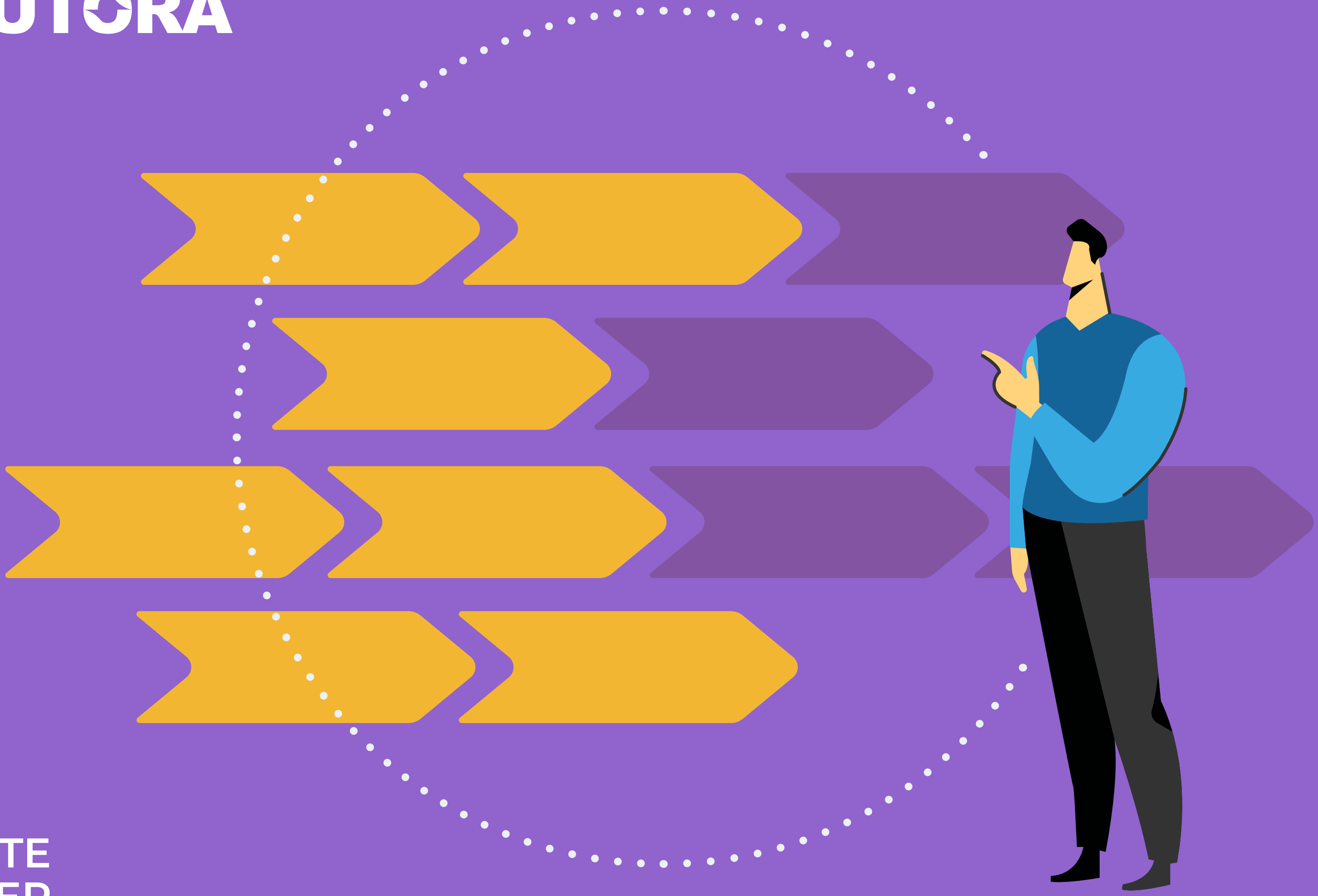
PLUTORA



WHITE
PAPER

An Introduction to Release Management and SAFe

- Agile software development allows companies to deliver production software at unprecedented speed.
- Large enterprises do not enjoy the same advantages. Green-field projects, and projects that involve legacy systems, tend to integrate at different rates, leading to the problem of multi-speed IT.
- The Scaled Agile Framework (SAFe) can help solve this problem.
- Plutora provides development teams with a seamless understanding of metrics across an entire project, allowing SAFe to achieve its full potential.

