



# MASTERING CHAOS WITH VALUE STREAM MANAGEMENT AND PATTERN THINKING

An Expert Series by Broadcom



# Our Expert

**LANCE KNIGHT** | Chief Value Stream Architect | Broadcom

Lance Knight is the Chief Value Stream Architect at Broadcom Inc. Previously, he was the CEO and President of ConnectALL LLC and held SVP/VP roles at LeadingAgile, Tasktop Technologies, and Accept Software, specializing in field operations, sales development, and customer success. Lance started his IT career with a large aerospace manufacturer, where he learned about Lean Manufacturing and Systems Thinking. He's a published author of books and white papers on leadership, software development, and software sales.

**Enhance flow by identifying and reducing noise, improving visibility, alignment, and efficiency to increase predictability.**





# Table of Contents

## Chapter 1: Introduction

## Chapter 2: Noise in the Software Delivery Value Stream

What is Noise

Types of Noise

Effects of Noise on Software Development and Delivery

Tools to Help Reduce Noise

## Chapter 3: Value Stream Management Fundamentals

What Value Stream Management Is and Isn't

Using Lean Principles and Systems Thinking in Value Stream Management

Identifying Non-Value-Added and Value-Added Activities

How Are Organizations Implementing Value Stream Management to Reduce Noise in Their Value Streams?

## Chapter 4: Pattern Thinking

An Intro to Pattern Thinking

Pattern Thinking in Value Stream Management

Benefits of Using Pattern Thinking in Value Stream Management to Accelerate Software Delivery

04	How to Recognize Patterns in the Software Delivery Value Stream Using Systems Thinking and Lean to Control Flow	18
05	How to Start Thinking Patterns to Reduce Noise That Leads to Non-Value-Added Activities	19
05	Uncovering Patterns in Software Delivery Value Streams	20
06	Wasteful Patterns in Software Development and Delivery	21
07		
08	<b>Chapter 5: Value Stream Automation</b>	<b>22</b>
	An Intro to Value Stream Automation	22
09	Pattern Thinking and Value Stream Automation	23
09	How Value Stream Automation Helps Accelerate Value-Added-Activities	24
10	Automation Patterns and How It Helps to Reduce Noise	25
12	How to Start Implementing Automation Patterns	26
13	Examples of Patterns in DevOps and Agile and How to Use Automation to Accelerate Software Delivery	26
	How a Value Stream Management Platform Can Help with Value Stream Automation and Pattern Recognition	27
14		
14		
16	<b>Chapter 6: Conclusion</b>	<b>29</b>
17		

## CHAPTER 1:

# Introduction

With countless variables, software delivery can be chaotic, noisy, overwhelming, and unpredictable. From development to deployment, you need clear direction for navigating the chaos to avoid drowning in data and noise. But what if we told you there's a way to reduce the noise and identify both organic and structured patterns within your software delivery value stream, even those not built for efficiency, to help you navigate the chaos to clarity and order?

Our brains are wired to find patterns in the world around us; pattern recognition is essential, from identifying faces in a crowd to recognizing the flow in a method or process. In value stream management, pattern and systems thinking can identify inefficiencies and improve productivity. Have you considered how this will work in business operations?

By understanding these patterns, you can optimize your processes through automation and streamline your workflow for maximum efficiency, thus addressing the enduring issues such as organizational silos, friction, overproduction, and other inefficiencies. It will help you identify and remove bottlenecks, slowing development

cycles, and accelerate your team's software delivery pipeline while reducing errors and boosting efficiency.

Value Stream Management (VSM) helps businesses maximize their efficiency and productivity by providing visibility into the value stream, aligning teams towards common goals, and optimizing workflows. It helps identify inefficiencies or areas where improvement can be made. Just like how our lives are filled with value streams, those are filled with patterns.

Pattern Thinking is an essential concept in VSM, as it helps to identify patterns that can be used to improve the overall process. It looks at how each step of the process interacts with each other and how they can be improved by making small changes that have significant impacts. By understanding these patterns, businesses can optimize their value streams and create a more efficient workflow and deliver value to customers more often.

Thinking in patterns is a vital approach in software development and delivery today because it provides a framework for understanding and improving complex systems. Software development and delivery are inherently complex processes that involve multiple teams, technologies, and stakeholders. Pattern thinking helps developers and teams to make sense of this complexity by identifying recurring patterns and structures that can be optimized and improved.

Whether it's an overloaded inbox or a bottleneck in the production line, these issues can quickly lead to frustration and lost productivity. That's why pattern thinking and value stream management are two essential tools for improving flow.

