Application Delivery Observability: Critical Capabilities for Modern Software Delivery

Sageable Solution Brief, presented by Pulse42

The Critical Challenge of Cloud Software Delivery

The shift to cloud computing has changed software delivery forever. Dumping on-prem systems for on-demand compute accessible from anywhere with pay-as-you-go billing has had dramatic implications for every software development leader, such as:

- The agility and flexibility of cloud infrastructure adoption and deployment has radically increased the pace of software development. Software Leaders are now expected to deliver 20x faster over the last 5 years. [Forrester, 2019]
- The easy accessibility of cloud software allows your 'empowered developers' to choose from a literal world of inconsistent platforms, tools, and resources, multiplying the time and cost of skill development, talent management, and of course vendor license costs.
- The shift in cloud computing architectures from monolithic single-system applications to highly modular, distributed services leveraging containers, microservices, and platforms drives up the volume of integrations and interconnects, adding complexity and reducing *visibility into the delivery process*

Unfortunately, we continue to see Development leaders, despite their best efforts, continue to miss objectives and key results, including estimate accuracy, software quality, delivery speed, or deadline achievement. Some of the reasons they cite include:

- Lack of process maturity and team skills in Agile, Scrum, SAFe, DevOps, SRE, CI/CD, Kanban, Extreme and other 'cloud-native' approaches slow down delivery with new skill requirements and higher costs for attracting and retaining talent
- Inconsistent tools for planning and reporting make it harder to fully grok software delivery teams' roadmap plans, backlog, sprint execution, burndown rate, quality metrics, estimate accuracy, release timelines, or productivity.
- Limited developer pipeline data, especially across multiple complex systems, means leaders have no visibility into team or product status, and therefore no predictability for planning and alignment beyond well-intended 'best effort'.

Impacts Extend Across Engineering … And Into Your Business

These challenges have far-reaching implications too – far beyond just your software engineering and product teams. This all flows into the lack of trust, speed, opportunity from business leaders frustrated that Engineering leadership cannot provide an accurate and reliable software development service.

Drives negative outcomes far and wide across the org, incl. lack of trust, shadow IT, outsourcing, inability to deliver business goals for new product introduction, extended and unreliable marketing lead-times, reduced customer acquisition, higher development and delivery costs, leadership calls for layoffs, and more.

Because when you cannot measure what matters, you end up over-rotating toward 'busy work' and easily gamed metrics (like 'lines of code' or 'story completion'), rather than zeroing in on the specific data points that you need to manage and deliver high quality software, on time, on budget.

For many technology leaders, these challenges will often cascade to one or more (or all!) of the following:

- **You cannot predict development workloads** – so you cannot accurately allocate your teams, and without accurate direction, team members work on whatever they want, wasting time on vanity projects, resume building, and greasing squeaky wheels, whether aligned to the business or not.
- **You cannot reliably deliver new software** – with limited insight and measurement, you miss your predicted estimates, and when you miss launch deadlines as a result, you still cannot accurately adjust your people or processes to address the delivery issues or provide a reliable new estimate.
- **You deliver the wrong software** – with limited visibility into roadmap, backlog, and burndown, and with 'empowered developers' making their own decisions, even if you can deliver on time, without aligning business needs you are frequently unable to deliver what your business really needs.
- **You fail to meet your commitments** – as you miss on timing, quality, feature sets, you lose trust you're your business peers, as they cannot rely on your predictions and promises as a basis to align their plans and people with new technology deliverables.
- **Your business fails to deliver on time** – new improvements and innovations lagging the market, so you lose any competitive edge you had, cannot catch up to more agile disruptors in your market, or attract new business in an increasingly competitive market.
- **You lose your customers and talent** – as more and more promises become lies, you not only lose the trust of your peers and your leaders, but also your customers, and your own team members (especially the good ones!), so your net retention sinks while new business continues to shrink.

What To Do … That You Haven't Already Tried!?