Executive Summary

Agile and waterfall software development methodologies operate at different speeds. What happens at companies that rely on older software and systems? We will explore how an enterprise relying on older software can deliver continued innovation by coping with multi-speed IT using a Scaled Agile Framework (SAFe).

Imagine working on a software development project in the days before Agile. Software took a long time to develop back then—months, maybe years, before it went into production.

The above is waterfall, the first software development methodology. Waterfall has its origins in a document dating back to 1956, in a discussion about how to program software for a building-sized air-defense supercomputer known as SAGE.

When early software engineers began to understand the industrial process of writing code, there was no suitable precursor for them to draw on. Instead, they cribbed from design processes used in construction and commercial manufacturing.

A typical development process would look like this:

1. Client requests new functionality.
2. Development team works on code.
3. Raw code is released 4 into multiple test Release phases for integration.
4. Release

In the modern era of software development, slower processes like waterfall are no longer relevant. A client's request is now a moving target. A feature that the client wants right now is not the same feature that they'll want eighteen months from now. A development team that moves slower than the pace of decision making will never be able to achieve customer satisfaction.

Thus, Agile. Seventeen years ago, Agile emerged as a collaborative framework for software development that promised to deliver functional production code in weeks, rather than months.

Agile breaks up development teams into one team per feature. These teams are known as "scrums." Each scrum works dynamically towards the goal of complete, tested software, and is capable of delivering production-ready software in four to six weeks.

In a startup environment, where development teams are expected to rapidly create software from the ground up, or rapidly iterate new features onto an existing platform, Agile is king.

> In the modern era of software development, slower processes like waterfall are no longer relevant. A client's request is now a moving target.
> A development team that moves slower than the pace of decision-making will never be able to achieve customer satisfaction.

Not every company is a startup, however, and not every startup is working with entirely new technology. Companies that work within the banking industry, payment processing, or telecom infrastructure must often work with software and protocols that were developed in the 1970s and 80s. Agile does not play nicely with these systems, leading to a problem known as multi- speed IT.

In order to cope with multi-speed IT, while delivering the benefits of continued innovation to industries that rely on older software, enterprises should investigate the Scaled Agile Framework, also known as SAFe.

What is Multi-Speed IT and Why Does it Hurt?

When development teams create applications for the banking industry, they might find themselves interacting with systems of record. If they develop for telecom, they might end up learning about the SS7 Protocol. These systems, developed during the dawn of the information age, tend to be difficult to understand, and they don't integrate well or predictably with modern programming languages. This bodes ill for applications whose shiny frontends conceal a backend that resembles MS DOS. Agile teams working on the newer component will create at their usual breakneck speed, while the teams that are assigned to the older systems will plug away at waterfall speed. Creating features for systems of record will take a long time, and integrating those features will take longer still. In the meantime, the Agile teams who've already finished their portion of the work will be left sitting on their hands.

That's multi-speed IT in a nutshell. While Agile teams will be able to develop some features relatively fast, the time to production remains overlong, and the advantages of Agile—rapid iteration, lessened time to production—are lost.

The Scaled Agile Framework (SAFe) Can Solve Multi-Speed IT

The intention of SAFe is to make up for the shortcomings of Agile that emerge when dealing with legacy software. Its key functional unit, the Agile Release Train (ART), enforces a holistic approach. That is to say, instead of isolated teams working on individual features, each train works together to create a working, integrated piece of software.ARTs are typically made of up five to twelve Agile teams. Each ART knows the start date of their project, they know the release date, and they know how much work and which specific goals they need to hit in between those two dates.

Creating features for systems of record takes a long time, and integrating those features takes longer still. In the meantime, the Agile teams who've already finished their portion of the work are left sitting on their hands.

This is a ten-week increment by default. At the end, the train delivers production software with integrated features—a Potentially Shippable Increment (PSI). Along the train's journey, there are several different "stations," representing, for example, system integration testing, user acceptance testing, and staging.

These stations are the same for every team aboard the train. If a team's feature isn't ready by the time they arrive at one of these stations, then they don't get to stay on the train. Their feature will not get included in the PSI, and will have to wait until the next increment is delivered.

In SAFe, instead of isolated teams working on individual features, each train works together to create a working, integrated piece of software.

