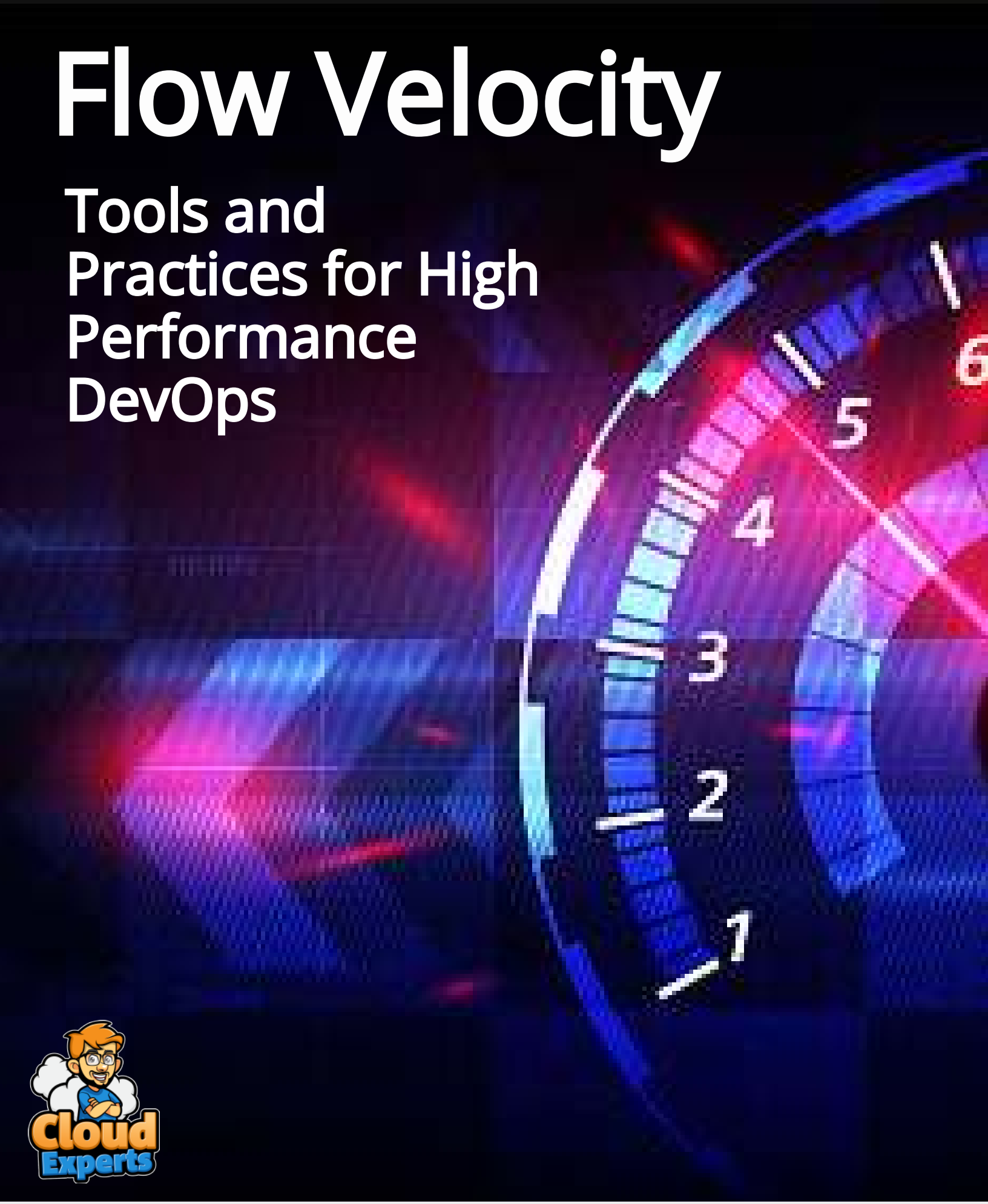
Flow Velocity

Tools and Practices for High Performance DevOps



Cloud Experts

DevOpsFlow.net

DevOps Flow - Tools and Practices for High Performance Software Development

In the Cloud era senior executives are investing into key capabilities like DevOps with the expectation this will accelerate innovation and bring new digital products to market faster.

