

PLUTORA

Mastering Software Delivery with Value Stream Management

Learn how to identify and capture value stream management metrics from your DevOps toolchain for continuous, long term improvement.

By Jeff Keyes

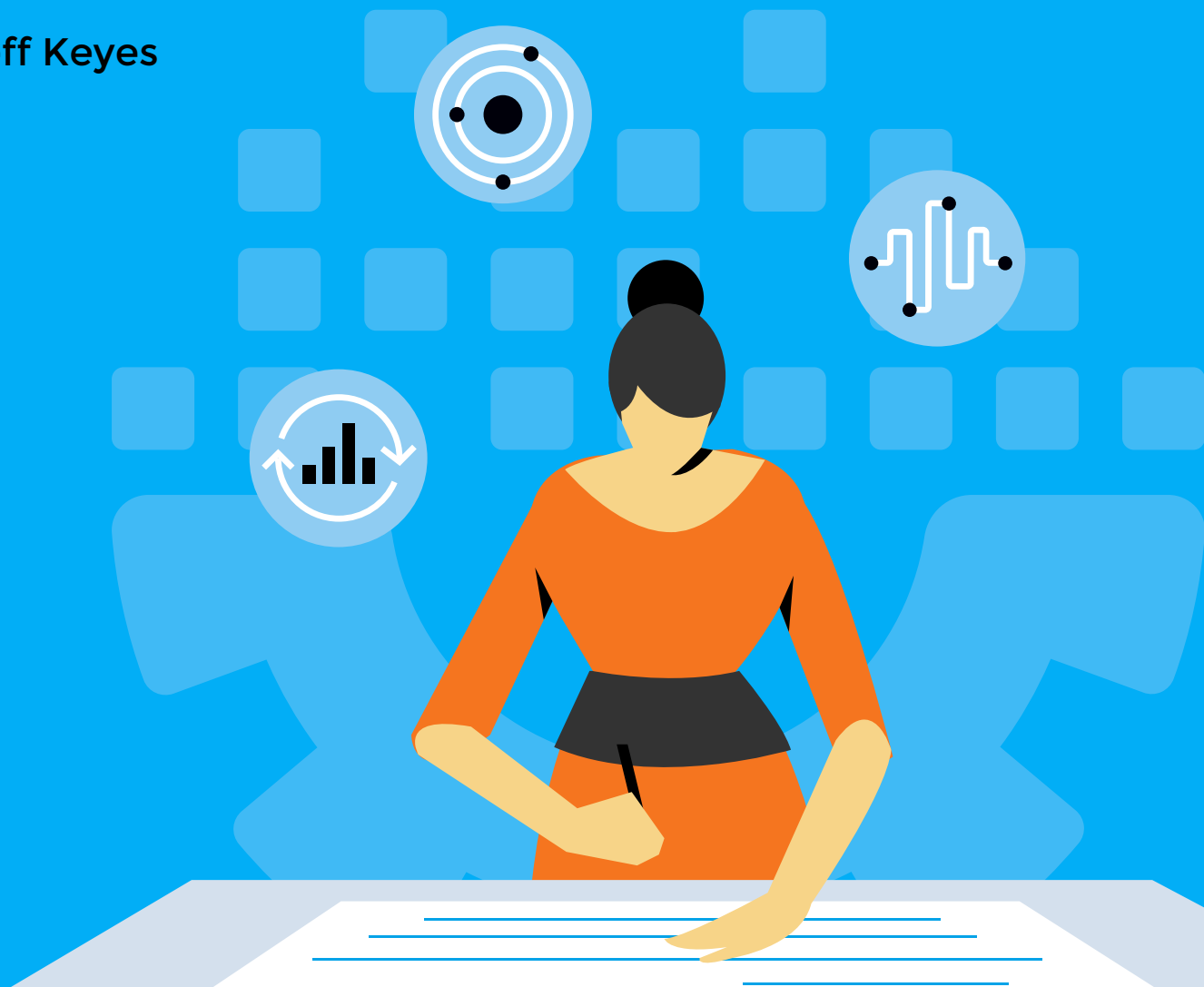


Table of Contents

Part 1: Value Stream Mapping and Management Fundamentals	4
Value Stream Mapping	5
Value Stream Mapping Goals	5
Value Stream Mapping in Technology Teams	6
Value Stream Management	7
Value Stream Management Goals	8
Using Value Stream Mapping and Management Together	9
Managing Multiple Value Streams with Dependencies	12
Release and Pipeline Management	13
Environment Management	13
Governance and Continuous Compliance	14
Monoliths and Microservices	14
Organizing Around Value Streams	14
Part 2: Building a Business Case for Value Stream Management	18
Calculating Value Flow	20
Cost:Value Ratio	21
DevOps Metrics	22
Team Metrics	23
Measuring Value Outcomes	24
Metrics in a Business Case	25
Your Value Stream Management Business Canvas	26
Part 3: How to Do Value Stream Mapping	28
Step 1: Choose Your Value Stream	29
Step 2: Gather Your Value Stream Mapping Team	29
Step 3: Get a Space	30
Step 4: Prepare Some Materials	31
Step 5: Enlist a Colleague to Support You	31
Step 6: Session One - Setting the Value Stream Mapping Scope	32
Step 7: Session One - Mapping the Steps in the Value Stream	33
Step 8: Session One: Calculating Your Current Value Stream Cycle Time	34
Step 9: Session One: Adding Some More Metrics	35
Step 10: Session One: Reflection	36

Step 11: Session Two: Mapping the Future Value Stream	37
Step 12: Session Two: Calculating the Future Value Stream Cycle Time	37
Step 13: Session Two: Adding Resource Utilization Metrics	38
Step 14: Session Two: Compare Your Metrics	38
Step 15: Session Two: Build Your Backlog	38
Step 16: Session Two: Commit to Value Stream Review Cycles	39
Part 4: How to Implement Value Stream Management	40
Maximizing Value from Value Stream Mapping	41
Defining and Establishing the Value Stream Teams	41
Appointing a Value Stream Manager	43
Identifying Initial Value Stream Metrics	44
Updating Metrics Inspection Practices	47
Managing the Value Stream Improvement Backlog	48
Part 5: The DevOps Toolchain and Value Cycle	50
The Evolution of DevOps Toolchains	51
The culture and people need to embrace a mindset of autonomy	51
Tools Standardization	53
CICD or DevOps Toolchain?	55
Release and Deployment Automation	55
The DevOps Tools Landscape	57
The Value Cycle and Value Stream Management	59
Part 6: Value Stream Management Metrics	63
The Importance of Being Measured	64
Shifting the Ownership of Metrics	65
Getting Started with Metrics	66
Advanced Value Stream Metrics	68
Flow Metrics	68
Making Metrics Visible	70
Continuous Adaptation	72
Part 7: The Value Stream Glossary	74
Further Reading	92



Part 1: Value Stream Mapping and Management Fundamentals

Value stream mapping and value stream management are terms that are easily confused. While there are similarities, they are quite different things.

This resource sets out to explain what each of them is and, most importantly, how, when they are used in tandem, technology teams and the entire organization gain huge benefits.

Value Stream Mapping

To begin, a **value stream** is anything that delivers a product or a service. Value stream mapping is a lean tool that helps teams visually collaborate on a value stream. It's an exercise that requires a team to be in the same physical or virtual space, usually for several hours at a time, to map out the value stream as a series of steps or activities.

Value stream mapping is not process mapping; rather it's a high-level, bird's eye view over the value stream. Thinking of a value stream as a collection of processes helps us understand the goals of value stream mapping. Value stream mapping exercises identify processes that need intervention. They are then often process-mapped as an actioned output of the exercise.

Value Stream Mapping Goals

Since value stream mapping is a lean tool, its primary goal is to remove waste from the value stream and optimize the flow of value from ideation to realization. The act of value stream mapping itself can't remove the waste, it makes the work visible. This enables the team to reach a deep understanding of how the processes in the value stream work together and quantify the time spent in each step (touch time) and waiting between steps (wait time). These measurements, combined with an understanding of what is a 'value-adding' step, highlight areas where there is a lot of waste. Knowing these areas of waste drives adaptation conversations.

Summing touch time and wait time provides a cycle time - the key metric we aim to reduce to accelerate the flow of value through the value stream and into the customer's hands. Adding more metrics around the people involved in each process allows us to understand the capacity and resource costs involved. This includes the percentage of their time used and the percentage of rework needed between each process. We need to be careful here though, as we don't seek to necessarily reduce costs; what we want is to inject more capacity.

Value Stream Mapping in Technology Teams

Most enterprise organizations today are on the path to becoming technology organizations. This global digital transformation and disruption drives organizations to change their ways of working from traditional, large batch, **waterfall** led approaches to **agile**, incremental and continuous ways of working, supported extensively by automation.

Using value stream mapping with technology-driven value streams throws light on what we gain from changing the ways of working.

It highlights lengthy cycle times to realize value as well as the problems with operational silos and handoffs to other teams.

Value Stream Management

Three key organizational characteristics drive the need for value stream management.

1. Most organizations are complex and made up of many value streams that often interact with each other. This creates dependencies on things like shared services and resources.

2. Digital transformation and disruption are driving a transition from waterfall to agile ways of working. As part of this we are continuously searching for improvement opportunities that require us to be able to easily measure and gain insights from what we are doing.

3. Governance guides us to do the right things and make them auditable, and as regulations continue to increase, so does the cost of compliance.

Value Stream Management:

- Is the process and automation that enables people to manage dependencies across a complex organization
- Provides insights into how to break those dependencies to optimize flow
- Supports the transition to agile ways of working
- Enables automated, continuous compliance and auditability

Industry analyst, Forrester, describes value stream management as:

“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.”

- Forrester Research

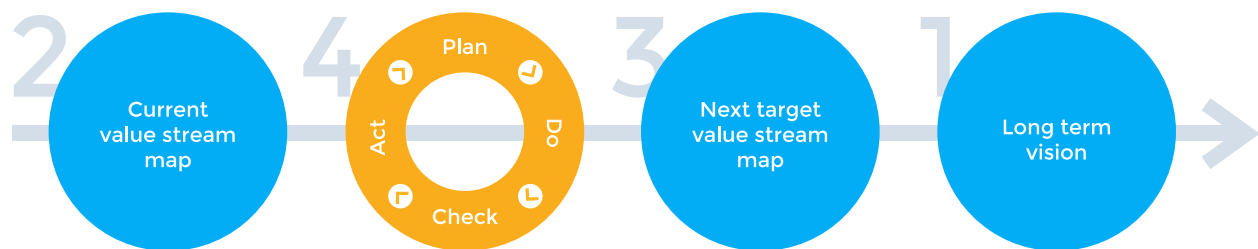
Value Stream Management Goals

Value stream management and mapping share the same core goal: the optimization of value cycle time from idea to realization - getting what the customer wants and needs to them as fast as possible (without compromising quality). Automating the process of managing the value stream offers several capabilities:

- Real-time, transparent data for continuous inspection
- Insights into the data to inform adaption
- Built-in governance and auditability

Using Value Stream Mapping and Management Together

A common mistake with value stream mapping is to do it once, probably have an output as a transformation or evolution roadmap or product backlog, and then never revisit it. Value stream mapping should be a continuous discipline in teams, ideally, something inspected every sprint review if teams are using this way of working. It is recommended to use the improvement kata to guide experiments for Product Backlog Items (PBI) as in the diagram below:



But, understandably, value stream mapping is not frequently used this way. Typically when a team first maps their value stream, that team will be made up of people from several separate teams. This is expensive and difficult to coordinate, particularly since a key requisite of the value stream mapping exercise is to include people with the authority to make and act on decisions that may significantly change existing company policy concerning funding, resourcing or approval gates. However, the gains they make in that discussion in terms of understanding one another are inestimable. But the lack of real data to support those conversations is the key problem we are trying to tackle here.

This is where value stream management platforms come in. A tenet of the way of working we are evolving towards is 'telemetry everywhere' - to be **data-driven** rather than opinion-driven. A key challenge to delivering on this is that the tools with the data are frequently fragmented, dispersed and operationally siloed. We need them to be integrated so that we can trace the value as it flows through the stream, much as we do when we map a value stream. But building integrations is hard and costly in terms of time and difficult when teams are also fragmented, dispersed and even isolated. Manually building these integrations also generates technical debt in the form of maintenance overhead and vendor lock-in.