This mechanism is in alignment with the Scaled Agile Framework's overall "definition of done." In vanilla Agile, a feature is done when the code for it is written and tested. In SAFe, a feature is done when it is written, tested, and proven to work in production. By delaying features that don't meet this definition, SAFe ensures that trains deliver PSIs with consistent levels of quality across the board.

SAFe thus includes baked-in quality control that is fundamental to solving the problem of multi-speed IT. This design philosophy helps to ensure that teams working with legacy software will deliver their features on time. It also allows Agile teams working with newer systems to create at their own pace, without getting too far ahead of the legacy teams.

Agile Release Trains Require Release Management Across Projects

Managing release trains is an art that requires extensive collaboration and coordination. Managing a single train may be relatively easy, but this single train may have dependencies which connect to other ongoing projects. That's why, in order to achieve even greater efficiencies, SAFe requires release management where multiple projects and multiple trains are involved.
As originally envisioned, multiple ARTs are managed via vast Portfolio Epics:
design documents that capture the entire scope and scale of a particular project.

Each Epic describes detailed ways that finished production software is supposed to provide new value for customers performing specific tasks within a particular industry vertical.

In order to achieve greater efficiencies, SAFe requires release management where multiple projects and multiple trains are involved.

Recall, however, that customer expectations are a moving target. The scope and scale of a Portfolio Epic may diverge from the finished production software.
Developers may find that the client changes horses midstream, and requires capabilities that differ from the original feature set, or may wish to market the product to a different customer base than originally envisioned. Alternatively, developers may discover that a planned feature might simply be beyond their capability to develop.
Due to the sheer scale of an Epic, the designers of SAFe admit that, “it may be difficult to assess how progress is being made on the development of its capabilities or features.” Thus, SAFe instructs Epic Owners to assess progress based on certain metrics, such as feature cycle time, team self-assessments, releases per year, and so on.

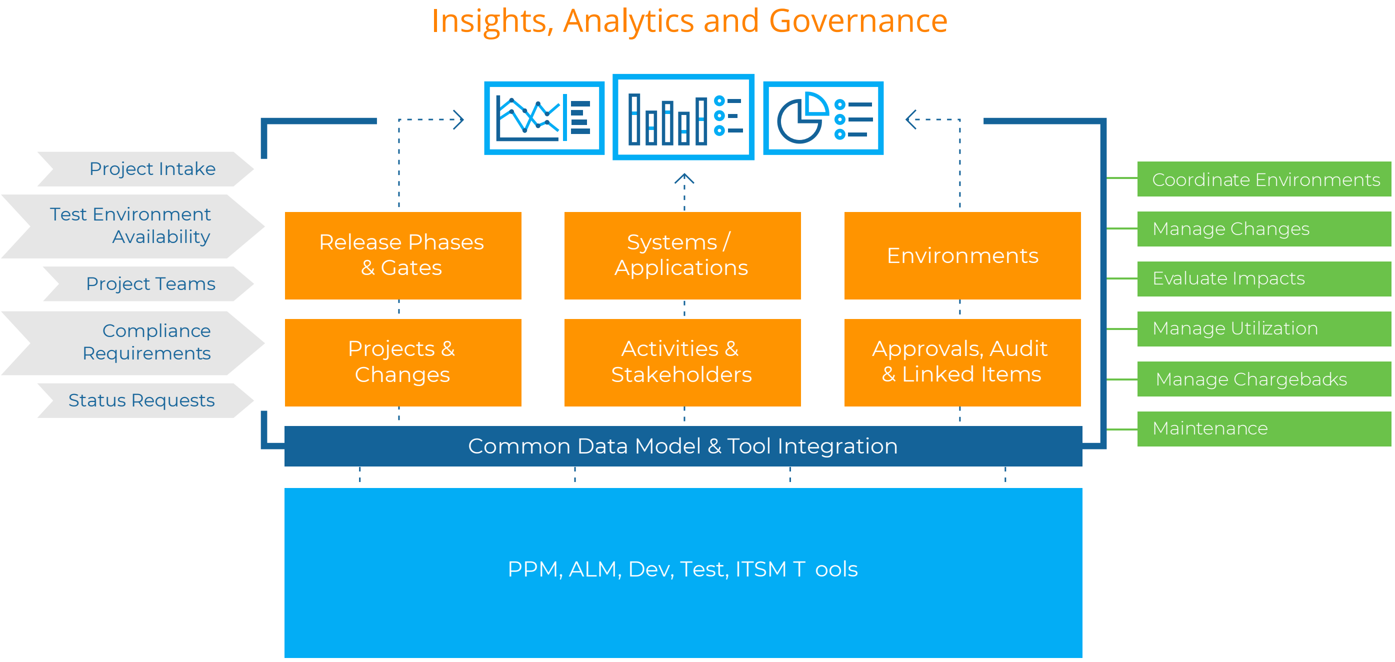
It is in the collection and interpretation of these metrics that Epic Owners might stumble. Attempts to coordinate the efforts of multiple ARTs and assess their progress usually take the form of a massive two-day meeting.