Of course, this is the life of a modern software development leader. Today, you likely deal with all of this as best you can, every day, because you need to. You are doing what you can, with what you have, even as inaccurate and unreliable as it may be. That is the job!

But you are over the chaos, cost, and career implications of 'best efforts'. You are ready to take your role, your team, and your business to the next level with true data-driven decisions. You know you need to differentiate with the products and services you deliver, instead of trying to repave a road that has been paved millions of times – the known process of software delivery. You are keen to adopt and follow best practices learned through blood, sweat and tears from across industry, across teams and across verticals rather than burning your team out trying to reinvent the wheel.

Introducing Observability for your Software Delivery Lifecycle

The solution to these significant challenges starts with accurate 'Observability' of product delivery processes. Observability happens when complex, opaque systems emit detailed data about internal activity, allowing someone outside the system to 'observe' what is happening inside that system – like reading a flow meter to 'observe' how much water you are using; or reading a car speedometer to 'observe' how fast you are going.

By collecting, integrating, analyzing all the disparate data about people, processes, and tools throughout your SDLC, you will have an easy, accurate, reliable, and dependable method to manage and direct product delivery with a comprehensive data-driven approach.

Observability across all the many and varied data sources throughout the SDLC – including people, processes, and technologies – gives you accurate and real-time visibility into every phase, every action, every outcome of your product delivery system.

- **Collect data from idea to cash** – starting at planning and through dev, test, secure, stage, instrument, release to prod, and service level achievement.

- **Integrate SDLC toolset data** – correlate data from Jira, Asana, GitHub, GitLab SonarQube, Perforce, Sauce Labs, Chef, Puppet, Ansible. Docker, Kubernetes, and any tool available to your empowered developers.
- **Integrate service systems data** – correlate data from staging and production cloud environments like AWS, GCP, Azure, IBM, Oracle or any other platform (even on-prem!) where you run code
- **Create feedback loops** – from prod back to planning, regardless of application, team, platform, methodology, or tool, with a single source of truth for deep insight and control over your delivery pipeline.

SDLC Observability in Practice: Value Stream Management

Observability in turn allows you to adopt 'Value Stream Management' a management technique or practice that focuses on increasing the flow of business value from customer request to customer delivery. Where engineering leaders monitor the 'value added' at every phase of production, to enable accurate, real-time, data-driven management of inventory, production, quality, staffing, logistics, and more.

In software product delivery, VSM leverages observability to allow leaders to actively manage the efficiency, effectiveness, productivity, and quality of software delivery using objective insights to improve accountability and reliability with data-driven decisions. For example:

- **Connect all relevant tools and sources** across dev, test, and prod, so you can measure and adjust, and in turn meet or exceed KPIs, MBOs, SLOs, OKRs, or other management metrics – accurately, objectively, and without distracting your team or leads with reporting.
- **Provide insight into 'metrics that matter'** across the SDLC, whether productivity metrics like Lead Time, Cycle Time, or 'Idea to Cash' time; quality metrics like Change Failure Rate or Mean Time Between Failures (MTBF); or operational metrics like Deployment Frequency or Mean Time to Recovery (MTTR)
- **Enable data-driven leadership decisions** for software delivery processes like roadmap planning, resource alignment, teaming, estimate updates, training allocations, release scheduling, and cost reduction.
- **Meet or exceed predictable and realistic estimates** for time to market impact, eliminate missed launches and deadlines, bridge competitive gaps, and deliver outcomes that allow you to argue for a new budget to drive new innovation.

Solution Spotlight: Product Delivery Observability

Presented by Pulse42

Pulse42 delivers a leading solution for Product Delivery Observability that will let you adopt the data-driven management techniques of Value Stream Management.

The solution provides a unique data platform that builds data relationships between teams, tools, and processes with advanced Data Analytics, Machine Learning, and Artificial Intelligence built into the foundation to make the impossible as easy as pie.