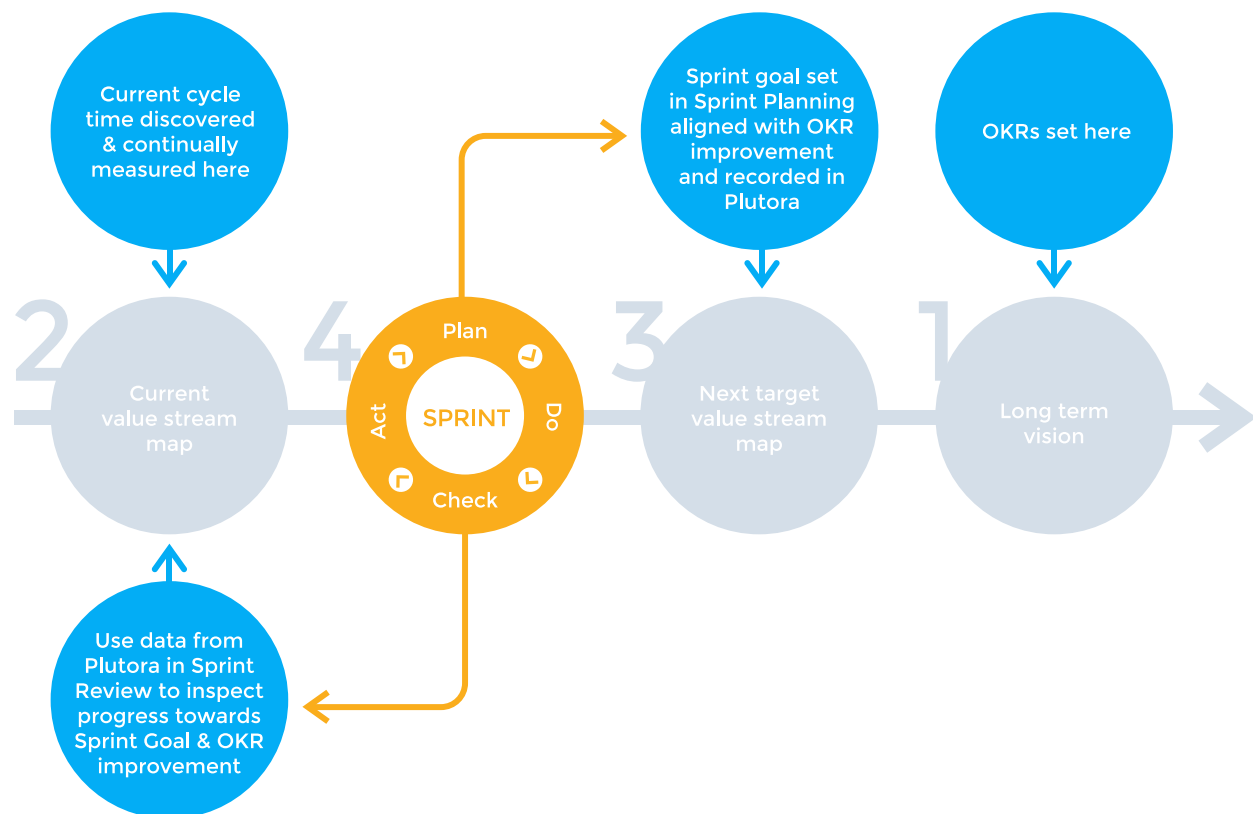
Value stream management platforms, such as **Plutora**, integrate the parts of our **DevOps toolchain** or **CICD pipeline** and the rest of the tools used in the value stream too.

Now we can automate the collection of the data we generate in value stream mapping; the touch time, the wait time and the cycle times.

We have real data to work with and it's automatically generated so we can inspect it easily and regularly (every sprint review for example) and see the impact of the experiments or interventions we make.

Plutora means Scrum or sprint-based teams can collaborate, using the improvement kata to:

- Ensure **OKRs** are in line with the long term vision
- Have a real-time and detailed view of the current state, map and cycle time (along with history)
- Set sprint goals in sprint planning that align with the next target state and the OKRs
- Define experiments (using the Deming PDCA cycle) which are automatically measured, so that they can be inspected in the sprint



In summary, it's best to use value stream mapping and management together. Starting with value stream mapping helps the team understand the detail and appreciate and empathize with each others' roles and pressures throughout the value flow and focuses their minds on the customer. Taking this initial map and producing an automated, transparent, real-time version allows the team to continuously inspect and adapt. Plutora's value stream management platform also provides:

- Business Intelligence
- Governance and continuous compliance
- Workflow automation and orchestration for:
 - Release and Pipeline Management
 - Environment Management

Managing Multiple Value Streams with Dependencies

In an ideal world, one value stream would equal one product which would be one release train. However, this is rarely the case in an enterprise, particularly when they are in transition from one (waterfall) state to the other (agile) state. Here organizational and systemic dependencies are common. There are several emerging frameworks for managing **agile at scale**, yet few are proven to be effective at managing these dependencies, let alone breaking them which is what needs to happen to have true autonomy at the team level.

Release and Pipeline Management

Plutora understands and manages the complexity of release processes and tracks releases from initial planning through realization across the value stream and DevOps toolchain and final deployment into production. If organizations are constraining their release cadence as a result of having centralized release teams coordinating release schedules through calendars, Plutora not only provides the data showing the cost of this to the value stream, it also provides a solution to support decentralization of their process. Additionally, it highlights and then can manage and measure inter-team and system dependencies whilst the teams work towards breaking them and eliminating this flavor of technical debt.

Environment Management

Similarly, many organizations have teams that share environments, particularly for testing. The management and scheduling of these environments is an administrative headache. Plutora consolidates an end-to-end view of configuration, allocation, and bookings that optimizes flow through the value streams that share these resources. Again, transparency into the value flow provides visibility into delays and dependencies that can then be tackled by the teams as they evolve towards organizational and system autonomy to accelerate value delivery.

Governance and Continuous Compliance

Regulations never go away. Quite the opposite in fact - across industries they continue to rise worldwide. Teams and organizations need to be confident they are doing what they should and will pass any audits with ease - without being slowed down by central teams or processes. By using Plutora to ensure that all workflows are compliant, teams and organizations can automate away risk. They can also automate auditability, removing the pain from what is frequently an onerous and time-consuming task.

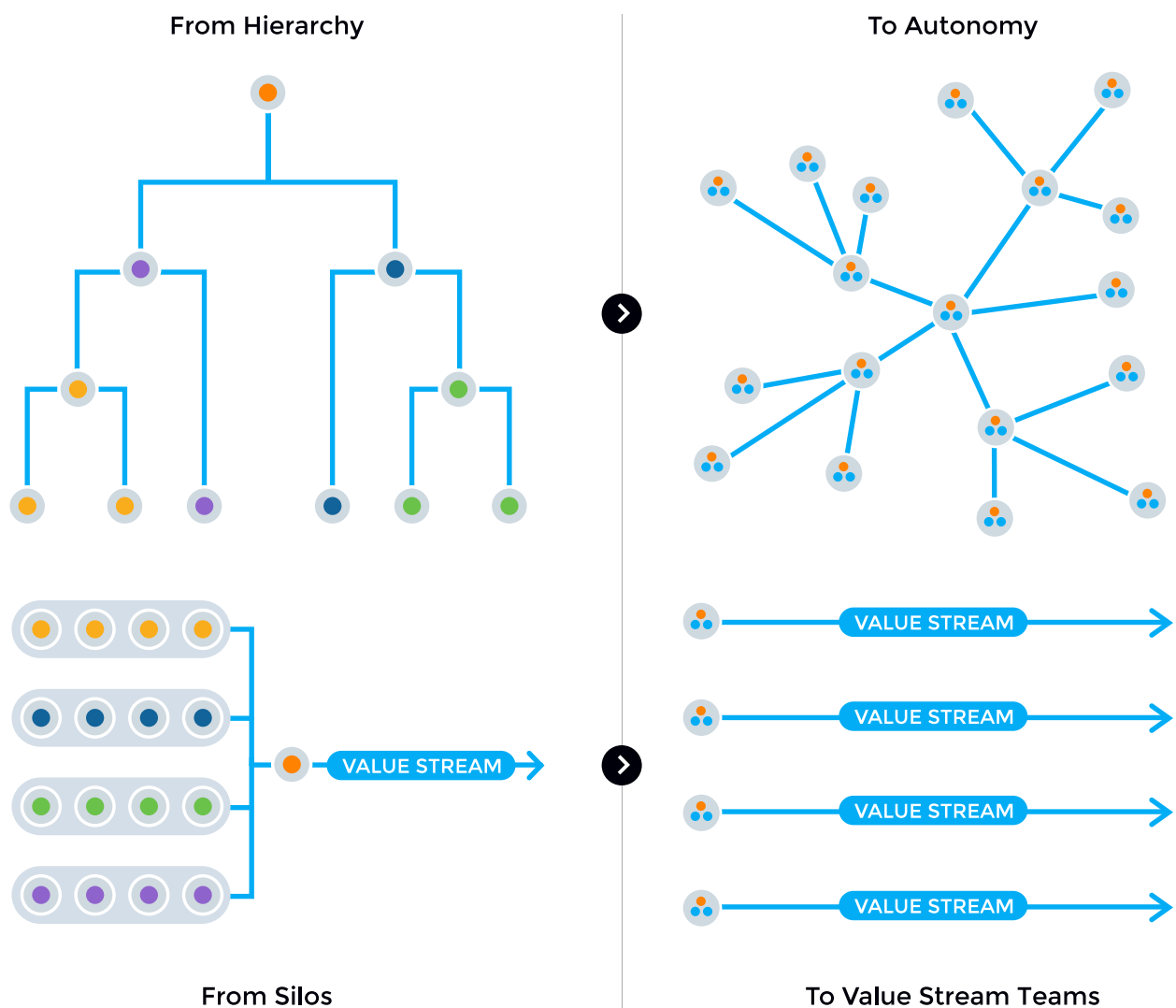
Monoliths and Microservices

Remember that a value stream is anything that delivers a product or a service. It can be tempting, and is often a habit, to use the descriptor 'applications' synonymously with value streams. The problem with this is that it's becoming increasingly difficult to define the boundaries of an application; sometimes a single application may be used by many others, and increasingly a **microservice** may be referenced by several products. Value stream-centric thinking addresses this ambiguity on the delivery of a product or service, not just the technology architecture.

Organizing Around Value Streams

Transitioning from waterfall ways of working to agile ways of working often requires change at an organizational design level. The aim is to

migrate away from silo-based models that create handoffs, queues and waste to a model of small, empowered and autonomous teams. Resistance to organizational change is frequently high in large and complex enterprises since the status quo is around hierarchy, power, and status. Many organizations also experience frequent large scale **organizational change** programs that seem to reap very little gain, leading to change apathy and demotivation across many individuals.



Moving from command and control mechanisms to a leadership team that distributes authority and focuses on discovery, particularly around impediment removal, requires belief, trust and courage - and proof. Proof, that builds trust and confidence, comes in the form of data. People evolve to become data-driven, not opinion-driven. As organizations evolve towards value stream centric organizational designs, the ability to measure the impact of the change is essential to its success.

VALUE STREAM	
Mapping	Management
Human-centric, storytelling and narrative	Tools-centric, science and insights
Opinion driven: qualitative	Data-driven: quantitative
Visual collaboration	Visibility and continuous inspection
Walls and post-it notes	Integrations, dashboards and AI
Build an initial business case	Ongoing incremental optimization
Highlights waste	Automates for continuous compliance

Humans and machines coexist. Using our brains together with the data our machines collect on our behalf presents an opportunity for us to reach a greater level of understanding of the world and work around us in a way we've never been able to do before.

Value stream mapping kickstarts humans to visually collaborate on our products, while transparent value stream management provides the ongoing platform to continuously inspect and adapt according to the insights it reveals to us via flow metrics.

Powerful automation means we can assure continuous compliance, reduce risk and manage dependencies when we have to. Together, these two processes deliver more and better value outcomes to our customers faster and our organization thrives.



Part 2: Building a Business Case for Value Stream Management

Traditionally, business cases have been written as large, complex documents, often backed up with a spreadsheet packed with calculations, showing potential cost savings and productivity improvements.

Value stream management requires a different approach. The approach is lean, continuous, incremental and focuses on value outcomes over cost and productivity. This follows the same approach as organizations who are transitioning towards implementing agile and DevOps principles and working with long-lived products.

Forrester Research defines value stream management as:

“A combination of people, process, and technology that maps, optimizes, visualizes, and governs business value flow through heterogeneous enterprise software delivery pipelines.”

- **Forrester Research**

For value stream management to work, an organization has to invest across people, process and tools so it's imperative that these investments are justified.

Value stream management enables teams to be truly data-driven. It drives value stream thinking which seeks to optimize the flow of work from idea to value realization, accelerating the time to value.