To deliver on these goals enterprise organizations seek to deploy high quality code, faster and more frequently, but they face a complexity of organization, process and technology that can hinder this objective, slowing throughput and incurring high error rates.

Therefore a whole system DevOps methodology that addresses the organizational level is required, one that maps processes end-to-end and identifies the improvements needed to speed production across the entire software life-cycle.

DevOps Flow is a methodology for measuring the entire development lifecycle end-to-end, defining metrics and improvements that enable continuous optimization to speed the deployment of new software releases, and thus accelerate digital innovation.

Organization and Process Transformation

Often when DevOps is discussed the focus is concentrated on the technology layer, particularly the latest tools like Kubernetes, and also when considering performance it zooms in on the productivity of each individual developer.

However for many organisations the main challenges will be at the level of their team practices and organizational models.

Writing for DevOps.com Mike Vizard highlights the reality of DevOps uptake, that most adoption issues are related to organisational challenges, not technology. The top two barriers to adoption are slow processes and speed of adaptation (29%), followed by budget and funding (21%). Only 18% identified technology limitations as an issue.

As Mike writes it is the broader organisational transformation that is more challenging. 'Addressing bottlenecks' might be a function of confronting the situation of a department manager who zealously guards their turf and acts politically to resist any changes required.

Flow Metrics for Elite Level Performance

Organisations need the ability to measure the entire system – end-to-end – to understand how value flows and where it is constrained, and most importantly, to correlate those metrics with desired business outcomes. This approach allows for continuous optimization in the pursuit of delivering greater and greater value to the organisation, faster.

Executives can understand, monitor and improve the systems performance of their software development organization by quantifying it's throughput metrics. Google research identifies the metrics that define an Elite level of DevOps performance, and McKinsey defines how this can be measured and accelerated through a 'Developer Velocity Index'.

DevOps Flow - Tools and Practices for High Performance Software Development

Implementing DevOps for one team is a relatively straight forward exercise, but for enterprise organisations, they face a much larger magnitude of complexity, as they typically have multiple teams, spanning multiple geographies and suppliers, with hundreds of developers all contributing to the same software development process.

Therefore a DevOps methodology that also addresses the organizational level is required, one that maps processes end-to-end and identifies the improvements needed to speed production across this entire life-cycle.

DevOps Flow is a methodology for crafting these metrics and implementing high performance, Cloud-centric software development, based on a science developed by pioneers like Toyoto to optimize factory lines and apply this to software engineering.

It can be applied to speed the deployment of new software releases, notably identifying and removing constraints, reducing batch sizes and eliminating waste. These are all steps taken within manufacturing to increase production throughput and are improvements that software teams can adopt to achieve equivalent benefits.

DevOps Flow - Accelerating Developer Velocity Through Software Factory Best Practices

Two McKinsey articles are ideally combined when considering the challenge of growing the speed and success of your IT & DevOps teams.

In '*Welcome to the Digital Factory*' McKinsey review the new organizational models being employed by disruptive digital leaders – The 'Digital Factory'.

Complimenting this is '*How software excellence fuels business performance*', where they explain how a headline metric of 'Developer Velocity' can ensure digital investments.

The two are an ideal combination as a *Digital Factory* approach identifies new ways of working that tap insights from the world of manufacturing, applying a production line metaphor for software workflow management, that can be monitored and managed through KPI metrics under a headline of '*Developer Velocity*', so that output can be improved via the same type of process improvement regimes.

As they observe in companies like Goldman Sachs, software engineers now make up 1/4 of the total workforce. In addition to mastering the nuances of their industry, businesses today need to excel first and foremost at developing software, and their articles form a helpful implementation model for getting started.

DevOps Flow - Accelerating Developer Velocity Through Software Factory Best Practices

Digital Factories

Transformation leaders are turning to a '**Digital Factory**' approach to streamline this complexity into a production line model that yields consistently high throughput and quality, using Lean principles derived from the world of manufacturing.