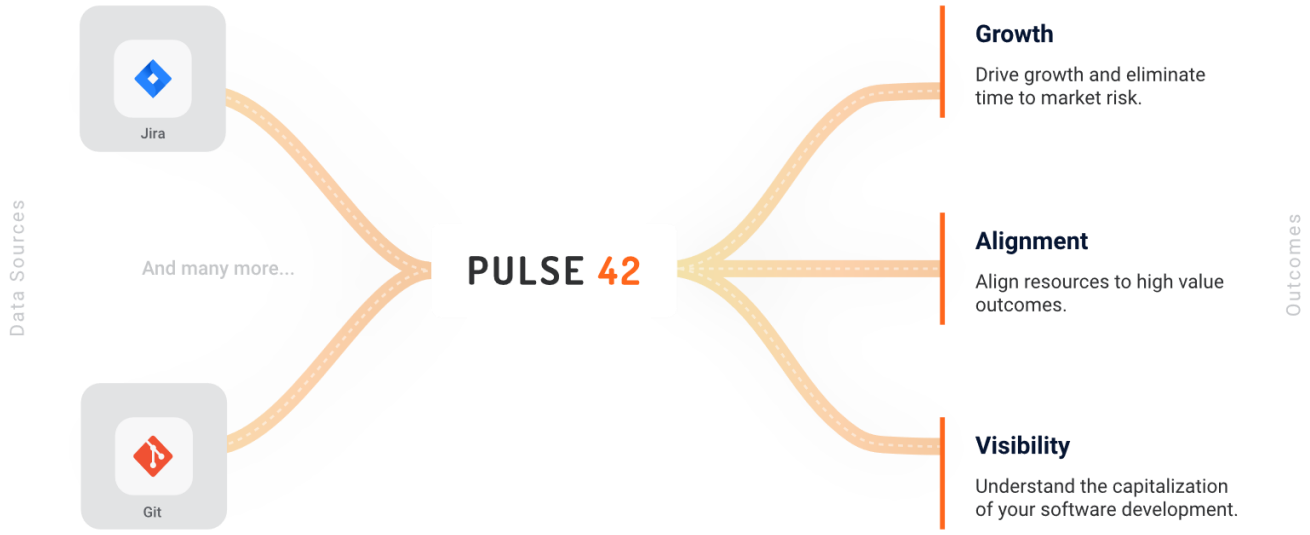
This allows Pulse42 customers to solve major challenges in product delivery and management by unifying, integrating, collating, analyzing, and reporting across a wide range of disparate processes, data sources, and siloed tools throughout their SDLC.

By connecting traditional dev tools, operations monitoring, and business insights, Pulse42 can provide deep observability not just into dev/test processes but can also bring in production and business outcome data to enable aligned context intelligence, and continuous improvement.

Benefits of Pulse42 for Product Delivery Observability

As a result, it helps you gain visibility and insight across your entire end-to-end software delivery lifecycle, enabling you to make objective and accurate data-driven decisions about the critical people, processes, and technologies needed to deliver software-based innovation. For example:

- **Observability for all tools and processes** enables better decision-making based on individual and team activities and results
- **Observability for process and progress** enables more accurate estimates and more 'promises kept' so you peers can trust your ability to deliver on time
- **Observability for tool usage and value creation** gives you better control over software, vendor, and training costs, so you can deliver on budget too
- **Observability for cycle times and quality metrics** gives you the confidence to make important and difficult decisions on teaming, forecasting, etc.
- **Observability for production release**, including operational failure and SLA/SLO achievement, facilitates agile response to boost customer satisfaction
- **Observability across projects and teams** enables better performance management with realistic, aligned, and measurable KPIs, MBOs, or OKRs.



Pulse42 Takes you Through This Journey!

Pulse42 can take you through this journey! Take advantage of our first step, our Product Delivery Maturity Assessment module and within hours get a clear snapshot of current Application Delivery Capability, with gap assessment and road map to increase Developer efficiency.

Learn more about Pulse42 at www.pulse42.io.