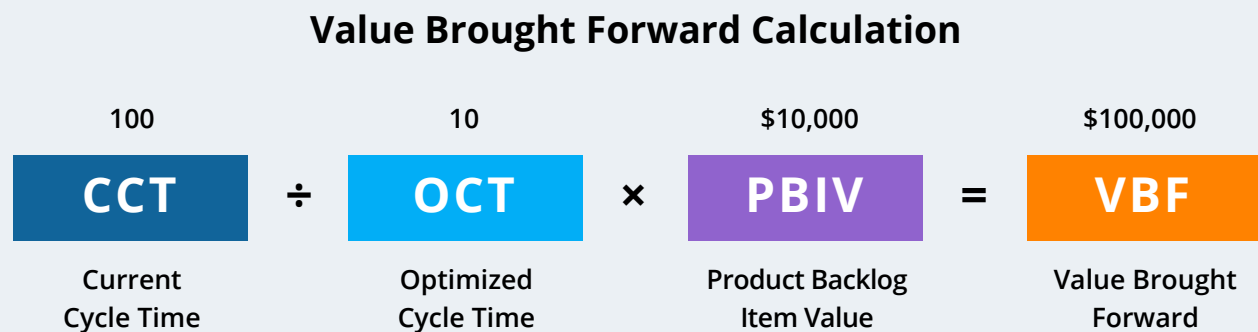
Value stream thinking aims to put valuable improvements into customers' hands as fast as possible. Optimizing the value streams (including their associated products or services) that teams are working with also means an organization can realize value outcomes as early as possible.

Calculating Value Flow

The primary metrics for value stream management are:

1. Cycle time or time to value
2. Value per Product Backlog Item (PBI)

If the current cycle time can be improved from **100 days** to **10 days** and the average PBI value is **\$10,000**, the following calculation can be used:



This means that 10 times the amount of value can be realized in the same period of time when the cycle time is reduced by a factor of 90%.

We begin to see why it's important to measure continuously and incrementally over time; this is different from traditional models where business cases for projects require significant upfront investment to deliver potential huge payback later.

Value stream management platforms (VSMPs) provide a way to monitor and measure team and organizational progress and improvements over extended periods of time. By automating these metrics for every team and providing insights into where bottlenecks are and where waste is, these platforms guide interventions for improvement. To achieve this, they integrate with DevOps toolchains to provide traceability of PBIs from idea to value realization, allowing for the calculation of cycle time or value flow.

Cost:Value Ratio

Measuring value is complex and can be done using a number of impact factors; NPS/eNPS, app reviews, customer experience, shopping cart health, conversions and click throughs for example. However, calculating the revenue of each value stream is the simplest approach for the purpose of a business case and also allows us to calculate the cost:value ratio.

Using value stream revenue also empowers team autonomy by effectively acting as their own profit and loss (P&L) statements. This focus on value streams works with the type of capacity or continuous funding models we look to when we aim for an autonomous, **multi-functional team** model in agile and DevOps ways of working.

To calculate the cost of the value stream we take into account a number of factors related to the cost of the team:

- Monthly salary of all team members
- Monthly subscription fees of all tools in the DevOps toolchain
- Monthly cloud and infrastructure spend
- Monthly spend on learning (technical, personal and leadership development)
- Monthly micro-bonus allocations
- Any other fixed overheads (office, stationery, equipment etc)















Again, we can see why it's important to measure this continuously (for example, inspecting every sprint) and incrementally.

DevOps Metrics

There are four other indicators that can be used alongside the calculation of value brought forward that are proven by the **DORA Accelerate State of DevOps Reports** to correlate with organizational, operational, and software delivery performance:

- Deployment frequency
- Lead time (from code commit to live in production)
- MTTR (mean time to restore service)
- Change fail rate

Optimizing these metrics will have a direct effect on the value flow metric and real-time collection of these can also be automated by VSMPs, like Plutora, as per the examples shown in this table:

Metric	Deployment Frequency	Lead Time	Change Fail Rate	MTTR	Cycle Time
Tool Categories Involved	CI Server and/or ARA	Backlog & CI Tool	CI Tool or ARA	Service Desk	Value Stream Management
Example Tools	  	  	  	   	

Team Metrics

Because value stream management requires a different, more agile approach to building a business case, teams will need to break down all costs into smaller increments. An example of a cost that will need to undergo this treatment are bonuses, which are typically a yearly expense determined by long-view OKRs. But to be structured in a more agile way, teams should consider adopting continuous, peer-driven microbonus schemes. This coincides with leadership evolving their roles from management and administration to coaching teams to use measurements to discover, learn and make improvements that optimize the value stream.

Measuring Value Outcomes

In time, every team should aim to have a method of measuring the actual cost:value ratio of every PBI that goes live. This requires a number of activities:

1. Estimating the anticipated value in the original record of the PBI (e.g. in Jira)
2. Traceability of the item through the value stream
3. Linking the deployment of the PBI to business monitoring (e.g. in New Relic)
4. Reporting the value impact of the item

VSMPs integrate and provide insights into the DevOps toolchain from idea to value realization so they can report on the actual versus estimated value outcomes of a PBI.



Metrics in a Business Case

Now you need to know how much it will cost you to implement value stream management at a team and global level, and you will still need a Return on Investment number. VSMPs give teams insight into the flow of their value stream and enable them to make decisions around interventions accordingly. For the initial experiment, write this as a hypothesis, for example:

“We, the mobile app value stream team, believe that if we implement Plutora in June and inspect the flow metrics every sprint and make interventions accordingly, then by December we will have accelerated the value flow by 30%. Since we have an average PBIV of \$10,000 and deploy 120 PBIs to live every 6 months, this will represent \$360,000 of accelerated value outcomes or revenue. Our initial investment in Value Stream Management for this team will be \$100,000 so our return is 360% over 6 months.”

Return on Investment Calculation

$$\left(\left(\begin{array}{c} \$10,000 \times 120 \\ \text{PBIV} \\ \text{Product Backlog Item Value} \end{array} \times \begin{array}{c} 30\% \\ \text{OCTI\%} \\ \text{Optimized Cycle Time Improvement \%} \end{array} = \begin{array}{c} \$360,000 \\ \text{VBF} \\ \text{Value Brought Forward} \end{array} \right) \div \begin{array}{c} \$100,000 \\ \text{VSMC} \\ \text{Value Stream Management Cost} \end{array} \right) \times 100 = \begin{array}{c} 360\% \\ \text{ROI} \\ \text{Return on Investment} \end{array}$$

Your value stream management cost should include ALL costs associated with the initial experiment: software subscription, implementation consulting and support and learning/training.

Once you have completed your initial experiment, you need to decide what to do next. Take a look at **Part 4: Implementing Value Stream Management** for guidance here.

Your Value Stream Management Business Canvas

Now you know what you want to do, you'll need to document it and share it so that you can gain approval from whomever you need to take the next step as well as buy-in from the people in your team. You'll also use this as the basis for inspection and adaptation as you incrementally iterate.

On the next page you'll find an example lean canvas for a value stream management business case that you can refer to. You can download the template for your own use [here](#).

Problem	Solution	Unique Value Proposition	Key Tools	Next Steps
<ul style="list-style-type: none">Lengthy cycle times to realize valueOperational silos & handoffs between teamsComplex dependencies on shared resourcesIncreasing regulation & cost of complianceDifficult transition to agile ways of working	Value stream management combined with your DevOps toolchain	Value Stream Management <ul style="list-style-type: none">Data-driven insightsAutomated key metricsReal-time dashboardDevOps toolchain integrationTraceability through pipelineIntervention guidanceVisibility access teamTransparency for leadershipAligned with OKRsAccelerated value flowInspection & adaptation opportunitySupports release & test environment	<ul style="list-style-type: none">JiraGitLabServiceNowNew RelicPlutora	<ul style="list-style-type: none">Agree to proceedSign off from sponsorUpdate teamFormally engage PlutoraImplement PlutoraTrain teamComplete tools integrationConfirm baseline metricsInspect in sprint planningAdaptInspectAdaptDec: check experiment results & actRoll out to further x number of teams
	Key Metrics		Value Flow	
	<ul style="list-style-type: none">Cycle timeDeployment frequencyLead timeMTTRChange fail rate		<ul style="list-style-type: none">Current: 50 daysFuture: 40 daysROI: 320% over 6 months	
ROI		Project Cost:Value Ratio		
We, the mobile app value stream team, believe that if we implement Plutora in June and inspect the flow metrics every sprint and make interventions accordingly, then by December we will have accelerated the value flow by 20%. Since we have an average PBIV of \$20,000 and deploy 20 PBIs to live every sprint, this will represent \$160,000 of accelerated value outcomes or revenue.		<ul style="list-style-type: none">Current: \$1,256,000:\$1,804,567Future: \$1,256,000:\$2,345,009 <p>Note: This is based on us being able to optimize the flow of value through our pipelines and realize it sooner. These are monthly figures. Value is product revenue.</p>		



Part 3: How to Do Value Stream Mapping

This section will take you through a step-by-step guide for conducting a value stream mapping exercise.

Value stream mapping is a tool that originated in the manufacturing industry through the lean movement and is designed to remove waste and optimize the flow of value through visual collaboration.

Step 1: Choose Your Value Stream

A value stream is anything that delivers a product or a service. For the purposes of this step-by-step guide we will be focusing on technology-driven value streams that use software products and platforms to deliver value to your customers.

A value stream is comprised of a number of steps, activities or processes defined from the point where an idea is conceived to where the value is received by the customer.

Step 2: Gather Your Value Stream Mapping Team

Unless you have highly sophisticated autonomous teams and systems focused around product already, your value stream mapping team will likely be composed of individuals from **several different teams** in your organizational design. You will need to gather the people responsible for having the idea, validating and evolving the idea, building, testing and deploying the idea. You will need to have representation from people involved in **change and release management**, security and governance, risk, and compliance (GRC).

You want a mix of people in the meeting - those with the authority to make fundamental changes to business processes like funding models, organizational design and approval processes. You also need deep

understanding of what happens on a day to day basis for the steps in your value stream. Typically, you'll want 10 - 15 people on your value stream mapping team. You may find that it's the first time this particular group of people have been in a meeting together.

Step 3: Get a Space

And a date. You will need several hours across two sessions for the value stream mapping exercise. In the first session you will be mapping the current state of your value stream; in the second session you will be mapping the future state.

If you are working in a physical space, the room needs a screen or projector and it needs to be large enough to fit the group so members can move around easily. It will also need a large wall that it's okay for you to stick paper to. You'll also need to make a trip to the stationery cupboard: you'll want a big stack of Post-it notes (different colors and sizes), a flip chart or roll of paper, tape, and some pens. If you are meeting virtually, you will need to use web conferencing and whiteboarding software. If these tools are new to you or your team, take time to test them out and familiarize yourself with their functionality to be able to create a value stream map.

Now it's time to invite your team.

Step 4: Prepare Some Materials

You'll likely want some slides to share with the team that explain what value stream mapping is about. You'll also want to think about how you are going to guide the exercise and how you are going to document the outputs. Suggested slides would be:

1. The purpose of value stream mapping
2. A step-by-step description of the exercise based on this guide
3. A framework for documenting the value stream mapping scope
4. A way of electronically recording the map that is created
5. A way of documenting the metrics that are collected as part of the exercise
6. A way of documenting the backlog of actions that will be created during the exercise

Step 5: Enlist a Colleague to Support You

You may choose to ask a colleague (probably from your value stream mapping team) to support you with this exercise as it's intense work - you will need to keep the exercise on track and adjudicate discussion. The conversations that go on will be invaluable and your team will learn a great deal about each other's work and a deep understanding of the overall value stream and its flow. However, sometimes you may need to call a halt to a discussion if it's become too detailed or circular. For the first session, since you're mapping the current value stream, you'll

also want to avoid solutionizing or commenting on the potential future value stream. You could park suggestions as a list to follow up with in the next session.

Step 6: Session One - Setting the Value Stream Mapping Scope

The day has arrived; your value stream mapping team is all together and you are going to facilitate the exercise.

Start by creating a value stream mapping scope and working through it with the team. Here are some suggestions of what you may want to include in the scope:

- The name of the value stream being mapped
- The name and roles of the value stream mapping team including:
 - The name of the facilitator(s)
 - The name of the value stream sponsor
- The dates and meeting space the value stream mapping exercise took place
- What the value stream mapping exercise intends to achieve
- Where the value stream starts
- Where the value stream ends

Step 7: Session One - Mapping the Steps in the Value Stream

It's now time to get the paper up on the wall and warm up the Post-it notes or virtually fire up your whiteboard. For in-person sessions, you'll want a series of flip-chart pages stuck to the wall about 9 feet long. It needs to be about shoulder height - you'll be adding Post-it notes to it for the next few hours so you'll want it easy to reach.

You and your team are going to start by identifying the steps in your value stream. They'll change as you work through them so you'll need to note them (in as few words as possible) on Post-it notes so that you can move them around as needed. Your wall will end up looking a bit like this, but likely with a lot more Post-it notes with words describing your steps on them.