Individual Release Train Engineers will come forward with self-reported quality and deliverable metrics. Methods for capturing and storing these metrics might include whiteboards and spreadsheets. This is inadequate for complete visibility.

With manual data collection and self-reporting there’s a big opportunity for error and uncertainty to creep into the development process. In the worst- case scenario, individual teams within an ART may lose sight of the quality of their feature. Either accidentally, or through an excess of zeal, they might report that their feature is in better shape than it is.

If they successfully push for inclusion, it will result in a PSI with inconsistent quality that doesn't meet customer expectations. Other negative outcomes include delays in the project overall, or a de-scoping of requirements, neither of which benefit the business when delivering critical applications.

Teams using Plutora will benefit from a real-time status and risk profile of the project. Live visibility drives collaboration and prevents problems from occurring, particularly when there are dependencies within the project.



Plutora helps SAFe teams successfully deliver multi-speed IT projects.

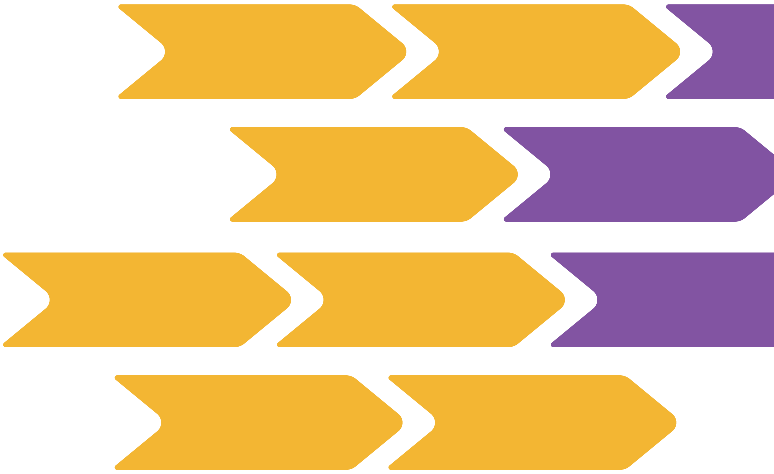
Plutora Release Management Helps Refine SAFe for the Best Possible Outcome

Plutora allows Release Train Engineers to sense problems within the release, and correct them before they become irreversible.

Trying to coordinate SAFe across an entire enterprise can be tricky, especially if the methods of data collection aren't up to the task. Teams using Plutora Release will benefit from a real-time status and risk profile of the project. Live visibility drives collaboration and prevents problems from occurring, particularly when there are dependencies within the project.

Plutora provides several dashboards that display metrics from within trains, and across each train, allowing each stakeholder to have a detailed view of metrics from across the entire project. This results in a higher-quality PSI overall, which arrives with move of its features intact. As a result, the already substantial productivity gains from SAFe are further increased.

> SAFe represents an amazing way for enterprises to adopt the speed and high-quality output of Agile, even if they don't happen to be dealing with green-field projects. Release management aligns with SAFe in a way that delivers better visibility and improved governance.



As for the worst-case scenario, Plutora release management helps to obviate this problem when used in conjunction with Plutora test environment management. The release management solution can be set up to “gate” features that enter a PSI and the environment management component can ensure that environment conflicts and contention are avoided ensuring that releases pass through the correct quality processes with the highest level of quality.

Features that don’t pass testing will automatically be removed from the PSI. Thus, RTEs can avoid some of the significant drawbacks of self-reporting while managing Agile teams within a release train.

SAFe represents an amazing way for enterprises to adopt the speed and high-quality output of Agile, even if they don’t happen to be dealing with green-field projects. Release management aligns with SAFe in a way that delivers improved visibility and improved governance.

This allows SAFe to fully live up to its potential as a design philosophy that solves multi-speed IT and helps deliver business-critical objectives.

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