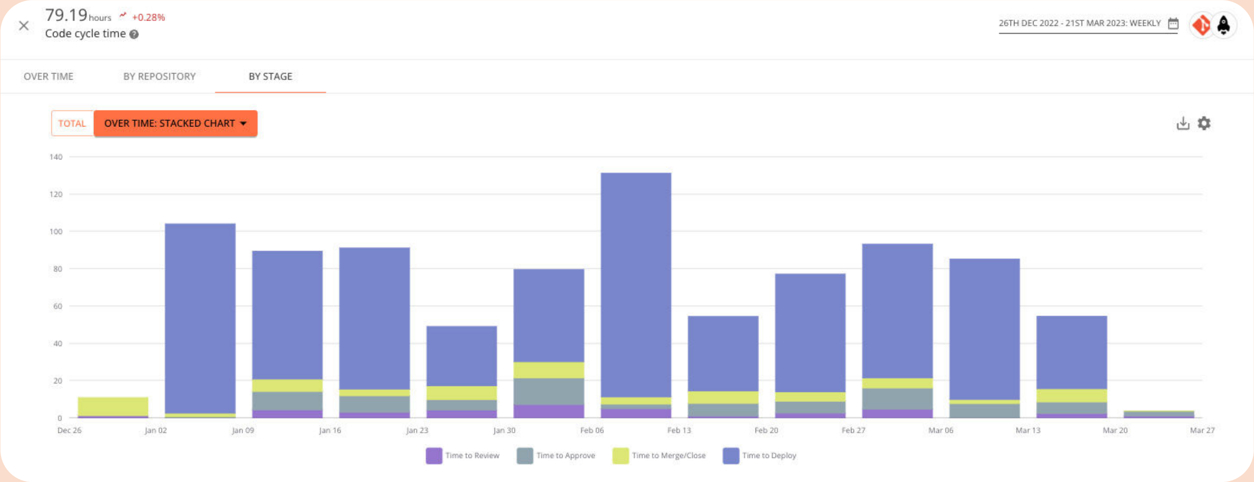
Typical opportunities to remove inactive bottlenecks included time spent with Tickets awaiting definition (eg. Sizing) and Tickets awaiting QA. Where QA queueing is considered excessive, Delivery Managers may reconsider resource allocation or enhance collaboration upstream to minimise the time Tickets are waiting.

These metrics are adopted by each team and related Scrum Master, Team Leads and Delivery Managers, so that they are tracked and analysed in daily stand-ups, Sprint retrospectives and management review meetings.

First Time Pass Rate (%) is an excellent measure of overall team health. As the name suggests, it measures the percentage of Tickets that pass QA first time (without stimulating a return transition of defect sub-task). Too often this metric is seen as an engineering quality metric, when indeed it is a better reflection of how well a team is working together and supporting one another.

A pass first-time requires the inter-dependent elements of an Agile development team to be working well and is therefore a great team-level value stream metric.

Code Cycle Time provides a deeper understanding of your Pull Request process (from open to merged/closed), which is often found to be a key bottleneck and hence is a potential area to save time and reduce overall Lead and Cycle Times. Very significant variations in time to resolve PRs are often seen between teams and individuals, with waits of over 50 hours not uncommon.



Code Cycle Time
Plandek Delivery Dashboard

Sprint Completion metrics

Experienced Agile software delivery organisations view Sprint Accuracy as a critical building block to software delivery dependability.

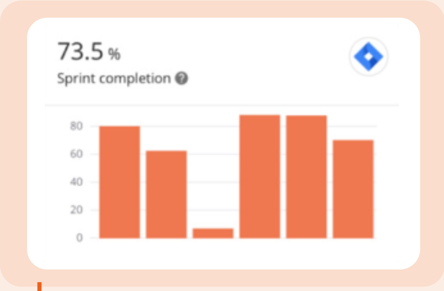
With multiple teams working on complex product workstreams over many weeks – delivery outcomes are far more predictable if individual teams consistently meet their own Sprint delivery goals over the two-week cycle of a Sprint.