Our mobile app value stream



Idea

Analysis

Dev

Test

Release



Step 8: Session One: Calculating Your Current Value Stream Cycle Time

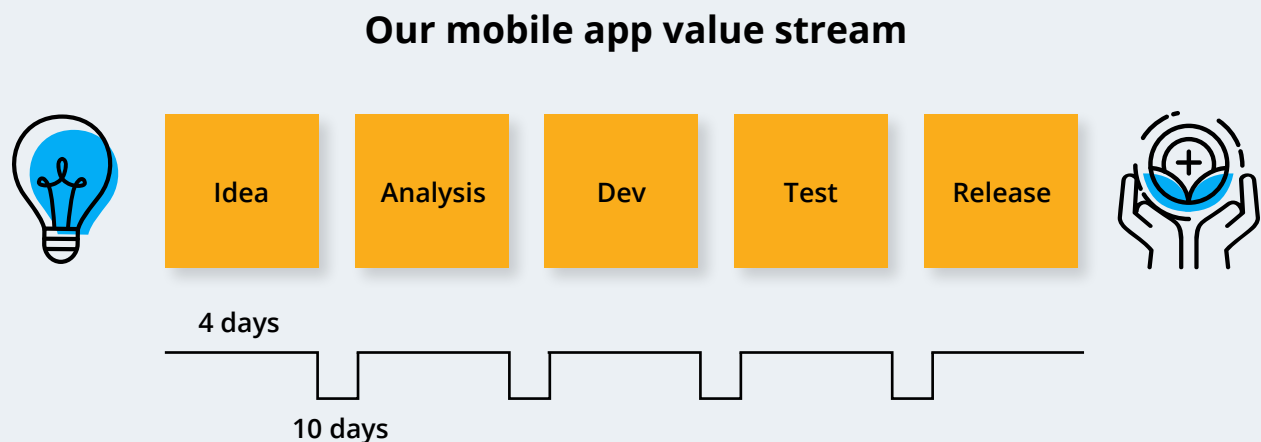
Once the value stream mapping team has agreed all the steps in the current value stream are captured, give it a final run through and make sure they are all in the right order and nothing has been missed. Then you are ready to start to calculate your cycle time.

You do this by collecting two metrics:

1. The touch time: the elapsed time that people are working on that step
2. The wait time: the elapsed time between steps

TIP: Pick one time parameter (e.g. days) and stick to it

Don't worry about how much actual time people spend working on the step or how many people work on the step - you're going to add that data later. You can draw on the paper as the team work through the steps together and you'll end up with something that looks a bit like this, with all your touch and wait times added:



When all your touch times and wait times are added, you are ready to calculate your cycle time. First sum your touch times, then sum your wait times. Then:

Total touch time + total wait time = Cycle Time

Step 9: Session One: Adding Some More Metrics

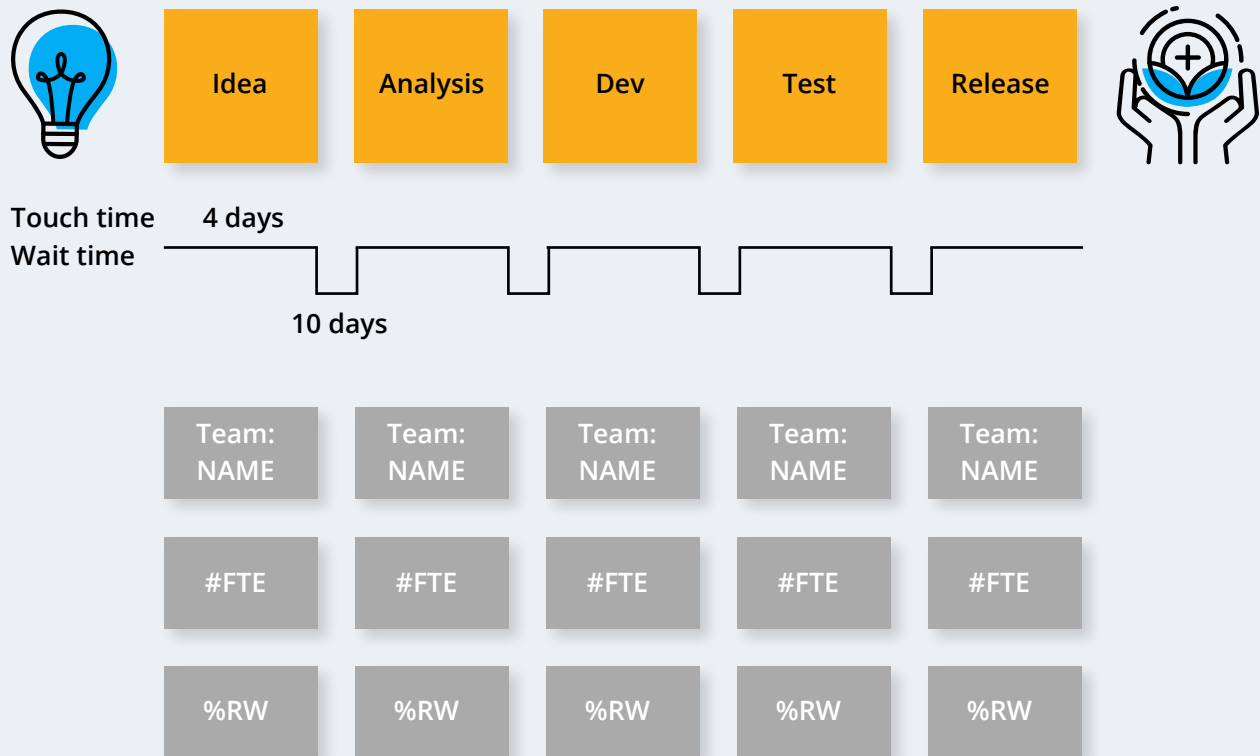
You have the key metric now - cycle time - which tells you how long it takes the team to deliver value to your customer. Now you can explore more about the resources needed for each step. Work through each step with your team and identify:

- Which teams or roles are required
- How many people are needed
- What percentage of time people are working during the elapsed touch time
- What percentage of the work has to be reworked

TIP: Highlight value adding activities in your map - ask the team to define what value adding means

Using this data you can calculate the cost of each cycle or iteration which is how much money is needed to provide a new piece of value to your customer. Your board will end up looking something like this:

Our mobile app value stream



Step 10: Session One: Reflection

Now that you've completed your current value stream map, it's time to reflect on the day that you've had together. Ask your team how they are feeling and what they have learned. People may be starting to have ideas about improvements - you can collect these on another board, ready to refer back to in the next session, when you are working on the future value stream.

Step 11: Session Two: Mapping the Future Value Stream

Start your second session with a short recap of what you learned about the current value stream and then dive into mapping your future state. It's best to start with a blank sheet/whiteboard. Now start working on what the steps look like in the future when you embrace ways of working that improve the flow through the value stream.

Start a board to make a note of the improvements the team are talking about - this will begin to form the experiments or hypotheses in your evolution backlog which is a key output of the value stream mapping exercise. Here are some examples of improvements which may come up:

- Appoint a Product Owner to manage the backlog of ideas
- Create an autonomous, multi-functional team
- Create a CI/CD pipeline including functional and non-functional testing
- Automate change and release checklists
- Manage the work in sprints

Step 12: Session Two: Calculating the Future Value Stream Cycle Time

Just as you did with the current value stream, add the touch and wait times for each step and calculate the cycle time.

Step 13: Session Two: Adding Resource Utilization Metrics

Add the metrics for the team, number of people and percentage of time worked and rework just as you did for the current value stream.

Step 14: Session Two: Compare Your Metrics

Now that you have metrics for both your current and future value streams, it's time to compare and analyze them. Ask the team questions like:

- What percentage improvements are you going to see?
- Where are the biggest improvement opportunities?
- What are the easiest things to fix?

Step 15: Session Two: Build Your Backlog

Now it's time to build the backlog of activities that need to happen in order to make the improvement happen:

- Write backlog items as experiments or hypotheses
- Each backlog item needs an owner
- And a timescale for completion

TIP: Use the improvement kata to make incremental changes without being overwhelmed at the scale of the task at hand

Step 16: Session Two: Commit to Value Stream Review Cycles

It's imperative that you don't waste the valuable work your team has just invested so much time in. You'll likely need to identify a Value Stream Manager (it could be you!) to ensure the team meets together regularly to inspect the value stream and report on improvements.

You'll want to share the work you have done too, perhaps with leadership to get their buy-in, commitment and access to budget for the changes you have proposed.

Automating the data collection and analysis moving forward will really help you - look at [Part 4: How to Implement Value Stream Management](#).





Part 4: How to Implement Value Stream Management

Here's how to start the process of implementing value stream management in your organization so you can enable continuous improvement.

Maximizing Value from Value Stream Mapping

Once a team has completed their value stream mapping exercise, they will have a number of assets:

- A map of the current value stream with flow metrics
- A map of the future, desired value stream with projected flow improvements
- A backlog of experiments for adaptive evolution

We want to make sure that the organization achieves the goal of delivering more and better value outcomes to customers faster. The key is to ensure that these assets are not left to languish and be wasted. Value stream management puts these assets to use and works toward that goal. Here's how to start the process of implementing value stream management in your organization so you can enable continuous improvement.

Defining and Establishing the Value Stream Teams

A group of people will have been brought together to perform the value stream mapping exercise. Typically, this group is not organizationally on the same team. Even if an established development, product or feature team completed this exercise, you will likely identify upstream and downstream team members that need to be included in the value stream team. This can include business stakeholders, product owners, GRC, security, change boards, release and IT operations.

The long-term goal is to have small, autonomous teams aligned to each value stream (product or service). However, transitioning from a highly siloed organization takes time, particularly in enterprise scale organizations where large amounts of people are involved.

Organizational redesign is difficult for many reasons: leadership frequently resists it in order to protect status and authority, while individuals resist it because of uncertainty or change apathy. Additionally, implementing new processes and culture change takes time to be adopted and may require upskilling and other learning.

While in transition, it's possible to create a virtual value stream team with representation from each step in the value stream. The individuals in these teams may not be fully dedicated to the value stream yet, as the organization moves from a siloed model towards long-lived product centric teams.

Individuals need to be committed to the goal of the value stream team; which is to accelerate delivery of value to the customer without compromising quality or compliance. They must be enabled to spend time collaborating with other members of the team on a regular basis.

Appointing a Value Stream Manager

This is a role, not a job title and does not necessarily need to be full time. Roles are more important than job titles because as teams transition towards autonomous teams, individuals are increasingly required to perform multiple functions in order to optimize flow. People need to be good at more than one discipline: developers learn security, system administrators learn to script, testers learn to automate and so on. The ultimate goal of the Value Stream Manager is to safely accelerate the value stream flow. The responsibilities of the Value Stream Manager are:

- Establishing improvement metrics and their discovery
- Continuously inspecting the current value stream map with the value stream team
- Ensuring value stream improvement backlog items are completed
- Evangelizing value stream thinking to teams within the value stream

Anyone can take on the Value Stream Manager role; however, there may be people with skill sets that are particularly well-suited for the role. Those who already have an end-to-end view of the value stream are good candidates. Examples include Release Managers who work closely with the teams, Product Owners who carefully monitor the whole product lifecycle or Scrum Masters that are particularly interested in team metrics.

Identifying Initial Value Stream Metrics

During the initial value stream mapping exercise, you will have identified several important metrics:

- Cycle Time = Touch time + wait time
- Cycle Cost = The summation of people, percentage of time spent working, and percentage that must be reworked at each step

Cycle time in this instance is measured from the moment there is an idea to when that idea is realized in the customers' hands. This is a key flow metric. There are four other DevOps metrics as established by **Google's DevOps Research and Assessment** team that indicate overall technology and organizational performance:











1. Deployment frequency
2. Lead time (from code commit to deploy)
3. Change fail rate
4. Mean Time to Recovery (MTTR)

As organizations and teams establish agile and DevOps practices, they gain the capacity to collect these four metrics. However, the metrics exist in different tools, which makes it difficult to present them together and there are no standard frameworks or patterns yet available to do this. Additionally, these tools are rarely integrated, making work visibility and traceability across value streams very difficult.

These operational silos lead to an onerous manual process of collecting the metrics needed to calculate the cycle time.

It should also be noted that these metrics are indicators that correlate to performance, but they do not provide insights into the flow of value. They describe outcomes: how fast the throughput of value is and how stable the value stream is. But these metrics alone don't tell us how we got there or how to improve.

Using Value Stream Management Platforms (VSMPs) like Plutora provides automated integration between tools, delivering end-to-end value stream visibility. It also provides data and insights around the waste and delays in the value stream to guide interventions and experiments for improvements. The table below describes which tools can provide which data.

Metric	Deployment Frequency	Lead / Cycle Time	Change Fail Rate	MTTR	Flow
Tool Categories Involved	CI Server	Backlog & CI Server	CI Server or ARA	Service Desk	Backlog, CI Server & Service Desk
Example Tools	  GitLab	 Jira   GitLab	 	 now  cherwell	 PLUTORA

Implementing a VSMP will provide organizations with real-time visibility into their value stream that they can leverage for continuous inspection. It also means that leadership has an organization-wide performance heat map to help them direct their efforts to support discovery and improvement in teams.