In their in-depth article, McKinsey explores the dynamics and successes of the Digital Factory model, a new approach to organising digital teams, a self-organising structure vs rigid departmental hierarchies, and they describe the KPI improvements the Digital Factory enables:

"A Digital Factory is the "construction site" where change happens. It comprises dedicated, cross-functional teams that work together on change-the-business programs. They resemble factory workers in that they employ reusable tools and repeatable processes to build specific "products" in the form of new experiences, services, or solutions. The secret to the Digital Factory's success is that its small teams, working closely with the business side, function as a start-up accelerator."

"We see reductions in management overhead of 50 percent for technology teams in the DF, 70 percent in the number of business analysts needed to write technology requirements, and, as test automation becomes the norm, a drop of 90 percent in the number of testers. Finally, we see top engineering talent performing at eight times the level of their peers, as measured with metrics such as code commits."

DevOps Flow - Accelerating Developer Velocity Through Software Factory Best Practices

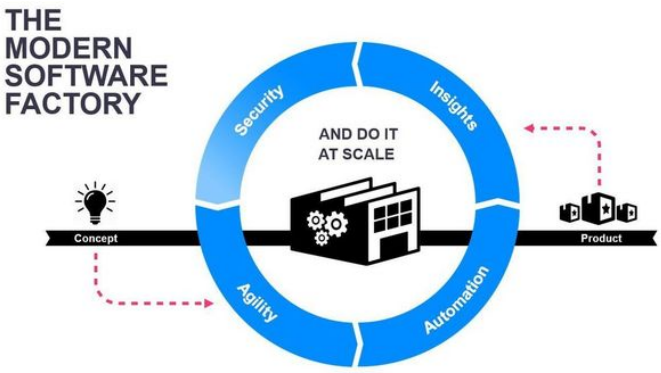
Through the story of a visit to the Digital Factory of a Telecomms business, McKinsey introduce and explain an approach to IT organization that has yielded benefits including:

- bring products to market faster (in six months versus two years)
- do more with existing resources (eight product launches per year versus one or two)
- create dramatically reimagined experiences (opening an account in five minutes versus ten days)
- reduce tech development costs by a third (fewer managers per engineer)
- attract the great talent required to compete in a digital world

Given a context that only 16% of executives rate their digital transformation projects as successful, this recipe for a better way to improve IT delivery overall is therefore very appealing. Especially as they describe how it can be treated as a 'sidebar' unit, a new innovation-centric digital team that doesn't disturb and 'bolts on' to the existing legacy business, or as a complete new broom, wholesale transformation.

'Software Factories' are described as such because the underlying management science is literally derived from the world of manufacturing, utilizing practices such as Lean and Six Sigma, pioneered by organizations like Toyota.

DevOps Flow - Accelerating Developer Velocity Through Software Factory Best Practices



Ultimately the important success factor is building them around key strategic goals – For example reducing mortgage loan rate application turnarounds from three weeks to minutes. These kind of major leaps in the market deliver real strategic advantage.

Digital Factories mainly address the large scale enterprise complexity required to achieve those kinds of improvements, notably breaking down into self-managing, autonomous teams. These 'Agile Squads' are typically 10-15 people in size and bring together business and IT stakeholders to action, and validate, ideas much more quickly.

They are guided by a central Centre of Excellence, which acts as the 'control tower' and provides for central functions notably skills and tools assets and services, such as:

"One of the most important roles in the nerve center is the lead product owner. This person is not a manager but a "doer-in-chief," who works with the local product owner to ensure quality of execution, provide weekly coaching, pressure-test business cases, and review team progress with the goal of understanding bottlenecks and helping to break them. The lead product owner is particularly involved during the first month that the squad is in place and ideally acts as its mentor."

DevOps Flow - Accelerating Developer Velocity Through Software Factory Best Practices

If done right, the Digital Factory is a powerful engine to enable and accelerate not only the business but also the IT transformation agenda. That's because the missions that Digital Factory teams work on have sufficient scale and clarity of focus for IT to develop next-generation capabilities such as automated testing, cloud-based applications, secure coding practices, and application programming interfaces (APIs) that can be put to immediate and practical use.