The three metrics we recommend here are Sprint Completion, Sprint Target Completion and Sprint Work Added Completion.

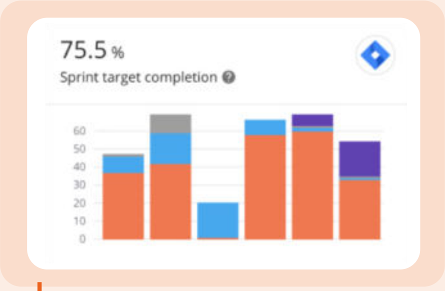
Sprint Target Completion looks at the scope you agreed to during Sprint planning and tracks how much was completed, showing you how effective the team is at establishing the right priorities and delivering them.

Sprint Work Added Completion focuses only on work that was added to a Sprint after it started (which is a very common problem for Scrum teams), whilst Sprint Completion looks at the whole picture, regardless of whether work was planned for the Sprint or added afterwards.

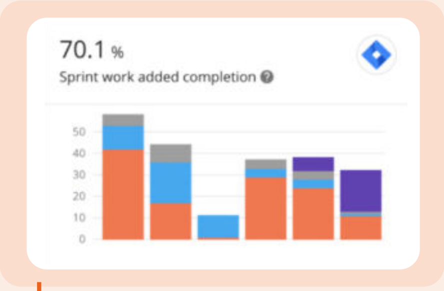
In our experience across multiple clients, Sprint Target Completion rates lower than 80% can start to cause serious dependability problems.



Sprint Completion
Plandek Reliability Dashboard



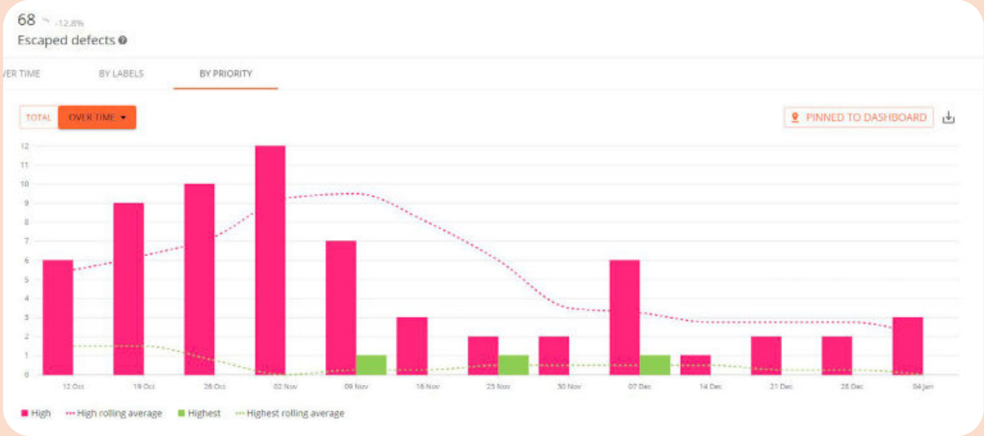
Sprint Target Completion
Plandek Reliability Dashboard



Sprint Word Added Completion
Plandek Reliability Dashboard

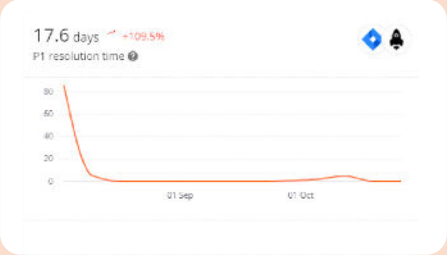
Escaped Defects and related quality metrics

And finally, Escaped Defects is a simple but effective measure of overall software delivery quality. It can be tracked in a number of ways, but most involve tracking defects by criticality/priority as per the example below.

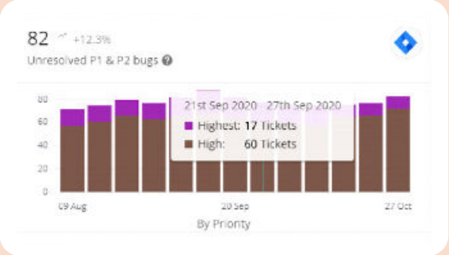


Escaped Defects
Plandek Delivery Dashboard

Other related quality metrics that directly impact defect rate and improve the quality of value delivery include P1 Resolution Time and Unresolved P1 & P2 Bugs. Both help to reduce time spent fixing bugs and reduce time diverted from feature development (and hence the delivery of value).