In this e-book, we will show you how to uncover hidden insights into your software delivery value stream by thinking in patterns and how you can automate effectively to reduce the noise in the value stream. Before we dive deep into the specifics of pattern thinking and value stream management, let's define the terms.

# Noise in the Software Delivery Value Stream

## What is Noise

Software development and delivery is a complex process that involves multiple teams, tools, and techniques working together to deliver high-quality software. However, noise is a common enemy in this intricate web of activities. Noise can manifest differently, slowing down or derailing your software delivery value stream.



In a software delivery value stream, noise refers to interruptions that prevents your teams from doing the actual work. It is not the value-added work but all the things that will make take a decision about the value-added-work. This creates variation in the flow of information, artifacts, or processes that can negatively impact the efficiency and effectiveness of the value stream. This can include delays, errors, rework, bottlenecks, poor communication, or other inefficiencies.

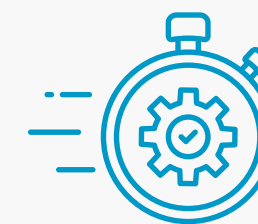
Noise can come in many different forms. Overcommunication, excessive meetings, streams of emails, context switching, relentless feedback, over-processing, over-governance, over-production, changing requirements, and approvals are some examples of noise. Let's look at couple of them in-depth.

For instance, a colleague could stop by your desk for a quick chat when you're working on an urgent task. Or it could be a never-ending stream of emails and notifications constantly demanding your attention.

Another common source of noise is ambiguity around project requirements or expectations. When these need to be clearly defined, team members may save time trying to clarify them instead of focusing on their tasks.

Noise within the software delivery value stream results in waste, increased lead times, reduced quality, and diminished customer satisfaction. This noise, rooted in challenges such as organizational silos, friction, overproduction, and inefficiencies, undermines the overall performance of the value stream. Identifying and eliminating noise in a value stream is a crucial principle of lean manufacturing and continuous improvement. It allows organizations to increase efficiency, reduce costs, and improve overall performance.

By identifying and reducing noise within the value stream using automation and simplifying workflows, organizations can:



**Improve overall efficiency**



**Reduce cycle times**

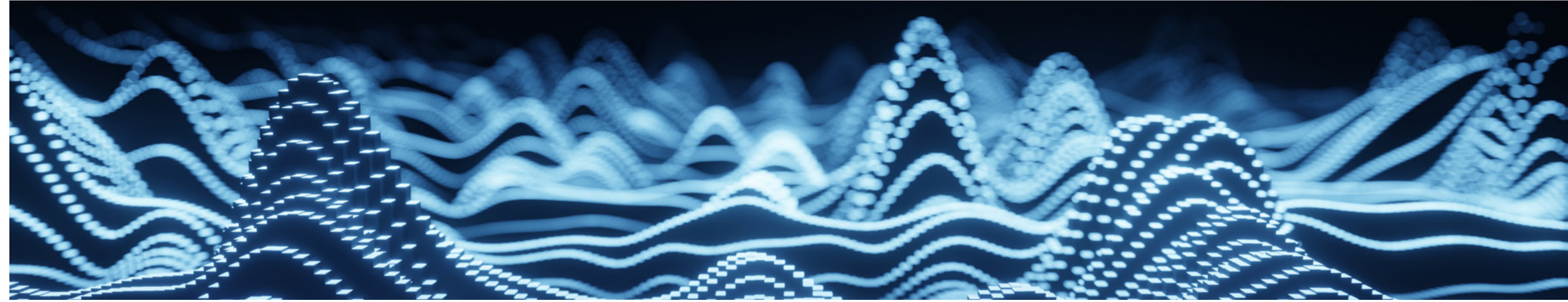


**Streamline communication and collaboration between teams**



## Types of Noise

Noise in a software delivery value stream can take many forms, such as incomplete or unclear requirements; Poor communication; Bottlenecks; Technical debt; Manual processes. All of these can be classified under four types of noise.



### Communication Noise:

This type of noise occurs when there is a breakdown in communication between team members or external stakeholders. Poor communication can lead to misunderstandings, delays, and errors in the development and delivery process. Excessive emails, updates, meetings, and other overcommunication within a software delivery value stream are examples of communication noise.

### Feedback Noise:

This type of noise refers to any extraneous or irrelevant information introduced into the feedback loop, making it difficult to assess the quality of the software being delivered accurately. This can include non-specific, contradictory, or untimely feedback that does not provide valuable insights or is not actionable or relevant to improving the system.