The concept of a 'software factory' is a very powerful metaphor because much of the insights are drawn literally from the world of manufacturing, where over many decades they have perfected the science of optimizing the efficiency of a production line.

McKinsey developed the Developer Velocity Index (DVI) from an in-depth survey of senior executives at 440 large enterprises, more than 100 expert interviews, and extensive external research, to define a metrics framework based on this factory metaphor.

The principle function of a DevOps Centre of Excellence and the critical framework for defining and achieving high performance are process metrics.

Establishing the right ones will enable development teams to define and measure what matters, providing a control loop for ensuring their transformation and new working processes deliver the results they want.

To determine the business case for Continuous Deployment practices the key dynamic is 'Business Value Throughput'. Ie. not just speeding up the production of deployed code but of software that adds quantifiable value to the organisation.

DevOps Metrics

In his DevOps.com blog, Matt Dickens also explores the same area, providing a very helpful guide to 'Choosing the Right Metrics for DevOps Adoption'.

Matt starts with the key point that the transformation is an ongoing, long term process of transformation, and like always the success factor is having a clear end goal in mind and then breaking it down into a series of 'bite size chunk' steps to progress along that journey.

His article is very helpful because he provides an initial list for suggestions of what these steps might be for different organisations at different stages of their adoption, and then maps each step to the most effective tools and practices improvements to make to progress against that specific metric, such as:

- Deployment time
- Change failure rate
- Release cadence
- Lead time
- Recovery time

In the feature video Mik Kersten, Founder and CEO of Tasktop, describes his journey and evolution as a software developer that led him to understand what is required to achieve high performance software development teams.

He has encoded that learning into the principle of Value Stream Networks that can be implemented and managed through Flow Metrics.

There are four Flow Metrics that measure how value flows through a product's value stream, calculated on four Flow Items – units of work that matter to a business: features, defects, debt, and risk. Any task or effort a software delivery organization undertakes can be categorized as one of these core Flow Items.

- **Flow Velocity** gauges whether value delivery is accelerating also referred to as throughput.
- **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** is the ratio of active time vs. wait time out of the total Flow Time and
- **Flow Load** monitors the number of Flow Items currently in progress (active or waiting) within a particular value stream.

Flow Metrics - Defining Measurements for Achieving High Performance Software Development

Project to Product

Mik has also documented his insights into a book Project to Product, and in this talk he summarizes it by showing how a product operating model can provide the critical glue between the hierarchical and finance-oriented structures of the business and the agility enabled by delivery teams adopting SAFe.

As the title suggests this is centred around a shift from a waterfall, project budgeting approach to one of 'Lean Budgets'.

Lean and value stream thinking originated in manufacturing but have become highly popularized in software delivery by the DevOps movement: With developers releasing code changes more frequently thanks to Agile, DevOps set its sights on getting those code changes running in production faster.

Flow Metrics: Flow time, Flow Velocity, Flow Efficiency

In this webinar Dominica DeGrandis explains Flow Metrics in detail and how to implement these key metrics at an organization and how to enhance the impact of software products.

From 2.08 she sets the scene defining that that flow metrics are tied to business value and are based on outcomes. Flow metrics provide a feedback loop to improve decisions.

At 4:36 Dominica explains the metric, '**Flow Time**'. Flow Time is the duration from when work enters the value stream to its completion. This metric is helpful in answering questions like 'What's the probability of completing work in X days?

This metric helps in identifying when the time to value is getting longer, and is the measure of time taken for the items to process from 'start' to 'complete' state. This would include both active and wait times.

At 16:32 she moves on to discussing the next important metric, which is the '**Flow Velocity**'.

Flow Velocity can be defined as the number of items completed during a given duration of time. This metric is easy to calculate and provides data to the software development teams to view the delivery rates, gauging if the value delivery is progressing in an accelerated manner. This metric is commonly referred to as **throughput**.

