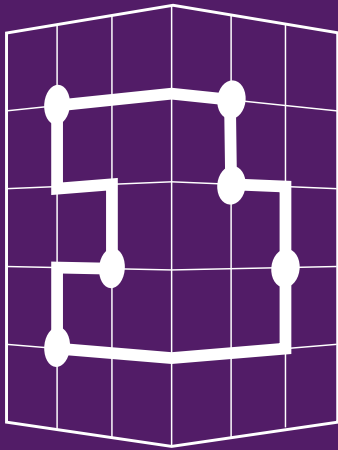


# Value Stream Architecture



Creating an Architecture  
to Connect the Dots  
in DevOps

---

# Value Stream Architecture

Creating an Architecture  
to Connect the Dots  
in DevOps

---



25 NW 23rd Pl  
Suite 6314  
Portland, OR 97210

Value Stream Architecture:  
Creating an Architecture to Connect the Dots in DevOps

Copyright © 2017 IT Revolution

All rights reserved, for information about permission  
to reproduce selections from this book, write to:

Permissions  
IT Revolution Press, LLC  
25 NW 23rd Pl, Suite 6314  
Portland, OR 97210

First Edition  
Produced in the United States of America  
10 9 8 7 6 5 4 3 2 1

Cover design and interior by Devon Smith

For further information about IT Revolution, these and other publications,  
special discounts for bulk book purchases, or for information on booking authors  
for an event, please visit our website at [www.ITRevolution.com](http://www.ITRevolution.com).

# PREFACE

In April of this year, we at IT Revolution had the pleasure of hosting technology leaders and experts from across the DevOps Enterprise community at the DevOps Enterprise Forum event in Portland, Oregon. The Forum’s ongoing goal is to create written guidance to overcome the top obstacles facing the DevOps Enterprise community.

Each year at the Forum, the topics covered have included organizational culture and change management, architecture and technical practices, metrics, integrating and achieving information security and compliance objectives, creating business cases for automated testing, organizational design, and many more.

For the first two years, we organized the participants into large teams that worked on a small number of broad topics. However, this year, we shifted our approach in two ways—first, we invited a core group of past participants to propose topics they would like to work on and second, we asked them to narrow their topics so that they could have “nearly shippable” artifacts by the end of the second day. The result was more teams working on more problems with more written guidance.

After the Forum concluded, the groups spent the next eight weeks working together to complete and refine the work they started together. The results can be found in this year’s collection of Forum papers. I hope you will agree that the smaller teams and reduced scope of the guidance benefits both the teams as well as the reader.

IT Revolution is proud to share the outcomes of the hard work, dedication, and collaboration of the amazing group of people from the 2017 DevOps Enterprise Forum. Our hope is that you will gain valuable insight into DevOps as a practice.

—Gene Kim  
June 2017  
Portland, Oregon

# INTRODUCTION

While traditional architecture principles focus on the “ilities” of software development, such as usability, maintainability, scalability, availability, extensibility, and portability, the value stream architecture model focuses on creating a transparent, repeatable, self-healing model for the delivery of software. The key difference in the two architecture paradigms is that the traditional “ilities” model focuses on the software product while the value stream model focuses on the *delivery* of the software product.

As organizations continually evolve and rely on cross-functional teams coming together and adopting DevOps practices, the process of delivering software products must also evolve. We believe value stream architecture is the facilitator of organizational process change and transparency, and should be the primary focus for every organization striving for DevOps greatness.

# STORY

CraftSoft Inc. is a new company established in Portland, Oregon, in April 2017. CraftSoft was formed through a merger of three businesses that offer related but unintegrated products, allowing it to acquire a large software portfolio right out of the gate. These products range from financial services to gaming software to beer delivery. The main challenge CraftSoft's leadership is facing is determining how to combine the software architecture and operations into a cohesive set of offerings that provide economies of scale. CraftSoft is racing to differentiate itself in the market by providing new, innovative customer experiences based on their offerings. For example, they are leveraging gamification of their payment services to provide free craft beer to their most loyal customers.

Technical due diligence completed prior to the launch of CraftSoft identified that the architectures would be compatible, with each product suite based on Java and REST APIs. However, post merger CraftSoft's CIO noticed that the three sets of products had completely different delivery models, operational environments, and toolchains. While there was a clear plan to connect the architectures, no such plan existed for connecting what one board member referred to as the "value streams" of each of the products, i.e., the processes and tools that each used to deliver features and fixes to its customers. The board member had a background in manufacturing and pointed out that until these separate value streams were integrated and connected to the business strategy, the organization should hold off on its very aggressive staffing plans, as nobody could currently tell where the bottlenecks would be.