- **Silos**
- **Lack of communication**
- **Unconnected systems**

### Process Noise:

This type of noise occurs when there are inefficiencies in the software development process, such as manual processes, bottlenecks, technical debt, and incomplete/unclear requirements that slow down software delivery.

- **Under/Overproduction**
- **Under/Overgovernance**
- **Under/Overprocessing**
- **Under/Overengineering**
- **Defects and rework**
- **Manual handoffs**
- **Moving between systems**
- **Waiting (for approvals, work to be completed, feedback, etc.)**

### Technical Noise:

Technical noise can result from hardware failures, software bugs, or other technical issues that disrupt the flow of work and information.

# Effects of Noise on Software Development and Delivery

When the tools, people, and processes in a software delivery value stream are disconnected, noise can exacerbate existing challenges and create new ones. Here are some examples of how noise can negatively impact software delivery in this scenario:

## 01 | Increased Handoffs:

Handoffs between teams or individuals in the value stream can be a source of noise if they are not well-defined or adequately executed. When noisy, handoffs can create confusion about responsibilities, delays, and errors.

## 02 | Inefficient Use of Tools:

If teams use different tools that don't integrate well, it can create noise as data and information are transferred between them. This can lead to manual workarounds, wasted time, and increased chances of error.

## 03 | Miscommunication:

Lack of communication or miscommunication between teams or individuals can introduce noise into the value stream, leading to misunderstandings, rework, and delays.

## 04 | Increased Risk:

When there is noise in the value stream due to disconnected tools, people, and processes, it can increase the risk of defects, outages, and security vulnerabilities.

**It is essential to establish clear workflows, roles, and communication channels to mitigate the effects of noise in a value stream with disconnected tools, people, and processes. Teams should also strive to use integrated toolchains that minimize the need for manual intervention or data transfer. Finally, regularly analyzing and optimizing the value stream can identify potential noise sources and help reduce their impact over time.**



## Tools to Help Reduce Noise

Efforts to reduce noise within the software delivery value stream are crucial for optimizing efficiency, creating alignment, and increasing visibility. Automation, workflow simplification, and the adoption of integrated toolchains are key strategies for eliminating noise and improving overall performance.

By establishing clear workflows, roles, and communication channels, organizations can mitigate the effects of noise and enhance collaboration between teams. Integrated toolchains and value stream management techniques help streamline processes, reduce waste, and minimize the need for manual intervention.

Incorporating lean principles and value stream mapping into software delivery practices enables organizations to identify and eliminate sources of noise, ultimately improving efficiency and enhancing the value delivered to customers.



Value Stream Map



System Map



## CHAPTER 3:

# Value Stream Management Fundamentals

## What Value Stream Management Is and Isn't

The software industry is overwhelmed and is drowning in thousands of VSM definitions — some are broad and some unclear. The market is also flooding with new vendors who claim to provide VSM solutions to drive agility and velocity and accelerate software delivery — some of which are not even complete solutions.

Having worked in the B2B SaaS space and specifically practicing lean methodologies like VSM for over two decades, we have advocated the correct understanding and adoption of VSM in the software industry.

First, let's get one thing straight — what VSM isn't.

- **It isn't the same as a value stream**
- **It isn't a defined process**
- **It isn't a tool or a platform**
- **It isn't the same as value stream mapping**

- **It isn't the same thing as DevOps, Agile, Scrum, Lean, or Kanban**
- **It isn't just project management**
- **It is a bit of a misnomer**

So then, what is VSM? Gemba Academy states, "A value stream can be defined as all the steps — both value-added and non-value-added — required to take a product or service from its raw materials into the waiting arms of a happy customer."

Value stream management is about managing the way you deliver value. According to Taiichi Ohno, father of the Toyota Production System, "All we are doing is looking at the timeline from the moment a customer gives us an order to the point when we collect the cash. And we are reducing that timeline by removing the non-value-added wastes."

Value stream management is a human-driven continuous improvement strategy that connects the needs of the customer and leadership with the needs of the business and technical operational groups to ensure predictable delivery outcomes.

## Here's what you need to know about VSM:

Value stream, value stream mapping, value stream management, and value management are all different but related

You don't need to create a value stream, you already have them

VSM is Human and is all about improvement

Value management is getting valuable work through the value stream

Organizations have to do both value management and value stream management

A VSM platform gives you the power of connections and allows you to do both VSM and value management



# Using Lean Principles and Systems Thinking in Value Stream Management

Before we understand how lean principles and systems thinking can be used to improve the use of VSM, let's know what the two mean.