At 19:12 Dominica states that there are two important factors that need to be considered while choosing the batch size. Batch size is often decided based on the transaction cost and holding cost. Reducing the batch size of work helps in reducing the WIP (Work in progress) limit and improves the flow.

The optimal batch size primarily depends on holding cost including the cost for delayed feedback, inventory decay, and delayed value delivery). Software development teams must always focus on reducing the transaction costs of any batch.

Flow Metrics - Defining Measurements for Achieving High Performance Software Development

At 20:51 Dominica explains the '**Flow Efficiency**' metric, the percentage of time where work is in an active state and is a metric to expose wait time. Flow efficiency metrics can easily help in identifying when waste is decreasing or increasing in a process. In other work, these metrics can be defined as the ratio of active time vs wait time of the overall flow time.

At 30:22 she defines the metric '**Flow Load**', a metric to measure the balance between demand and capacity. Flow Load is responsible for monitoring the over-utilized and under-utilized value streams, where often the over-utilization and the under-utilization of value streams can lead to reduced productivity. This metric measures the number of Flow Items that are currently in the active or waiting state within a specific value stream.

At 31:32 she emphasizes that WIP (Work in progress) is a leading indicator, for example a high WIP means that the other items are waiting in the queue for a longer time. The important factor which affects queue size is capacity utilization, the metric to calculate the rate at which the outputs are being achieved.

Queueing Theory

From 33:00 Dominica discusses the 'Queueing Theory', which enables quantifying the relationship between wait times and capacity utilization. As the capacity utilization approaches 100%, wait time increases exponentially.

Flow Metrics - Defining Measurements for Achieving High Performance Software Development

Software development teams need to consider managing work by queues if the desired goal is speed. At 35:47, she mentions that the amount of WIP (Work in progress) is the primary factor of speed. If there is a requirement to achieve maximum business value, the proportion of work items in a value stream needs to be adjusted. Often a decision to do one thing is a decision to delay something else.

At 43:20 she illustrates a sample value stream dashboard displaying all the flow metrics. The dashboard would comprise prominent flow metrics like Flow Efficiency, Flow Time, Flow Load, Flow Velocity, and Flow Distribution. The dashboards can also have the business value, business cost, business happiness, and business quality index.

At 46:35 Dominica talks about the metric to gauge happiness, a metric that answers questions like 'How likely are you to recommend working for this company to a friend?

If you are interested in visualizing and optimizing the business value of your software delivery, measuring business outcomes is a must, and Flow Metrics provide a framework for achieving this.

Using Tasktop and Flow Metrics to Drive an Integrated DevOps Toolchain

The DevOps Toolchain refers to the combination of tools and technologies used to progress code through the full life-cycle from development to production.

How effectively this interlinked chain is integrated together is key to the velocity of throughput, as manual hand-overs between steps can introduce significant delays and potential for error.

In this presentation Nicole Bryan of Tasktop and Jeff Zahorchak of Select Medical explain the process of mapping DevOps workflow to build an integrated toolchain and optimize it for high performance through implementing Flow Metrics.

Jeff explains that when they started their journey to look at Flow Metrics, they had six main problems they were trying to solve:

- i) "Swivel chair" entering of data into multiple systems. ii) Combined with multiple methods of fielding work requests from a plethora of tools such as instant messages, email etc., forcing them to context switch between many different applications.
- iii) 60% of what the team delivers were incident resolution, so iv) they weren't working on value adding new features.
- v) Ultimately this meant they were struggling to co-ordinate work across multiple teams, and vi) this required the team to always be over-working to be IT heroes all the time.

Using Tasktop and Flow Metrics to Drive an Integrated DevOps Toolchain

Transformation Journey

Jeff set out to address these issues through a transformation that would:

- Get control of demand through intake management.
- Perform Value Stream Mapping.
- Simplify and integrate delivery tools.
- Implement Flow Metrics to monitor delivery.

The first step was to consolidate all the many different sources of work requests, through developing a custom portal for business users and product owners, with an ROI and approval framework, reducing the interaction methods to three main models: i) An Idea, ii) a Service Request, or iii) an Incident.