As organizations continue to transition from a waterfall, silo-based working model to one built on autonomous teams delivering value incrementally, a key change is needed to financial models. As people focus on their product or value stream, project-based funding ceases to make sense. This requires more than a change in technology because technology teams have traditionally been order takers, and the purse strings held by 'the business'.

However, as teams become autonomous and use VSMPs to measure their value stream, the funding model can be shifted. With VSMPs, the teams are able to measure both their cost of operating and the value that they are creating. The ratio of cost to value is essentially a profit and loss (P&L) for each team. This real-time, decentralized P&L information can drive the conversation around changing funding models to more agile ones, such as capacity-based-funding models.

The need for certain metrics will also change as the organization evolves. For example, when a team is able to deploy on demand several

times a sprint using continuous delivery, deployment frequency is a much less interesting metric. Similarly, as teams architect for resilience and practice chaos engineering and limited blast radius techniques, such as canary or blue/green deployments, the resulting little or no downtime makes MTTR less interesting. When teams have evolved to this level of capability, the remaining key metrics are flow metrics - in particular cycle time.

It is up to the Value Stream Manager and the teams to establish the initial set of key metrics that the VSMP will provide and the plan for managing the evolution of the relevance of the metrics over time.

Updating Metrics Inspection Practices

As organizations transition to agile ways of working, they discover opportunities for inspection in agile meetings such as sprint planning, daily Scrums or stand-ups, and sprint retrospectives. At these events, the team can use the VSMP to gain insights into their progress and possible improvements at a local and global level.

When organizations are practicing agile at scale, VSMPs can also be used to explore patterns common to multiple teams and these impediments can be specifically targeted.

Opportunities for inspection in agile

Sprint Planning	Scrum	Sprint Review	Sprint Retrospective
Setting the Sprint Goal may involve reviewing team performance and OKRs. Seeing how factors such as change fail rate are trending over time helps prioritize features	The team gets together daily to review progress and impediments - VSMP provides real-time data on delays in the value stream helping to guide the ScrumMaster and provide evidence to help remove impediments	VSMPs show how the team performed against their Sprint Goal and whether they are meeting their target cycle times. It can also provide insights into value realized	Having real and immediate data that shows where PBIs may have been stuck provides insights into delays. This tells the teams where they may need to adjust their processes to improve

Note: If the team is practicing Kanban without Scrum, they can still use VSMP metrics and insights whenever they perform an inspection.

Managing the Value Stream Improvement Backlog

In the transition from waterfall to agile, it's rare for teams to have true autonomy at first. This means it's not obvious which backlogs to assign the improvements identified in the initial value stream mapping exercise. The Value Stream Manager is accountable for ensuring the value stream management backlog items are complete, so they may have to collaborate with other teams and their backlogs to achieve this.

This becomes simpler as teams become more autonomous and PBI's relate only to their own backlog, or **agile at scale** techniques can be used to coordinate improvements globally or across multiple teams.

The Value Stream Manager uses the data from the VSMP to work with the team to understand insights and create, refine, and prioritize backlog items accordingly.

Sharing VSMP Best Practices Throughout the Organization

To train teams on the VSMP, the Value Stream Manager should pull together some onboarding and performance support content to share on **Confluence** or the organization's wiki. In addition, rather than sending whole teams away for days of training, an incremental, microlearning approach is recommended. This is where the Scrum events like planning, review and retrospectives can be used for the team to work together to review how to leverage the VSMP and build up knowledge sprint by sprint.

And in order to develop and share the organization's VSMP best practices, Value Stream Managers should work with other Value Stream Managers in a formalized group, perhaps as a Chapter, Guild, or Community of Practice. Time and resources should be allocated for them to share insights that have resulted in improvements.



Part 5: The DevOps Toolchain and Value Cycle

There's a global movement toward agile and DevOps. These ways of working encourage teams to think in value streams and organize around long-lived products to deliver features incrementally.

The Evolution of DevOps Toolchains

The culture and people need to embrace a mindset of autonomy and experimentation, where processes become lighter-weight and peer-oriented. We need to automate waste (waiting time, handoffs, onerous manual labour) out of the equation wherever possible. Automation in this context means building a DevOps toolchain.

The establishment of a **DevOps toolchain** in an enterprise follows a fairly well established pattern. Steps may vary in order, or occur in parallel but generally include:

Step	Development Team Activity	Tool Category
1	Implement a version control tool and probably a code artifact repository to eliminate confusion and align on a single source of truth	Version control Artifact repository
2	Use a Product Backlog	Product Backlog
3	Implement a wiki-based information sharing tool	Wiki collaboration
4	Start to automate builds, likely using a Continuous Integration (CI) server	CI server
5	Practice CI or evolve towards it. Where CI is when 'Developers commit code at least daily to trunk or master and perform automated unit, integration, and user acceptance tests	CI server Unit testing

Step	Development Team Activity	Tool Category
6	Since the desire to be more agile and work in smaller batches is driving these changes in behavior, there should be a conversation going on about short-lived feature branches and how to manage these, potentially using the CI tool	CI server
7	The definition of CI includes three types of tests: most teams start with automating unit tests	Unit testing
8	Begin integration tests afterward	Integration testing
9	Lastly, begin user acceptance tests	User acceptance testing
10	Start thinking about non-functional tests (NFTs), like performance and security tests around this time	Performance testing Security testing
11	Begin having conversations with other teams about preferred tools and practicing Continuous Delivery (always having software in a releasable state). They may also start to use release and deployment automation	CICD server Release automation Deployment automation
12	Tools integration should become part of the conversation as teams automate tasks between tools and seek visibility and traceability into the value flow	API integration
13	IT Operations should be working more closely with the development team now, particularly if the team has been thinking about those NFTs and working towards the 'we build it, we run it' mantra. As a result, ChatOps may be implemented, and the teams may start looking to the Service Desk for data on change, release and incident management	ChatOps Service desk

Step	Development Team Activity	Tool Category
14	Early forays into performance testing may be revisited, particularly as the team becomes more practiced in delivering small batch incremental change. They are also becoming more interested in what's happening in live	Application performance management, telemetry and observability
15	They may be asked by leadership for more detailed metrics on their performance, or they may want to inspect and adapt themselves	Dashboards
16	The team now wants to consciously optimize from idea to value realization and gain insights. Leadership wants a global view into cycle time and value flow	Value stream management

Tools Standardization

If you have been through the process of evolving a DevOps toolchain you'll know that at **Step 11** tensions can surface. Tools standardization is a conversation that triggers conflict as people have their preferences and can be quite attached to them. Large enterprises will have multiple teams frequently doing similar things without knowledge of what the other teams are doing, or knowledge but no influence. Teams' agile and DevOps capability will also vary across the organization. This combination of factors frequently leads to tool proliferation which only halts when a central group takes notice, identifies it as a problem and coordinates an enterprise-wide response. The problems with tool proliferation (having multiple tools from a single category) boils down

1. **Licence management:** some tools are free or low cost and easily accessed online at the point of entry but become increasingly expensive when more users are added or additional support is needed
2. **Learning:** even when teams are arranged around long-lived products or value streams, people will still move from team to team and needing to learn new tools when they do so slows the whole team and organization down
3. **Visibility of work:** as the organization evolves further into agile and DevOps ways of working, they likely will want to make the work more visible outside of the teams. Asking leadership to use multiple product backlog tools, for example, can create a barrier to entry and slow the evolution down

Another approach can be that a central group is responsible for enabling the teams across an enterprise in DevOps tooling and they provide access to their chosen tools, or even a DevOps toolchain as a service. Remember, that in these ways of working we are trying to distribute authority, partly to eliminate delays caused by waiting for handoffs to and from other teams. Teams operating as a central group here need to walk a fine line between dictating the tools and enabling the teams to have autonomy.

Some organizations like to use the term 'sensible defaults.' This means that if a new team asks the organization or central team (e.g. IT Operations, the DevOps toolchain platform team or the DevOps Center of Excellence) what tools they should use, they are guided to the tools that are most prevalent across the organization. These tools will have the largest user base and will

therefore be the most easily supported.

Of course, not all tools can be standardized. Some, like testing tools, are very platform specific. But others, in particular the product backlog tools, are very important to standardize across an organization.

CICD or DevOps Toolchain?

These terms are often used interchangeably, however, the main difference is that a **CICD** (Continuous Integration and Continuous Delivery) toolchain tends to stop at production (or before, in cases where production rights continue to be held separately). The DevOps toolchain, on the other hand, should extend end to end across the value stream, from idea to value realization and include service desk and monitoring capabilities.

Release and Deployment Automation

Definitions of **release** and **deployment** vary from organization to organization and it can be confusing. For example, many organizations build a release, deploy it through the route to live (or test and staging environments) and then release it to production. Generally, when we talk about release, we're talking about the manual activities that need to be coordinated to prepare for the deployment; for example, open port, and flag with the change team. It's important that every organization ensures that all teams have the same understanding of these terms.

It's notable that the closer teams get to **continuous delivery** - that is, an automated process that takes features from code commit to ready for production - the less they need to manage the release process.

Note, that the common definition of Continuous Delivery (CD) is to always have software in a releasable state. Pushing to live requires a human to perform a manual task.

Continuous Deployment (CDep) means that when a new feature passes all of the tests in the CI/CD pipeline, it is automatically deployed into production. Team and enterprises' appetite for CDep vary considerably in the current market, depending on their current capability and the particular market they are in.

For instance, if an organization is not releasing frequently, their appetite to perform continuous deployment is likely to be low until they increase their cadence. Some industries, like insurance, that are highly regulated may also be more reticent to automate into production in this manner, although other highly regulated industries, notably retail, are more likely to continuously deploy because competition has demanded the ability to respond to feedback at a higher pace.

The DevOps Tools Landscape

The tools market has changed considerably during the decade that DevOps principles have emerged and established. At first, the **Software Development Lifecycle** (SDLC) and **Application Lifecycle Management** (ALM) tools markets were ruled by a handful of the large software vendors alongside open source alternatives, but then a host of new offerings flooded into all categories. These offerings were more accessible to teams who, in many cases, no longer had to wait for procurement to negotiate a huge multi-year deal with the vendor. Instead, they could simply begin using them, often paying for monthly subscriptions via a credit card.

Though new tools do still appear, the market has settled somewhat in recent years, with category leaders appearing, market consolidation accelerating and some older tools disappearing off the map completely. Here are examples of tools frequently seen in enterprise DevOps toolchains (some tools span multiple categories):

DevOps Toolchain Tool Category	Example Tools
Version control	GitHub, GitLab, BitBucket
Product backlog	Jira, Azure DevOps
Wiki Information Sharing	Confluence, Teams
Artifact repository	Nexus, Artifactory

DevOps Toolchain Tool Category	Example Tools
CI server	Jenkins, GitLab, Azure DevOps
Unit testing	JUnit, NUnit
Integration testing	Jasmine
User acceptance testing	Selenium
Performance testing	LoadRunner
Security testing	Nexus Lifecycle, ZAPP
Release automation	Plutora, XebiaLabs, Cloudbees
Deployment automation	Plutora, UrbanCode
API Integration	-
ChatOps	Slack
Service desk	ServiceNow, Cherwell, JiraSD
Application Performance Management	Dynatrace, AppDynamics, New Relic
Dashboards	Hygieia
Value Stream Management	Plutora