P1 Resolution Time
Plandek Quality Dashboard



Unresolved Bugs
Plandek Quality Dashboard

With these team-level value stream metrics, team leads can ensure that their team is effectively contributing within their value stream – to deliver value to the customer as quickly, dependably and efficiently as possible.

The metrics can be tracked over time, making sure that an improvement in one metric (eg. Cycle Time) does not lead to a detrimental effect on another metric (eg. Escaped Defects).

2.2 DevOps Value Stream metrics for leadership and managers

Technology leadership and value stream managers such as Product Managers, Delivery and DevOps Managers will take a more aggregated and holistic view of the end-to-end value delivery process.

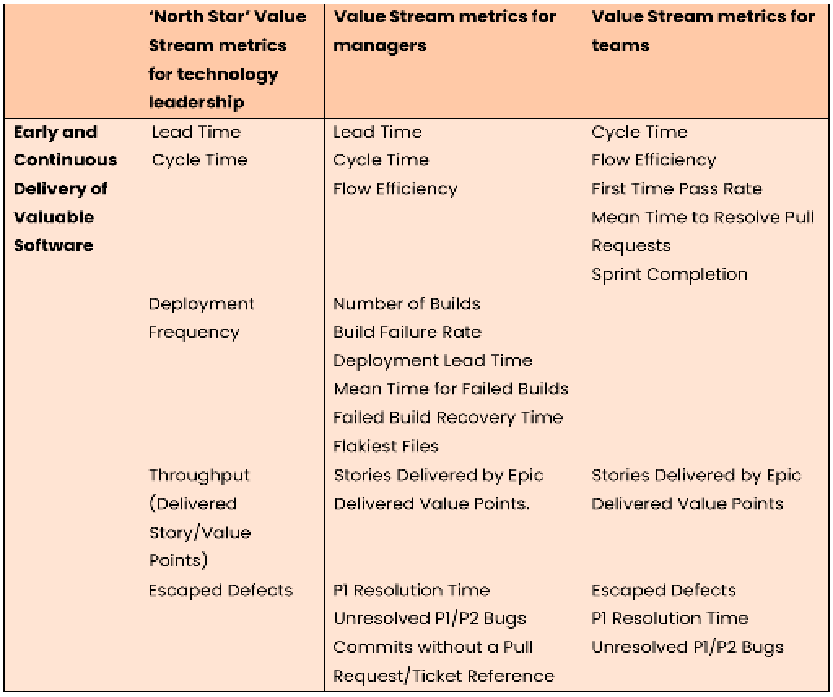
1. Technology leadership (who are overall responsible for the delivery of value across all value streams) may adopt a set of 'North Star' value stream metrics to set the overall direction for the delivery of value within and across value streams.

2. These 'North Star' value stream metrics sit above the value stream metrics adopted by the managers within value streams and the team-level value stream metrics. As such, everyone is pulling in the same direction with the manager and teams' value stream metrics helping drive improvement in the aggregate 'North Star' value stream metrics adopted by the technology leadership team.

The figure below shows how this hierarchy of value stream metrics works. Leadership, managers and teams all own complementary value stream metrics designed to track and accelerate the delivery of value software to customers.

The hierarchy of value stream metrics across an organisation

Flowing from left to right, this table shows how value stream metrics are all derived from a single goal: the early and continuous delivery of valuable software.