From 3m: 45s Jeff explains the heart of the challenge and how they tackled it. He describes a situation of chaos, where there are multiple process flows across multiple applications involving multiple stakeholders, with no central control system coordinating it together, no single source of truth or reporting.

To address this they develop a simplified process flow and value stream map, and transposed this on to their service delivery toolchain, which featured:

- ServiceNow for ITSM.
- Azure DevOps, for CI/CD and TFS for user stories.
- Sharepoint for storing artifacts.
- iRise for ideation and mockups.
- TestComplete for automated testing.
- SQL Server for reporting repositories, feeding Tableau and Microstrategy.

Tasktop Integration Hub

They then utilized the Tasktop Integration Hub to build bidirectional synchronization for every one of the state transitions and artifact types, eliminating the challenge of it being difficult to plan and execute across teams, as all those transition points are now automated.

For example an incident being logged through the help-desk, that would create a defect in the Azure DevOps backlog. When it is updated by a developer the help-desk team can see it in real-time.

At 6m:45s Jeff walks through the details of the benefits achieved from this approach. Remarkably it delivered value within 24 hours of going live – Over 1,200 incidents were synchronized out of the help-desk, de-duplicating 671 items from the backlog, over 4,700 hours of work. A full FTE per day was saved just by eliminating the swivel chair data entry.

Implementing Flow Metrics

Select Medical also then adopted the Tasktop Viz product, which as the name suggests caters for visibility reporting, providing the tools to implement Flow Metrics, explained from 11m:00s onwards.

Jeff begins by highlighting their primary challenge was having a lot of work items in flight at any one time, and they experienced a lot of bottlenecks.

They initially suspected this was due to a shortage of developers, but Viz provided them with the insights to realize that actually it was caused by insufficient BA resources. This data was then taken to the CIO who authorized hiring of the required resource, an unprecedented event for the company, which addressed the bottleneck and sped up the work.

Viz equips teams with the tools to identify work load bottlenecks, where you can click 'Analyze Load' to identify throughputs and restraints, and importantly, by connecting to all the tools used it provides a holistic view of the entire value chain.

For example Jeff showcases a key insight: Due to their adoption of these tools they were able to increase their release frequency to once a week, but this actually caused a spike in support incidents, as too many new features too quickly confused the end users. So instead they slowed this to once a month and better managed the roll outs, with more user training etc.

This highlights that simply improving the velocity of software development throughput is not a standalone goal, but rather the delivery of new products that are successfully adopted by users is instead the objective.

Seamless adoption

At 21m:40s Jeff describes how they connected Viz with their toolchain, emphasizing a key goal was to begin Flow Metric reporting with minimal impact upon their existing work.

Part of the challenge was that Azure DevOps is a very complex environment, but they were able to easily map it's queues, backlogs and touch points as is into Viz and begin collecting metrics. It didn't require them to do anything different in Azure or ServiceNow.

From 23m:30s Nicole explains that Tasktop can model all of the flows across the toolchain, identifying key activities and attributes like debt, features, defects and flow states, such as active, done or waiting.

This last point is key because often organizations aren't even tracking wait-state work. Jeff validated this by highlighting that Azure DevOps doesn't have an On Hold status for its' user stories, which they have since remedied.

Keynote Conclusion: Accelerating Concept to Cash

A critical insight that Nicole shares is the point about the scope of the value stream mapping. Often development teams focus only on a subset of steps, notably only those that they deal with in terms of code commits, often due to what is known as a local optimization bias.

The headline goal of DevOps Flow is to view activity from a holistic business perspective, where the full process actually extends right from the very first ideation right through the end result of deployed operations.

Tasktop synchronizes workflows across all of these applications so that metrics and optimization activity is dealing with this complete end-to-end value chain.

This means performance is being managed at a level that is meaningful to business users; in short the process often described as 'Concept to Cash' – How long does it take the business to translate a new idea into a revenue-generating service.

Thus they can identify and invest in addressing flow bottlenecks within a context of there being a clearly understood and expected improvement that can be quantified in hard ROI terms.

The Power of DevOps Combined with Value Stream Management

The key activity for realizing improvements in DevOps performance is Value Stream Management.