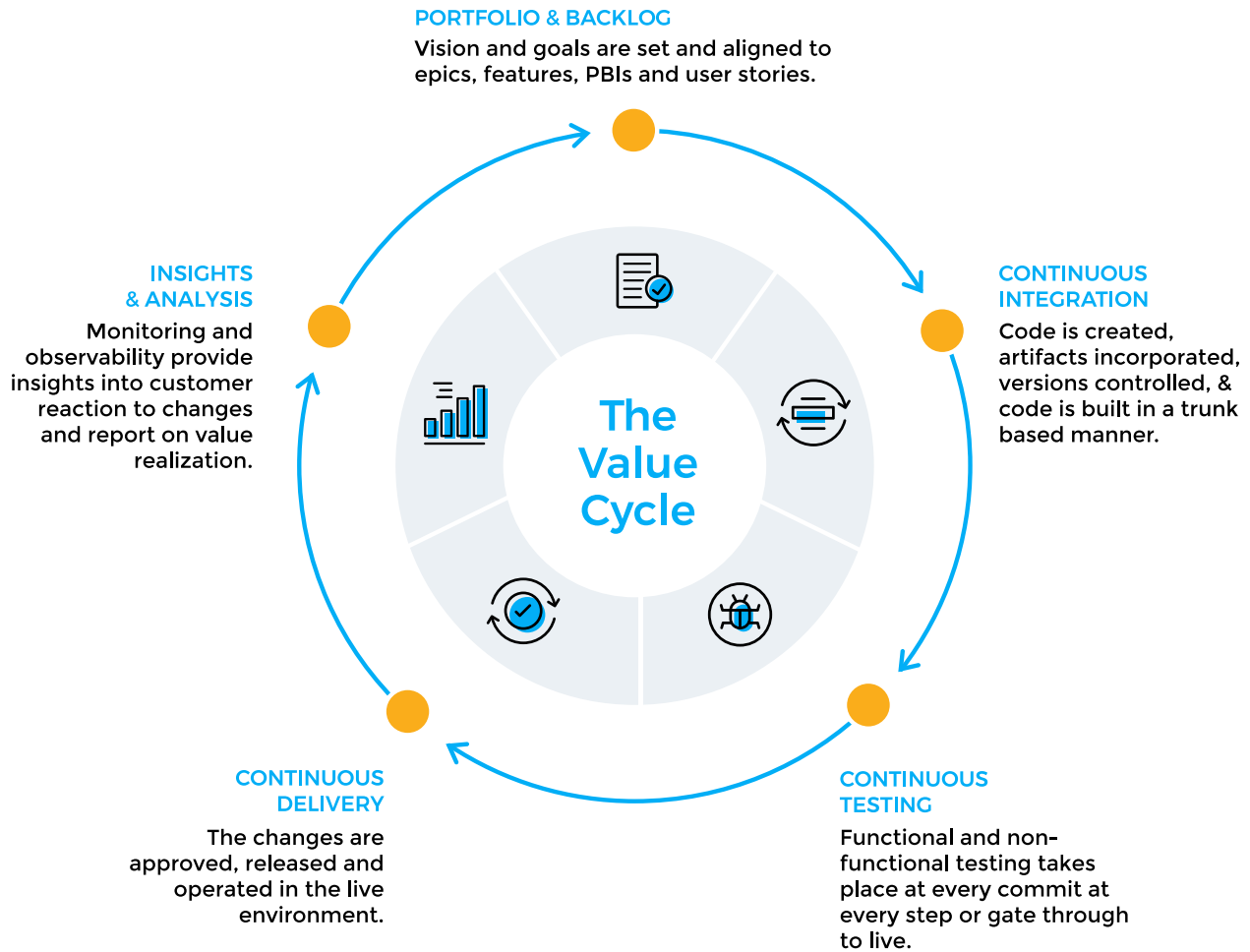
The Value Cycle and Value Stream Management

As long-lived product teams become increasingly proficient at agile and DevOps ways of working and expand their automation coverage, they experience improvement in the four DevOps metrics discussed in **Part 6: Value Stream Management Metrics**. Their focus moves from these performance and productivity metrics to a desire to understand how value flows through the value stream that is their product or service. They want to know the cycle time of every feature and whether value outcomes have been realized in the customers' hands.

VSMPs help achieve this by integrating with the DevOps toolchain to provide:

- Metrics for cycle time and other flow metrics
- Data driven insights into where delays exist in the value stream
- Global metrics for value flow across an enterprise
- Measurements of an enterprise's progress towards its agile and DevOps goals

Traditionally, the SDLC has been represented by a linear process that does not reflect the continuous nature of the incremental, highly accelerated lifecycle that DevOps seeks.

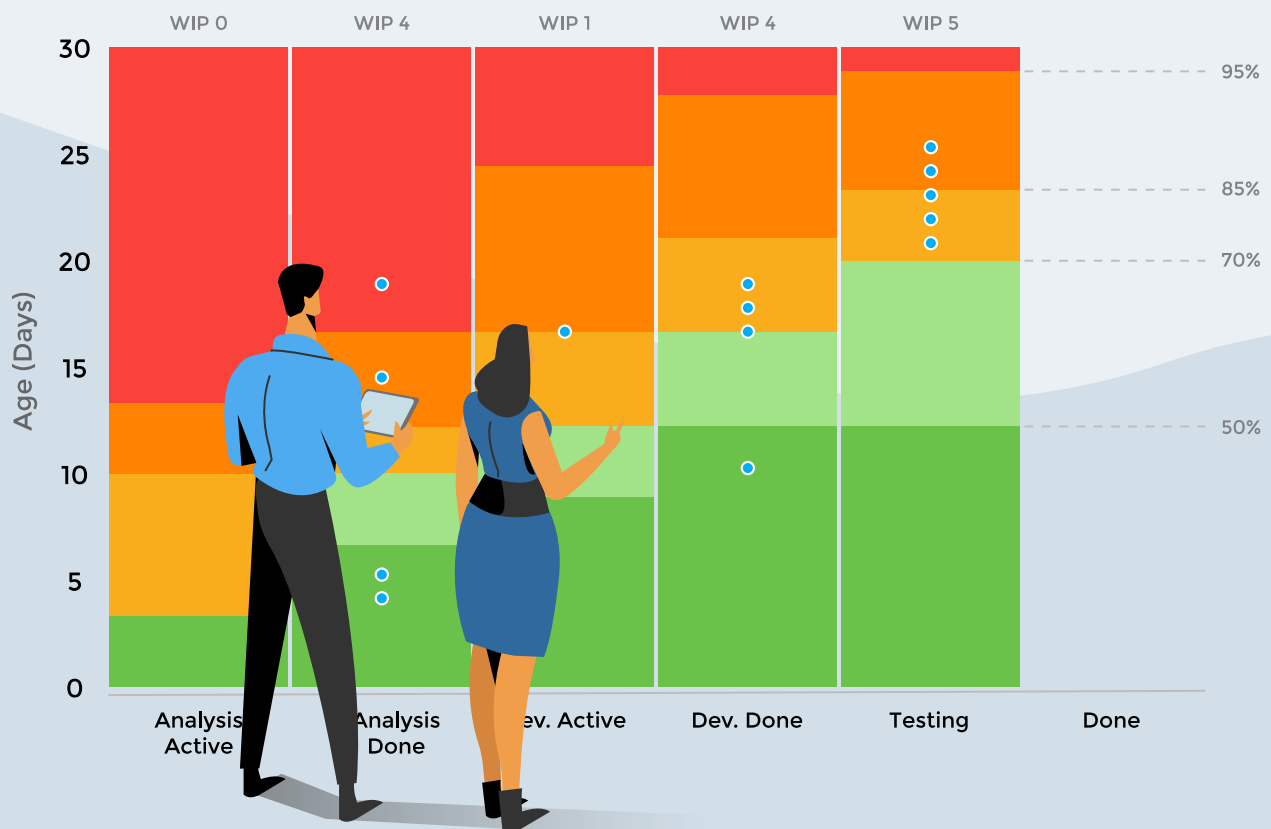


Thinking of the value stream as a circular cycle helps when sharing with others how the steps fit together and where steps need to be accelerated to achieve improvement.

It also presents an opportunity to overlay the DevOps toolchain on the value stream to show how the pieces fit together, as the table on the next page summarizes:

	Portfolio & Backlog	Continuous Integration	Continuous Testing	Continuous Delivery	Insights & Analysis
Activity	Vision and goals are set and aligned to epics, features, PBIs and user stories.	Code is created, artifacts incorporated, versions controlled, code is built in a trunk-based manner.	Functional and non-functional testing takes place at every commit and at every step or gate through route to live.	The changes are approved, released and operated in the live environment.	Monitoring and observability provide insights into customer reaction to changes and report on value realization.
Tool Categories	Product backlog Wiki Collaboration ChatOps	Version control Artifact repository CI/CD server	Unit testing Integration testing User acceptance testing Performance testing	Service desk ChatOps Release automation Deployment automation	Application performance management
Example tools	Aha!, Jira, Confluence, Teams, Slack	Nexus, Artifactory, GitHub, GitLab, Jenkins, CircleCI	JUnit, NUnit, Jasmine, Selenium, LoadRunner	Slack, ServiceNow, Cherwell, UrbanCode Deploy, Octopus Deploy	AppDynamics, Dynatrace, NewRelic

VSMPs enables teams to understand their value stream and DevOps toolchain from idea to value realization. Having these insights drives data-driven improvement which ultimately results in optimized organizational performance.





Part 6: Value Stream Management Metrics

In the DevOps world, the **CALMS** acronym is frequently used to succinctly explain the scope of DevOps.

It signifies:

- Culture
- Automation
- Lean
- Measurement
- Sharing

The Importance of Being Measured

Peter Drucker said, “If you can’t measure it, you can’t improve it.”

While improvements could happen without being measured, if they aren’t measured, we’ll never be sure if they did improve. Being sure that improvements happened is fundamental to creating a **DevOps culture**. The aim is to be data-driven, not opinion driven. Measurements are data, evidence, and proof. And proof builds trust. When trust is high, friction is low and value streams flow at high speed and with low cost.

DevOps is ultimately about optimizing the flow of value from idea to realization.

Worldwide organizations are transitioning from traditional, project-centric ways of working to agile and DevOps approaches centered around product or service value streams. These value streams’ goal is to accelerate the delivery of value outcomes to customers. All organizations want to show the impact that their transformations are having and being able to measure the transformation progress is essential. This is the underpinning of the value stream management metrics that matter. As teams automate their software development and operations activities, these types of measurements can be automated. The learnings and improvements uncovered should be shared so that local improvements become global improvements.

Shifting the Ownership of Metrics

Most teams and organizations find measurement very difficult in part because they are coming from a project-centric way of working. In the traditional waterfall approach, the focus is on delivery milestones, huge business cases, budgets and resource optimization. Projects are typically driven by productivity improvements over incremental value outcomes.

Culturally, organizations have operated in a command and control manner where metrics are set by the top and cascaded down to teams. These metrics become targets upon which individuals can have their performance praised or punished.

In agile and DevOps ways of working, authority is distributed to the teams, who are empowered to use their own metrics to discover and learn the improvements they can make.

This requires that leadership change their approach from management to servant-coaches. It also means that metrics are no longer given from a manager to an individual, but that teams measure themselves. However, moving from one model to another requires a significant amount of work culturally. People do not suddenly feel empowered to work in a new manner because they are told that they are empowered. This takes time because teams never previously had the authority over their own accountability.

It also takes time for leadership to change their own behavior from telling people what to do to asking them what they want to improve.

It's hard for an organization to stop targeting teams from the top and instead enabling teams to choose, measure and manage their own metrics. And sometimes teams just aren't ready to. Additionally, since many of these transformation initiatives have required significant leadership support and investment, leaders are anxious to show that there has been real improvement and return on that investment. It's a fine balance between telling the teams what metrics they must report on and encouraging the teams to find the metrics that they can use to discover improvements and adapt accordingly.

Leaders also need to understand that teams operate at different speeds and capability and it's their job to understand the different pressures and challenges each team faces. They need to focus on team and global metrics and not the management of targets on individuals.

Getting Started with Metrics

Most organizations suffer from having too many metrics (with too much noise) or having none at all. Best practice is to have a handful for each team, a subset of which are common to all teams, allowing leadership to have insights into overall organizational or global performance.

A popular set of metrics are the four key DevOps metrics as defined by the **Accelerate State of DevOps Reports**:

Metric	Definition
Deployment frequency	How often code is deployed to production in a team's value stream (product or service)
Lead time	How long it takes from code committed to code successfully running in production for the team's value stream
MTTR	How long it takes to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment) within the team's value stream
Change fail rate	The percentage of changes to production that result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch) within the team's value stream

The **Accelerated State of DevOps Reports** also give teams insights into organizational performance worldwide. It's worth noting that the metrics that matter to teams change over time as their capability and performance evolves. For example, elite performers' deployment frequency is on demand. Once a team has the capability to deploy on demand, this is no longer a metric that is useful for indicating improvement. So mastery of metrics means having the ability to inspect and adapt the metrics, as well as the underlying value stream.

As teams evolve their automation capability, they will begin to find that these metrics are available in the tools that they use. Fragmentation

of tools across an organization can make this more challenging, particularly if metrics collection is being driven by a central team. Technical debt is also frequently a significant obstacle, either by preventing teams from collecting metrics because they are constantly firefighting or by tightly-coupled **architectures** that make it hard to decouple value streams.

Advanced Value Stream Metrics

Once a team has got a handle on their base set of metrics, they're going to become interested in how value flows. The first metric the team should think about is cycle time. They may already have performed a value stream mapping exercise to build some human and opinion-driven data around their cycle time. Lead time and cycle time are metrics that frequently confuse people since they have many different definitions; the important thing is that the organization chooses a definition that works for them and ensures it's communicated and understood across the whole organization.

Flow Metrics

In the four key DevOps metrics, lead time is defined as the time from code commit to production. But cycle time should be measured from the time an item appears in the backlog - or even earlier if it has been recorded in a portfolio management tool. Teams might also find it useful to measure from sprint planning and from dev starts also.

Once the four key DevOps metrics are mastered, it's time to tackle the flow metrics described in the following table - a flow item is a unit of work that matters to the business: feature, defect, risk or debt.