In response, the CIO created a new initiative to apply the same kind of rigor to the value stream architecture of the organization as had been applied to the software architecture.

She set out to define a value stream architecture with each of the IT and line of business (LOB) leads. A key goal of creating this shared value stream framework was to minimize cost while offering the same architecture to the different lines of business and enable each to staff and scale as needed.

As a first step, the CIO decided to hire a dedicated leader, a value stream architect, to drive this transformation starting from different value streams inherited from the different lines of business. The role of the value stream architect was to continuously improve the value

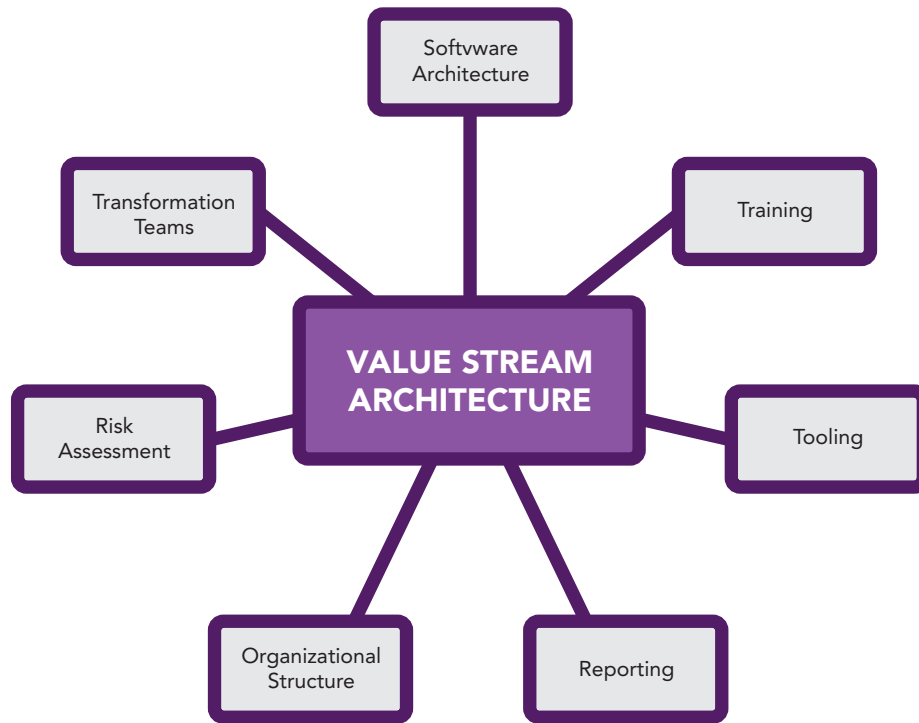
stream architecture based on different stakeholder needs (e.g., CAO, CISO, CIO, DevOps teams) and based on facts that are mined throughout the value stream map through analytics. The architect was also responsible for the value stream integration infrastructure needed to connect the end-to-end toolchain and processes ranging from Agile planning to ITIL.

## Tenets

The following are an initial set of tenets for the role of the value stream architect (VSA) in the organization.

### Value Stream Architect Tenets

<b>Software Architecture</b>	The VSA will impact the software architecture discipline of an application to be centered around flow. For example, APIs are created when they enable the creation of more features by a team or product line (e.g., increase business value flow for that line).
<b>Training</b>	The VSA will help identify what training programs are to be made available to teams and also identify role changes that will be required. For example, if a new code review tool is deployed, the VSA's perspective is to ensure that enough senior engineers are assigned to prevent a bottleneck on code review.
<b>Tooling</b>	While tool selection continues to involve numerous stakeholders, all production tools must be connected to the flow defined by the VSA. For example, a new build tool can be experimented with but will not be standardized until it is proven to integrate with upstream and downstream tools.
<b>Reporting</b>	An optimized reporting model for VSAs is one that allows for VSAs to report into the LOB to get visibility into what is happening at the ground level, similar to a manufacturing floor view, as well as have some VSAs report directly to the CIO to aggregate changes to tooling and platforms that need to be influenced across the LOBs.
<b>Organizational Structure</b>	The organizational structure must be aligned to the desired value stream architecture. For example, the organization may need to break down execution teams into squads or product organization so as to be self-sufficient in optimizing the value stream for the success of the LOB.
<b>Risk Assessment</b>	A core part of the VSA will be to ensure compliance and automate risk assessment. For example, if a new static analysis code review tool is deployed, the reporting capabilities make visible all teams that are not yet using the tool.
<b>Transformation Teams</b>	The VSA model will influence the organization to establish a transformation team to implement and support tools/platforms that can be cross-leveraged across LOBs. This will ensure that the LOBs backlog is not impacted with solving for value stream optimization but can instead focus on delivering business value.