	'North Star' Value Stream metrics for technology leadership	**Value Stream metrics for managers**	**Value Stream metrics for teams**
Early and Continuous Delivery of Valuable Software	Lead Time Cycle Time	Lead Time Cycle Time Flow Efficiency	Cycle Time Flow Efficiency First Time Pass Rate Mean Time to Resolve Pull Requests Sprint Completion
	Deployment Frequency	Number of Builds Build Failure Rate Deployment Lead Time Mean Time for Failed Builds Failed Build Recovery Time Flakiest Files	
	Throughput (Delivered Story/Value Points)	Stories Delivered by Epic Delivered Value Points.	Stories Delivered by Epic Delivered Value Points
	Escaped Defects	P1 Resolution Time Unresolved P1/P2 Bugs Commits without a Pull Request/Ticket Reference	Escaped Defects P1 Resolution Time Unresolved P1/P2 Bugs

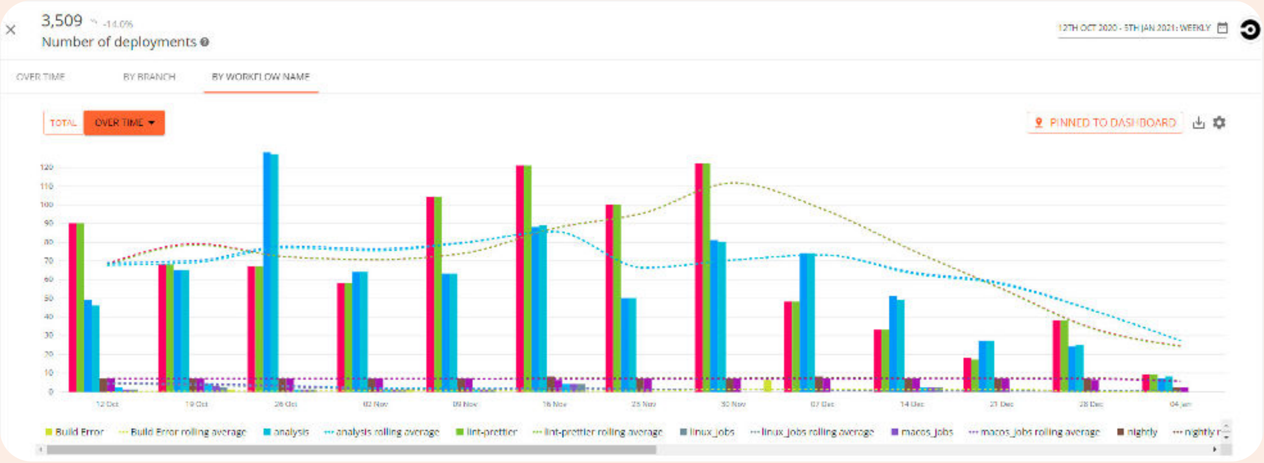
Lead Time

Lead Time is a ‘North Star’ value stream metric as time is such a critical factor in software delivery. Lead Time is a broader measure than Cycle Time as it looks at the time taken to deliver an increment of software from ideation to deployment (ie. the complete software delivery lifecycle). It is therefore a broader metric than Cycle Time which generally refers to the development element of the process only.

Lead Time is calculated by looking at completed Tickets over time and adding up the elapsed time in calendar days (including weekends) of all statuses of those Tickets from ideation to deployment.