As Wikipedia describes Value Stream Mapping is a lean-management method for analyzing the current state and designing a future state for the series of events that take a product or service from the beginning of the specific process until it reaches the customer.

A value stream map is a visual tool that displays all critical steps in a specific process and easily quantifies the time and volume taken at each stage.

DevOps Value Streams

Mapping DevOps value streams enables the team to understand how work flows through the different functions, right from idea inception through to deployment, and where constrictions occur that slow the overall system throughput, identifying key constraints like slow hand-off interactions.

Writing for the New Stack, Jeff Keyes of Plutora, defines that Value Stream is the Future of DevOps, highlighting that often DevOps mistakenly focuses purely on tools and automation, and that VSM is key to unlocking high performance.

The DevOps Institute explains why it is necessary, stating it's essential to understand the flow of work across the value stream from the idea to software deployment and ensure that we set up feedback loops until the customer or the user realizes the business value.

Helen Beal, Chief Ambassador for the Institute, shares this comprehensive presentation that demonstrates combining the power of DevOps with Value Stream Management provides unparalleled insights into the progress of DevOps adoption.

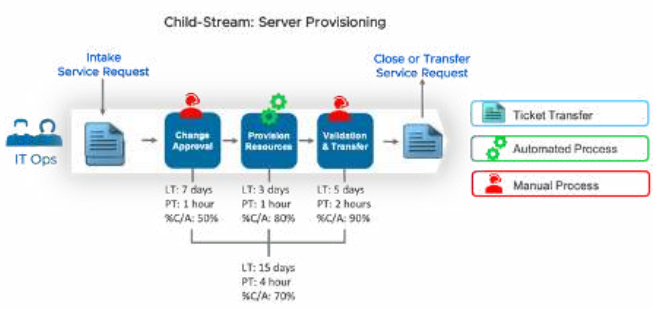
The Power of DevOps Combined with Value Stream Management

As Dominica DeGrandis explains in this blog the disconnects between systems and departments are typically the main focal point of errors and lost time and delays. The hand-offs generate a considerable wastage of time and effort, slowing the DevOps Flow.

The Carnegie Mellon Software Engineering Institute describes that VSM is a Lean technique for visualizing, characterizing, and continuously improving the flow of value across this set of end-to-end activities by eliminating barriers, whether procedural, cultural, technological, or organizational.

Implementing Value Streams

On the VMware blog Mandy Storbakken provides an example of value stream mapping IT workflows and how this can illustrate the means for defining DevOps Flow metics, via a server provisioning process:



50% Complete & Accurate – Half the time, this stage cannot be completed without gathering more information or correcting something; **1-hour Processing Time** – This stage could be completed in one hour, if there were no delays; **7 days Lead Time –** This stage typically takes seven days to complete, which could be time in the queue, time awaiting additional information, or time waiting for the change review board.

She adds:

"The metrics across all stages, from ops intake to ops hand-off, show that while it typically takes around 15 days to complete the stages, the actual processing of work only accounts for four hours over that time (15 days LT, four hours PT). An Activity Ratio (AR) – which is the total process time divided by the total lead time, times 100 – can also be derived from these metrics."

From this she concludes by articulating how VSM typically uses four core metrics to analyze the flow of work through each stage in a value stream, and that these metrics should be derived from *actual timings* for each stage (not based on SLAs):

- **Lead Time (LT)** – Time to complete the stage, from intake to hand-off.
- **Process Time (PT)** – Time it could take to complete the stage, if all information were complete and the process were uninterrupted.
- **Complete & Accurate (%C/A)** – How often the stage can be completed without needing additional information or corrections.
- **Value Added (VA)** – A 'yes' or 'no' indication of whether the stage adds direct customer value.

Speaking at the DevOps Conference, Stephen Walters of GitLab defines the measures that should be inspected to identify the typical measures for the flow of value. Stephen is a co-author of the DevOps Institute Value Stream Reference Architecture Paper, a best practice framework to correctly identify value streams within an enterprise, and to implement organization around those value streams.