*Fig. 1: Value Stream Architecture Tenets*



# THE VALUE STREAM ARCHITECT

As organizations adopt DevOps practices, many struggle with creating a holistic view of the value stream from portfolio/feature inception to the end-of-life of a function or capability. The lack of visibility into this continuous loop is due to poorly defined feedback mechanisms and the inability of the organization to anticipate and influence the platforms that use the automation of these processes and platforms. As products are delivered through the DevOps continuum, the ability to focus resources into areas that will continually streamline and increase velocity of deliverables across the organization is crucial to enterprise adaptability and success. Enter the value stream architect.

The value stream architect is an influencer, a consultant, an optimizer, and an architect. The role consists of several factors to ensure success:

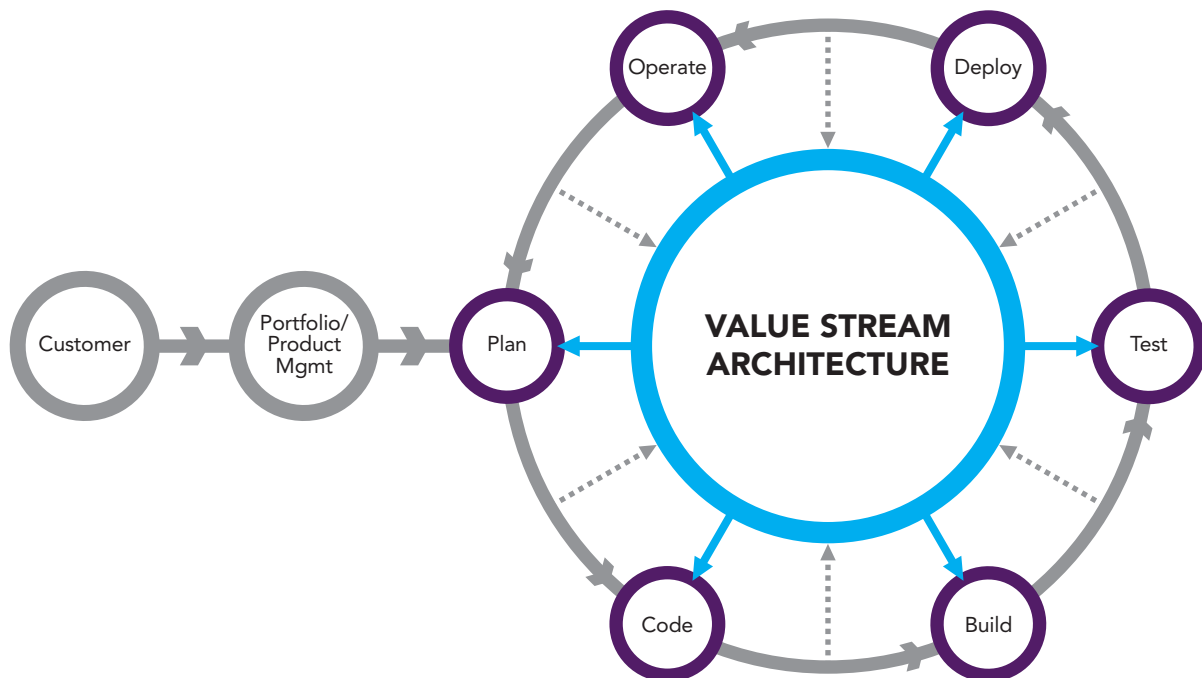
- **Visibility:** Without visibility into each of the lines of business/products, the value stream architect is working in a vacuum. It's crucial that feedback mechanisms are designed and implemented to give the value stream architect a 360-degree view of the entire software development and deployment process. These feedback mechanisms are the core component of any value stream architecture.
- **Influence:** The VSA will need the support of the organization to drive change into each of the processes and sub-systems supporting the value stream. The value stream architect's ability to influence crosses multiple boundaries:
  - ♦ Through the collection and analysis of feedback data, the mechanism for driving change should take the form of the value stream architect adding stories to each of the sub-system backlogs. These stories indicate inefficiencies within the sub-system that the value stream architect has identified based on the current backlog and the expected business result metrics. The inefficiencies identified in these feedback loops are tied directly to the current value stream.
  - ♦ Outside of individual value streams, the value stream architect influences and consults with each of the platform owners to drive higher level decisions about platform and tooling. These decisions are driven as core capabilities for the organization used across all value streams.
- **Architecture:** Architecture and modeling will be key to achieving success in this role:

- ♦ Create the framework for the feedback mechanisms; create the feedback flow.
- ♦ Visualize each of the key flows through value stream diagrams. Understand the current constraints and document the weak links in the chain of each value stream.
- ♦ Create target architectures for each of the value streams. The byproduct of this exercise will be the gaps that each of the value streams have from their current state to their target state. These gaps will be key for the VSA to understand the capabilities and liabilities of each of the value streams.
- **Mentoring and Training:** With a holistic view of the value streams, the VSA will continually identify training and skill deficiencies and will work with IT leaders to close the knowledge/skills gaps with those given teams.

# VALUE STREAM FLOW METRICS

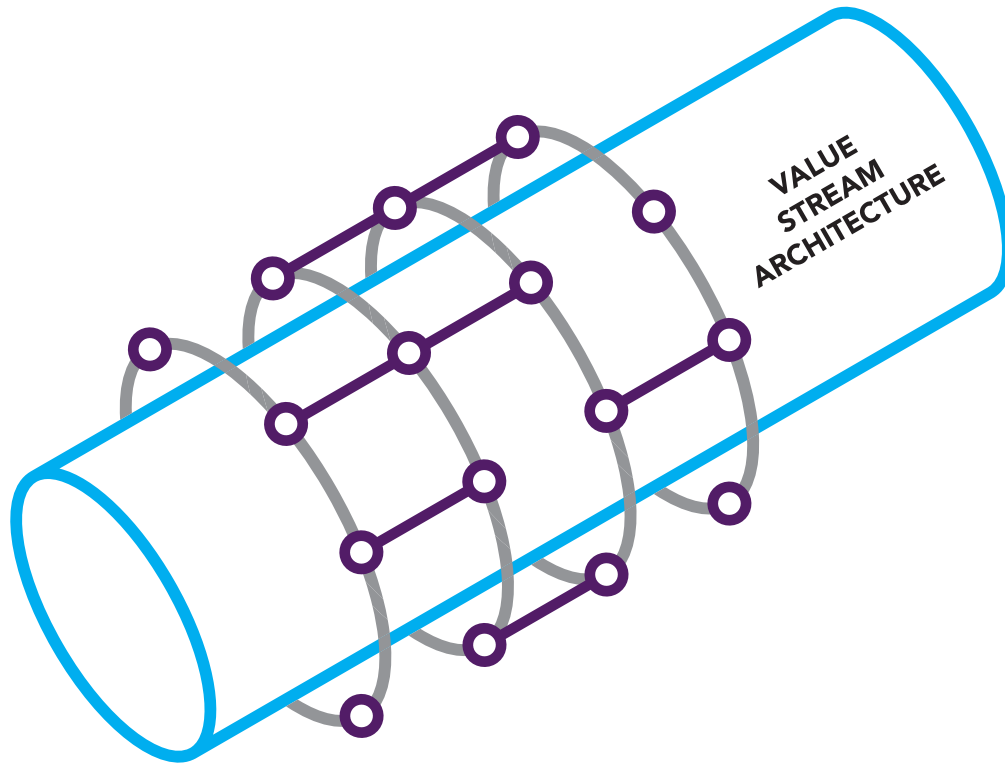
In order to create a consistent way of measuring value stream flow, we need to define the units of flow. In manufacturing, these represent the widgets leaving the factory floor according to the way that the customer pulls value from the producer. In software, the widgets aren't physical units, but need to similarly capture units of value that are pulled by the customer.