Deployment Frequency

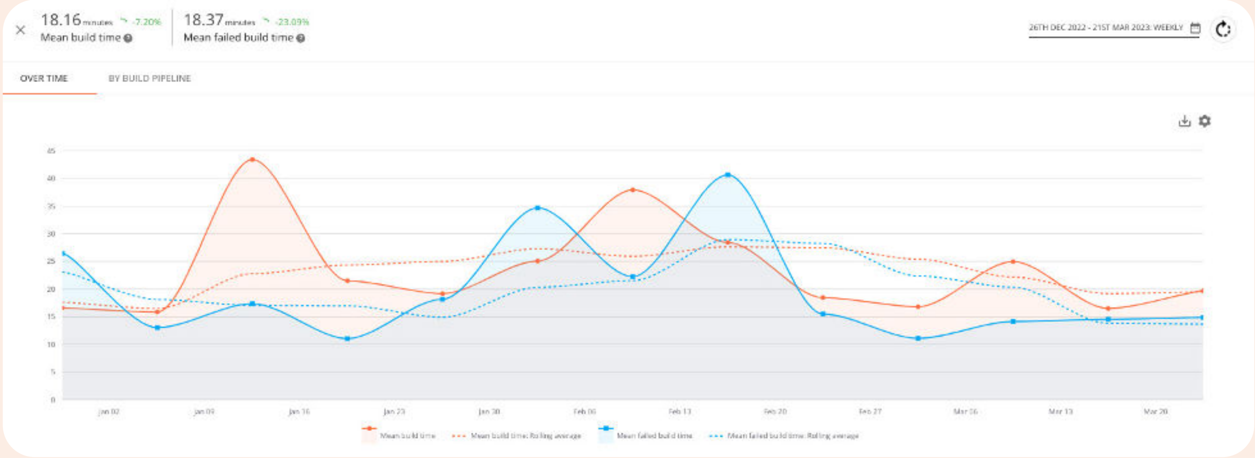
Deployment Frequency is another ‘North Star’ measure of an organisation’s Agility (when viewed alongside the other critical metrics described here). A core objective of effective value stream management is the ability to develop and deploy to live small software increments rapidly. Deployment Frequency tracks that basic competence and is a powerful metric around which to discuss effort.



Deployment Frequency
Plandek North Star Dashboard

In keeping with the 'North Star' metric of increasing Deployment Frequency, DevOps practitioners can track a range of metrics including: Number of Builds, Build Failure Rate and Deployment Cycle Time. All three are simple metrics which directly impact Deployment Frequency.

Other related DevOps metrics include Failed Build Recovery Time and Mean Time for Failed Builds, which are useful to reduce the impact of build failures. Flakiest Files is another specialist DevOps metric developed by Plandek which enables DevOps Managers to identify fragile source code files in their codebase which can be targeted for refactoring in order to reduce failed builds.



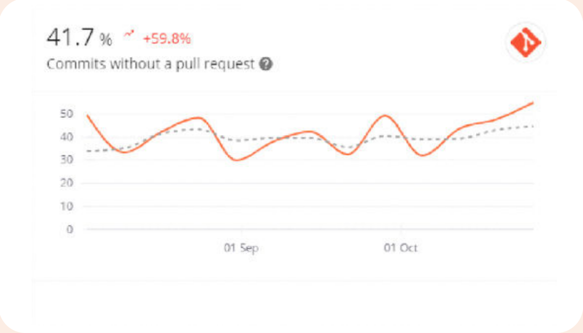
Mean Build Time vs. Mean Failed Build Time
Plandek Quality Dashboard

Our experience shows that typically you can expect to increase deployments per day (per pipeline) by 15% through a better understanding of the root cause of Build Failures and Deployment Cycle Time using Plandek.

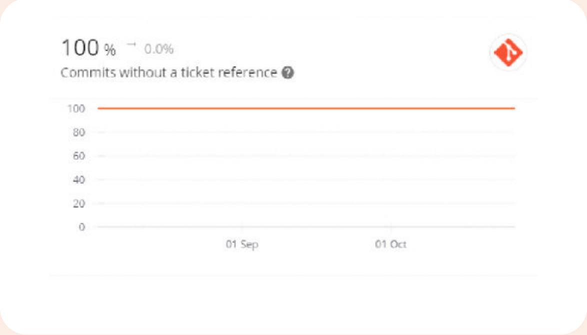
Additional engineering process quality metrics

We recommend that organisations adopt a broad view of 'quality' to include both the software output and the quality and security of the delivery process itself. As such, teams may also track and manage Commits Without a Pull Request and Commits Without a Ticket Reference.

The former ensures that all code is peer-reviewed before being committed (an important security requirement) – and the latter ensures that clear linkage between committed code and Jira Tickets, for security compliance.



Commits Without a Pull Request
Plandek Quality Dashboard



Commits Without a Ticket Ref.
Plandek Quality Dashboard

Contact Plandek

Want to learn more about Plandek?

Sign up for a free account today to explore the platform or book a custom demo with our team.

Or go to Plandek's Resources page to learn more about how you can utilise the power of metrics, Value Stream Management and end-to-end visibility.