Metric	Indicates	Definition
Flow Velocity	Whether value delivery is accelerating	Flow Velocity is the number of Flow Items of each type completed over a particular period of time - also referred to as throughput
Flow Distribution	Helps prioritize specific types of work during specific time frames in order to meet a desired business outcome.	Flow Distribution measures the ratio of the four Flow Items completed over a particular window of time
Flow Time	Can identify when Time to Value is getting longer	Flow Time measures the time it takes for Flow Items to go from 'work start' to 'work complete', including both active and wait times
Flow Efficiency	Can identify when waste is increasing or decreasing in your processes.	Flow Efficiency is the ratio of active time vs. wait time out of the total Flow Time
Flow Load	Monitors over and under-utilization of value streams, which can lead to reduced productivity	Flow Load measures the number of Flow Items currently in progress (active or waiting) within a particular value stream

Making Metrics Visible

Transparency is key to a high trust and dynamic learning environment. Automating the collection of these metrics makes them available to teams to inspect and adapt accordingly. The following table explains which category of tool may be used to access these metrics and some examples of tools where the information may be found. Flow metrics require visibility across all tool categories in the value stream or DevOps toolchain, which requires integration between the tools and intelligence to provide insights. For this reason a value stream management solution is required.

Metric	Tool Category	Example Tools
Deployment Frequency	CICD/ARA	Jenkins, GitLab, UrbanCode, Octopus Deploy
Lead Time	Backlog	Jira, Azure DevOps
MTTR	Service Desk	ServiceNow, Cherwell, Jira Service Desk
Change Fail Rate	Value Stream Management	Plutora
Flow Velocity	Value Stream Management	Plutora
Flow Distribution	Value Stream Management	Plutora
Flow Time	Value Stream Management	Plutora
Flow Efficiency	Value Stream Management	Plutora
Flow Load	Value Stream Management	Plutora

In order to obtain these metrics, someone needs to do some work. This is often a Scrum Master, but may also be a Value Stream Manager. The Value Stream Manager can be accountable for collecting and automating the metrics, and helping the team to continuously inspect them and adapt accordingly. The metrics will themselves evolve over time. As discussed above, deployment frequency becomes a much less useful metric for a team when they are deploying on demand. They will likely become much more interested in flow optimization and the realization of actual value.

An investment of time and effort and tooling will be needed to automate these metrics but the payoff is the insights that allow for interventions that accelerate the flow of value to the customer. Not having this data means teams are blind to what's happening in their value stream so it's best to prioritize this over other work as a foundational capability.

This may mean that the Product Owner will need to build time into sprints to make these improvements. But this is a common conversation during agile transitions as it is about prioritizing process improvement over new features. Some teams find it helpful to put standard percentages of sprints into categories like technical debt, learning, and unplanned work. This helps to manage process improvement and set the expectation around capacity available for new features i.e. flow distribution.

Continuous Adaptation

Once a Value Stream Manager has their team's metrics automated, and the team continuously inspects the metrics, they need to practice how to improve. Working in an agile manner means incrementally improving using experimentation. Using the improvement kata helps teams make improvements smaller and view them as experiments as they Plan, Do, Check and Act. These experiments are Product Backlog Items (PBIs).

Use the sprint review (flow optimization is a sprint goal) and retrospectives to inspect flow improvements and document them. Over time you will be able to show leadership how your value stream flow has improved - this is a proxy metric for how the organization's entire agile and DevOps transformation is progressing. This is how team metrics support global metrics.

Make sure this information is visible to other teams and other Value Stream Managers. This is a good time to set up a Value Stream Manager community of practice if you haven't already or don't have a Value Stream Manager chapter. You could gamify this and have scoreboards that show how different value streams are performing - as long as this won't cause ill-will between teams. When the team sees improvements or hits impediments, make sure the associated stories and learnings are also documented and shared with the rest of the organization. This is how local improvements lead to global improvements.



Choose your own adventure - this approach is not prescriptive. Value stream management will provide insights that are unique to your value stream, your team and your organization - the critical part is the opportunity for improvement it provides.



Part 7: The Value Stream Glossary

Learn everything you need to know about value streams, value stream management, and the jargon therein.

Term	Definition
Adaptation	The three pillars of Scrum are Transparency, Inspection and Adaptation. These three pillars uphold every implementation of empirical process control. If an inspector determines that one or more aspects of a process deviate outside acceptable limits, and that the resulting product will be unacceptable, the process or the material being processed must be adjusted. An adjustment must be made as soon as possible to minimize further deviation.
Agile	A lightweight framework that promotes iterative development and incremental delivery using self-organizing cross-functional teams.
AI - Artificial Intelligence	AI is intelligence demonstrated by machines that are programmed to learn and problem-solve as opposed to natural intelligence demonstrated by humans.
ARA - Application Release Automation	ARA involves using tools, scripts, or products to achieve the consistent and repeatable process of packaging and deploying an update, feature, or application from development to production.
Artifact	An artifact is any deliverable or documentation associated with a project that helps to describe the function, architecture, and design of the software being developed.
Artifact Repository	A tool designed to store, version, and deploy artifacts for software builds.
Auditability	The ease with which audit data can be substantiated by tracing it to source documents, and the extent to which auditors can rely on pre-verified and monitored control processes.
Autonomy	Autonomy refers to the ability to make decisions and execute them independently, without having to wait on or defer to others, usually someone higher up in the hierarchy.
Backlog	A collection of requirements expressed as epics, features or user stories (Product Backlog Items or PBIs), ideally in the form of experiments or hypotheses.

Term	Definition
Blue/Green Deployment	A technique that reduces downtime and risk by running two identical production environments called Blue and Green. At any time, only one of the environments is live. As a new version of software is developed, deployment and the final stage of testing takes place in the environment that is not live. Once the new version is deployed and fully tested, the router switches incoming traffic so that the testing server becomes the live server.
Bottleneck	A bottleneck is a step in the value stream where the value stream is limited by the volume the step can handle.
Capability	Accelerate recommends use of the term 'capability' over 'maturity' when considering team performance in order to minimize situations where DevOps is considered 'done' and recognize cross team variation.
Capacity	The volume of value a team can produce for their value stream in a defined period of time, e.g. 2 week sprint.
Capacity Funding	A team is funded as a long-lived value stream, rather than on a project by project basis. This term is interchangeable in this context with continuous funding.
Canary Deployment, Release or Test	Derived from the phrase "canary in a coal mine", the canary release functions in a similar manner. It is used to gradually introduce a software release to a small subgroup of users, making sure that it is safe and functional before being rolled out to the general populace.
CCT	Current Cycle Time: a metric used in value stream mapping and management to summarize the time taken for an idea to realize value in the current state. It's the sum of touch/processing time and idle/wait time.
Change Fail Rate	The percentage of changes to production that result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch) for the product or service (value stream) that a team works with.

Term	Definition
Change Management	The context for value stream mapping is the process in place for a system change to be approved to be deployed into production, rather than organizational change management.
Chaos Engineering	The discipline of experimenting on a software system in production in order to build confidence in the system's capability to withstand turbulent and unexpected conditions.
Chapter	In the Spotify model, a chapter is a group of people from multiple teams with a shared discipline, e.g. testing automation.
ChatOps	Integration between a group chat platform and DevOps tooling creating single pane of glass where teams can collaborate with each other and interact with the tools
CI Server	Continuous Integration server: a tool that automates the build process allowing developers to commit code daily to trunk and automate unit, integration and user acceptance tests.
CICD	Continuous Integration/Continuous Delivery: the practice of continuous integration allows for continuous delivery.
CICD Pipeline	The toolchain that supports the processes that allow a value stream team to perform continuous delivery i.e. version/source control, build server, automated testing.
CICD Server	A tool that automates continuous integration and continuous delivery.
Command and Control	When an organization's activities are led by a senior leader; authority is not distributed and teams lack autonomy.
Community of Practice (CoP)	An informal and voluntary group of people in an organization with a common interest in a particular practice e.g. value stream management.

Term	Definition
Complete and Accurate	A quality metric expressed as a % used in value stream mapping to measure the degree to which work from an upstream step is determined by the downstream step to be complete and accurate (or error free). This is a mirror metric (same, but opposite) to 'Rework'.
Constraint	Anything that limits a value stream from achieving higher performance relative to its goal.
Continuous Compliance	An ongoing process of proactive risk management that delivers predictable, transparent, and cost-effective results to meet GRC and cybersecurity goals, typically underpinned with automation.
Continuous Delivery	Always having software in a releasable state, but a manual button is used to push new code into production.
Continuous Deployment	When new features pass the tests in CI successfully, they are automatically deployed into production.
Continuous Funding	A team is funded as a long-lived value stream, rather than on a project by project basis. This term is interchangeable in this context with capacity funding.
Continuous Integration	Developers commit code at least daily to trunk where the build and unit integration and user acceptance tests are automated.
Cost:Value Ratio	The cost of the value stream team compared to the value (typically revenue) outcomes generated measured in the same units, e.g. currency over the same period of time e.g. month.
Customer Experience	Customer experience, also known as CX, is a customer's holistic perception of their experience with an organization or value stream.
Customer Journey Mapping	A technique that maps the steps a customer follows when they go through a particular business process in a value stream (product or service).
Cycle Time	A metric used in value stream mapping and management to summarize the time taken for an idea to realize value. It's the sum of touch/processing time and idle/wait time.

Term	Definition
Daily Scrum	A Scrum event where the value stream team meets daily for fifteen minutes to inspect progress, typically following the format of team members reporting on what they did the day, before, what they plan to do that day and highlighting any impediments to the ScrumMaster.
Dashboard	A summary of key metrics on a single screen, typically derived from multiple tool sources and tailored according to the value stream role of the viewers. Dashboards may or may not supply insights into the data. The metrics are typically not connected from tool to tool.
Data Driven	When a value stream team actively collects (ideally automatically and real-time) data from their systems for inspection and adaptation. The antonym is opinion driven.
Decentralization	When authority is distributed so that value stream teams are able to make decisions autonomously.
Deming Cycle	A continuous improvement model originating from W. Edwards Deming's work in lean manufacturing in the 1950's that follows a Plan, Do, Check, Act (PDCA) cycle.
Dependency	An organizational or system step that limits a value stream team from autonomously optimizing flow.
Deployment Automation	Tools (or scripts) that automate the manual steps taken to deploy new features through a route to live and into production.
Deployment Frequency	How often code is deployed to production for the value stream that a team works with.
Deployment Process	The steps taken to deploy new features for a value stream.
DevOps	A portmanteau of 'development' and 'operations' describing the need for these two technical teams to collaborate more effectively that led to a global movement around culture, automation and measurement that has evolved to focus on value stream optimization.

Term	Definition
DevOps Pipeline	The steps from having an idea to the value of that idea being available in production for a user to realize and be measured (from Dev to Ops and the focus is on automating these steps where possible.)
DevOps Toolchain	The tools in the DevOps pipeline that provide automation from having an idea to the value of that idea being available in production for a user to realize and be measured.
Distributed Authority	A cultural model where leadership empowers autonomous value stream teams to have authority to make their own decisions.
DevOps Metrics	Four key metrics identified by DevOps Research and Assessment as part of the annual State of DevOps Reports that correlate with technology and organizational performance, namely: deployment frequency, lead time, MTTR and change fail rate.
eNPS	Employee Net Promoter Score: how likely an employee is to recommend their employer to friends and family.
Environment Management	How a value stream team accesses or makes available the infrastructure (on-premise, cloud etc) that their product or service requires for development, testing and to run in production.
Evolution Backlog	A collection of requirements, features or issues identified for the improvement of the value stream's delivery capabilities.
Experiment	A requirement, feature or issue expressed as a hypothesis requiring inspection on completion and adaptation as needed.
Flow	How value travels through a value stream from idea to realization.
Flow Distribution	The ratio of the four Flow Items (feature, defect, risk, debt) completed over a particular window of time.
Flow Item	Units of work that matter to a business: features, defects, debt, and risk.

Term	Definition
Flow Efficiency	Flow Efficiency is the ratio of active time vs. wait time out of the total Flow Time.
Flow Load	Flow Load measures the number of Flow Items currently in progress (active or waiting) within a particular value stream.
Flow Metrics	A set of metrics value stream teams use to inspect and adapt flow including: velocity, distribution, time, efficiency and load.
Flow Time	The time it takes for Flow Items to go from 'work start' to 'work complete', including both active and wait times. See also 'Cycle Time' and 'Lead Time'.
Flow Velocity	The number of flow Items of each type completed over a particular period of time - also referred to as throughput.
Global Metric	A metric that is common to all value stream teams in an organization and inspected and adapted accordingly.
Governance	How value stream teams ensure their product or service complies with local and global regulatory requirements.
GRC	Governance, risk and compliance (GRC) refers to a strategy for managing an organization's overall governance, enterprise risk management and compliance with regulations.
Guild	A term from the Spotify model used to describe a Community of Practice (CoP).
Hypothesis	A supposition or proposed explanation made on the basis of limited evidence as a starting point for further investigation as part of an experiment.
Ideation	The act or process of having ideas, including documenting and planning development activities.
Idle Time	The time between two steps or activities in a value stream when nothing is happening. Also known as 'Wait Time'.