The value stream architecture flow for a single product is represented below (Figure 2). The diagram is a representation of the steps in a typical Agile/DevOps process with the value stream architecture touchpoints defined as arrows of feedback loops in and out of the value stream framework.



*Fig. 2: Value Stream Architecture Diagram*

Taking the above diagram and expanding that at an enterprise level, you start to see the commonalities and reuse of components of the architecture to increase reuse. In Figure 3, you see that the value stream architecture is the foundation that is cross-cutting through each of the product architectures. Each of the product architectures is added to the value stream through connecting APIs and feedback loops.



*Fig. 3: Enterprise Value Stream Architecture*

Using the feedback loops in the value stream architecture, we propose that the units of flow being pulled by the customer are the artifacts that define value to the customer experience with a software application. We define five types of artifacts to capture all business value flowing through the end-to-end value stream.

### Value Stream Flow Units

Flow Units	A.K.A.	Pulled By	Description
Feature Story	Requirements, User Stories	User/Customers	New business value added to the application, visible to the customer
Defect Story	Bug	User/Customers	Fixes for quality problems that affect customer experience
Security Story	Vulnerability	Security Officers, Auditors	Fixes for weaknesses, flaws, and vectors for exploits
Tech Debt Story		Software Architects, Teams	Improvement to the software architecture, invisible to the customer
Value Stream Story		Value Stream Architects	Improvement to the value stream architecture, invisible to the customer

In different segments of the value stream, these units can be further expanded. For example, a *ticket* or *incident* can trigger a software or infrastructure defect to be created, or a feature story can be expanded into development tasks and tests. But we propose that these are the only top level artifacts needed to represent throughput through the software delivery value stream.

We measure the effectiveness of the entire value stream using the following flow metrics:

### Value Stream Flow Units

Flow Measures	Description	Example
Backlog Size	Number of artifacts in the queue at a particular stage of the value stream	The number of security stories on an Agile team's backlog
Lead Time	Time from request to delivered artifact running in production	Hours from customer requesting a feature to using the requested feature
Cycle Time	Time from work start to completion	Days from developer accepting a feature story to merging the finished implementation to master
Process Time	Time an artifact is actively worked on	Cumulative days a feature story is actively worked on
Wait Time	Time from work request to work start	Time zone delay in hours for an offshore team to start work on a feature story
Utilization	Sum of process times of lead time (%)	Low utilization for a Dev team with an external dependency

Using these artifact types and flow metrics we can derive all relevant measurements, such as:

### Sample Value Stream Flow Metrics

Flow Metrics	Type	Description
End-to-End Lead Time	Velocity	Time from customer request to improvement running in production
Cost per Work Item	Cost	Work items per FTE (full-time equivalent), change over time
Mean Time to Repair (MTTR)	Responsiveness	Round trip time from support ticket, through defect story creation, diagnosis, fix through to delivery
Productivity	Throughput	Number of work Items delivered per team per release
New Business Value Ratio	Value	Proportion of artifacts of type feature story delivered per release
Utilization	Sum of process times of lead time (%)	Low utilization for a Dev team with an external dependency

# CONCLUSION

Organizations are constantly changing, whether the change is organic or dictated by management, the common goal of change is to deliver value. No matter how consistent we try to make our changes, without the proper processes those changes are dynamic and hard to measure. Value stream architecture is the repeatable process that allows for change while building constant feedback mechanisms to ensure the changes being made are adding the perceived value.

The adaptability that the value stream architecture gives an organization to optimize its delivery processes is measurable and repeatable through the flow units and flow metrics. These attributes of the architecture allow for a self-healing capability that drive positive change and value back into the architecture through feedback loops. The value stream architecture approach to DevOps and process improvement will continually add value back into the delivery process as long as the architecture is the key focus of the process improvement journey.

## *Contributors*

Mik Kersten, CEO & Co-Founder Tasktop, [mik@tasktop.com](mailto:mik@tasktop.com)

Nanda Kumar, Fellow at Verizon, [Nanda.Kumar@VerizonWireless.com](mailto:Nanda.Kumar@VerizonWireless.com)

Daniele Romano, Global Continuous Delivery Product Owner at ING,  
[Daniele.Romano@ing.nl](mailto:Daniele.Romano@ing.nl)

Jason Zubrick, Director of Solution Architecture at GameStop, [jasonzubrick@gamestop.com](mailto:jasonzubrick@gamestop.com)