Term	Definition
Impediment	Something, sometimes a dependency, that is causing a problem with the flow in a value stream.
Improvement Kata	A continuous improvement habit for daily work from the lean movement that has four steps: <ol style="list-style-type: none"> 1. Set the long term goal 2. Understand the current state 3. Define the next target state 4. PDCA (see Deming Cycle) between the states.
Incremental	When work is approached by working on and delivering small pieces consecutively, i.e. small batch.
Insights	Trends and indicators, often using AI, typically seen as part of a dashboard that highlight problems with flow for the value stream team to inspect and adapt.
Inspection	The three pillars of Scrum are Transparency, Inspection and Adaptation. These three pillars uphold every implementation of empirical process control. The value stream team inspects the flow, product, platform, processes, people aspects, practices and continuous improvements.
Integration Testing	The testing of a component or module of code to ensure it integrates correctly with other components or modules of code.
Intervention	An activity intended to improve value stream flow.
Kanban	A work visualization technique originating in the lean movement in Japan in the 1950's. Translation is 'visual board' or 'visual cards'.
Large Batch	When a lot of things are bundled together for a single processing event, for example: lots of requirements for a large project or lots of features for a large release.

Term	Definition
Lead Time	Definitions vary from team to team and organization to organization. The definition from the State of DevOps Reports is: How long it takes from code committed to code successfully running in production for the product or service (value stream) that a team works with. However, teams may also want to consider time from the idea first being documented (cycle time), first going into the backlog, first going into the sprint backlog and from development starting.
Lean	A movement originating in manufacturing with the goal of removing waste from pipeline flow.
Limited Blast Radius	An architectural and engineering approach to ensuring that failure is limited to a small set of users e.g. canary deployment.
Long Lived Products	An organizational approach and way of working that moves focus away from projects and onto products (value streams) where teams can focus on incremental improvement.
Maturity	See ' Capability '.
Metrics	A standard of measurement, in the context of value stream thinking used primarily to monitor the improvement of flow and therefore an organization's transition to DevOps and value stream centric ways of working.
Microbonus	An alternative to annual bonus schemes where small bonuses are given frequently, typically peer to peer via a microbonus platform.
Microservice	A small loosely coupled service which can be developed, deployed, and maintained independently. Each microservice is a discrete process that provides a unique business capability. Typically these are connected by API and frequently run in containers.
Monolith	A traditional software design architecture that is single-tiered, with tightly coupled components. This means that each component and its associated components must be present in order for code to be executed or compiled.

Term	Definition
MTTR	How long it takes to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment) for the product or service (value stream) that a team works with.
Multi-functional Team	A small team of people (typically 8-10) that are dedicated to a single value stream and contain a number of cross-discipline capabilities e.g. analysis, dev, test, cloud, security. Individuals within the team may also be multi-functional.
NPS	Net Promoter Score: how likely your customer is to recommend your organization's products or services to friends and family.
Observability	How well internal states of a system can be inferred from knowledge of its external outputs.
OCT	Optimized Cycle Time: the goal cycle time identified during a value stream mapping exercise.
OCTI	Optimized Cycle Time Improvement: the percentage by which the current cycle time improves in the current state map produced in value stream mapping, to the goal cycle time in the future state map.
OKRs	Objectives and Key Results: A framework for defining and tracking objectives and their outcomes.
Organizational Design	A human-centered approach to improving how people work together and how companies respond to change.
PBI	Product Backlog Increment: An item such as a feature, defect, or technical work that is valuable from the product owner's perspective held in the product backlog.
PBIV	Product Backlog Item Value: The estimated value of the PBI to be realized in live from an end user customer and/or business perspective.

Term	Definition
PDCA	Plan-Do-Check-Act: See ' Deming Cycle '.
Peer-Oriented or Peer-Reviewed	Where authority is distributed and teams are empowered and have autonomy, peers are used to approve work rather than leadership/management or centralized teams.
Performance Testing	A testing practice performed to determine how a system performs in terms of responsiveness and stability under a particular workload.
Pipeline Management	The act of building, maintaining and optimizing the process and automated pipeline or DevOps toolchain from idea to value realization. Part of value stream mapping.
Planning	Working together to establish the high level goals and activities, ideally mapping these, as planning becomes more granular, to the day to day work that delivers the features and improvements identified as needed.
Process Mapping	A process is a step or activity within a value stream which is a collection of processes that deliver a product or service. Process Mapping is detailed analysis of the step itself, not its relationship with the other steps in the value stream overall.
Processing Time	The elapsed time spent actively working on a step in a value stream. Also known as 'Touch Time'.
Product Backlog	The requirements planned, requested or underway for a value stream. See also Backlog .
Product Owner	The team member accountable for the product backlog, particularly for refining and prioritizing the PBIs so that the team can estimate work as they move them into development.
Production	The live environment used by customers.
Project-Centric	Organizations, teams or individuals who think and work in a project oriented manner, rather than thinking of end-to-end value streams or products.

Term	Definition
Realization	When the PBI has reached the customer and the value outcome can be measured.
Release Automation	Release automation is the tool-based orchestration of software delivery activities and resources across multiple, interdependent releases and change initiatives.
Release Pipeline	The steps needed to take a feature or release package from development to production. It can also refer to the automated steps or toolchain.
Release Process or Management	The steps and orchestration of software delivery activities and resources across multiple, interdependent releases and change initiatives.
Release Train	A form of software release schedule in which a number of distinct series of versioned software releases for multiple products are released as a number of different “trains” on a regular schedule.
Rework	A quality metric expressed as a % used in value stream mapping to measure the degree to which work from an upstream step is determined by the downstream step to be complete and accurate (or error free). This is a mirror metric (same, but opposite) to ‘Complete and Accurate’.
Return on Investment	A measurement for the gain or loss generated on an investment relative to the amount of money invested, usually expressed as a percentage.
Route to Live (RTL)	The path to production e.g. dev to test to staging to live.
Scrum	A commonly used agile framework.
Scrum Master	The member of a Scrum team accountable for impediment removal, conducting events and coaching the team and organization on agile practices.
Service Desk	A service desk is a communications center that provides a single point of contact (SPOC) between a company and its customers, employees and business partners. The purpose of a service desk is to ensure that users receive appropriate help in a timely manner.

Term	Definition
Silos	An organizational design in which teams or departments focus on their own goals instead of working toward organizational objectives. Typically, there is waste due to delays as a result of handoffs between silos.
Small Batch	When work is approached by working on and delivering small pieces consecutively, i.e. incremental.
Source Control	Source control (or version control) is a system that records changes to a file or set of files over time so that previous versions can be accessed and recalled later. It is useful for rollbacks and disaster recovery, among other things.
Spotify Model	An organizational design and agile methodology used and shared by Spotify.
Sprint	A sprint is a defined work period, usually two weeks, in which a scrum team completes a discrete unit of work.
Sprint Planning	A Scrum event where the team defines their Sprint Goal and creates a Sprint Backlog of work for the upcoming Sprint.
Sprint Retrospective	A Scrum event where the team reflects on how the last Sprint went from a process perspective and plans improvements.
Sprint Review	A Scrum event where the team reflects on how the last Sprint went according to the goal they set and the burndown of the Sprint Backlog. Frequently includes a 'show and tell'.
Takt Time	The rate at which a product must be complete in order to meet customer demand. Typically more useful in manufacturing than software delivery environments.
Team Metric	A metric defined and used by a team to drive discovery and improvement and not necessarily a global metric.
Technical Debt	Refers to the rework that must be done when speedy delivery and easy implementation is prioritized over a better, but usually slower or more expensive, approach.

Term	Definition
Technology Organization or Software Organization	An organization that has recognized that although technology or software delivery has not been their core business in the past, digital disruption now means their ability to compete requires their strategic elevation of their technology and software delivery capabilities.
Telemetry	The collection of measurements or other data at remote or inaccessible points and their automatic transmission to receiving equipment for monitoring.
Test Automation	Allows testers to reuse tests in a repeatable process, thereby eliminating time-consuming and repetitive tasks. Test automation is crucial for agile and DevOps environments.
The Business	In technology teams, a term typically used to mean 'the rest of the organization', frequently where requirements are seen to originate and budget is held.
Time to Value (TTV)	The amount of time it takes a customer to realize value from a new feature (or product or service). This is the same as cycle time.
Touch Time	The elapsed time spent actively working on a step in a value stream. Also known as 'Processing Time'.
Transparency	The three pillars of Scrum are Transparency, Inspection and Adaptation. These three pillars uphold every implementation of empirical process control. Significant aspects of the process must be visible to those responsible for the outcome. Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen.
Unit Testing	When individual units of source code together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Typically performed and/or automated by developers.
User Acceptance Testing (UAT)	A type of software testing that verifies that a given application works for the user. During this process, actual users test the software to make sure it behaves as expected in real-world scenarios or it can be automated.

Term	Definition
Value Adding	A step in a value stream in which actual value is created for the customer.
Value Brought Forward (VBF)	Value Brought Forward: Expressed as a percentage, this metric describes how much sooner the customer will realize the value of a new feature with the planned improvement to the value stream.
Value Chain	A term used interchangeably with value stream - more commonly used in the manufacturing industry.
Value Delivery	When a new feature is received by the customer.
Value Flow	Analysis and understanding of how value travels from idea to realization.
Value Outcome	What happens when value is realized by the customer (note: this could be negative).
Value Realization	The moment a customer uses a new feature or product and the value outcome can be realized.
Value Stream	Anything that delivers a product or service.
Value Stream Activity	A term used interchangeably with value stream step, this describes a process that occurs within a value stream, for example: approving a change or performing a test.
Value Stream Centric Thinking	A mindset that shows individuals or teams think of their product or service as an end to end value stream where possible value is perceived at ideation and realized when in the hands of a customer.
Value Stream Map	An output of a value stream mapping exercise: an artifact that visually describes the steps in a value stream and associated time, work and cost metrics. Can be a current or future state map.
Value Stream Mapping	A lean tool for visually collaborating to identify ways to remove waste from a value stream.

Term	Definition
Value Stream Mapping Scope	An output of a value stream mapping exercise: an artifact that documents the scope of the value stream, challenges, benefits and the people and team involved.
Value Stream Management	A combination of people, process, and technology that maps, optimizes, visualizes, and governs business value flow through heterogeneous enterprise software delivery pipelines.
Value Stream Management Platform (VSMP)	Technology that maps, optimizes, visualizes, and governs business value flow (including epics, stories, work items) through heterogeneous enterprise software delivery pipelines.
Value Stream Management Investment Canvas	A lean canvas individuals and teams can use to express a lightweight business case for value stream mapping.
Value Stream Manager	An individual accountable for value stream optimization.
Value Stream Sponsor	An individual with the authority and access to budget to make significant decisions and change on behalf of a value stream team where they do not have the autonomy to do this themselves.
Value Stream Step	A term used interchangeably with Value Stream Activity, this describes a process that occurs within a value stream, for example: approving a change or performing a test.
Version Control	Version control (or source control) is a system that records changes to a file or set of files over time so that previous versions can be accessed and recalled later. It is useful for rollbacks and disaster recovery, among other things.
Visibility	Similar to transparency, visibility demands that work is visible to all in order to break down silos and build trust.
Value Stream Management Cost	All costs associated implementing value stream management: software subscription, implementation consulting and support and learning/training.

Term	Definition
Wait Time	The time between two steps or activities in a value stream when nothing is happening. Also known as 'Idle Time'.
Waste	A lean term defined as any step or activity that does not add value to the customer.
Waterfall	A software development methodology where the software's requirements are gathered at the beginning of a project, then executed upon in linear fashion, with the next phase beginning only when the previous one is complete.
Ways of Working	A generic term used often when describing how organizations move from waterfall to agile ways of working. New ways of working also include: DevOps, agile service management, site reliability engineering, safety culture, Holacracy, Sociocracy and learning organizations.

Further Reading

Learn how to use value streams to navigate and accelerate the DevOps transformation of your software delivery factory, whether it be on-site, remote, or any combination of the two.

Become a software juggernaut.

What is Value Stream Management?

Learn DevOps: Enterprise DevOps at Scale

The Secret to DevOps Success: Release Management

CI/CD Tools Universe: The Ultimate Guide

Discover the power of value stream management with Plutora:

Why Plutora?

The Plutora Platform

Request a Demo



About the Author

Jeff Keyes is VP of Product at the value stream management platform, Plutora. He has spent his career writing code, designing software features and UI, running dev and test teams, consulting and evangelizing DevOps and value stream management. Outside of six years at Microsoft, he has been primarily focused on growing startup companies.