## Lean Principles

Lean principles are a set of guiding philosophies and practices initially developed by the automotive industry's Toyota Production System (TPS). These principles have since been widely adopted in software development and are known as "Lean Software Development" to improve efficiency, eliminate waste, and promote continuous improvement. In this context, the principles focus on delivering customer value through the efficient and effective delivery of software products and services.

Several core principles of lean include Value, Flow, Pull, Continuous Improvement, Elimination of waste, and Empowering humans. By applying these principles, organizations can create a more efficient and effective software development process, producing high-quality products that more closely meet the needs of their customers.

## Systems Thinking

Systems thinking is an approach to problem-solving that focuses on understanding how different components in a system interact and influence each other. For instance, In software development and delivery, systems thinking involves understanding the various parts of the software development lifecycle, including stakeholders, processes, technologies, and environments, and how they all impact each other.

Systems thinking in software development and delivery involves looking beyond individual tasks or processes and considering the entire system. This can include analyzing feedback loops, identifying patterns, and looking at cause-and-effect relationships throughout development.

For example, by applying systems thinking to software development, organizations can identify the root causes of defects, delays, and other issues and make targeted improvements to alleviate those problems. Systems thinking can also help organizations identify opportunities to optimize their software development processes by analyzing the impact of changes to one area of the system on other areas.





Lean principles and systems thinking are two fundamental approaches that can be used to improve value stream management. Here are some ways that these approaches can be applied:

## 01 | Value Stream Mapping

---

Systems thinking and lean principles emphasize the importance of taking a holistic approach to analyze processes and identify opportunities for improvement. Value stream mapping is a tool that can help you better understand your overall system and identify areas of waste and inefficiency.

## 02 | Continuous Improvement

---

Both lean principles and systems thinking focus on continuous improvement. By regularly reviewing and optimizing processes, you can identify areas for improvement and make changes to increase efficiency and reduce waste.

## 03 | Eliminating Waste

---

Lean principles aim to eliminate waste in all forms, such as overproduction, waiting, defects, overprocessing, excess inventory, unnecessary motion, and unused talent. By examining processes and identifying waste areas, you can implement changes to reduce or eliminate these wastes.

## 04 | Pull Systems

---

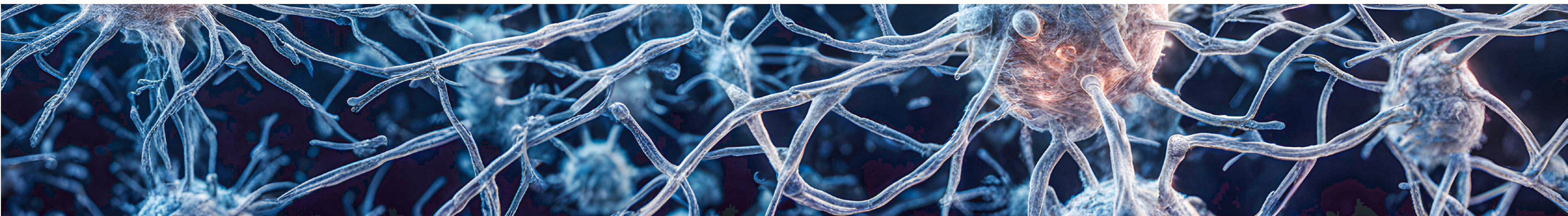
Both lean principles and systems thinking to encourage using pull systems, which involve only producing what is needed by downstream processes or customers. This approach helps to avoid overproduction and minimize waste.

## 05 | Cross-Functional Team

---

Systems thinking highlights the importance of collaboration between different departments to optimize the system. Lean principles also promote cross-functional teams and encourage everyone to contribute to process improvement.

Combining these approaches can create an effective value stream management system. First, you can use value stream mapping to visualize the system and identify waste areas. Then, by applying lean principles, you can design processes that minimize waste and follow a pull-based approach. Finally, by adopting a systems thinking mindset, you can ensure that the entire system works smoothly and efficiently, with all teams working collaboratively towards continuous improvement.





# Identifying Non-Value-Added and Value-Added Activities

VSM, which also stems from Lean, enables the use of value analysis technique that helps identify value-adding activities in a chain of activities. Value analysis in lean implementation involves assessing each process step through the eyes of the customer and determining whether the step is:

## Value-Added (VA)

Activities that will directly achieve customer requirements and those for which the customer is willing to pay.

### Examples:

- Requirements analysis
- Design
- Coding
- Deployment
- Training

## Non-essential Non-Value-Added (NVA)

Activities that will take time or resources but do not directly achieve customer requirements or those for which customers will not be willing to pay.

### Examples:

- Finding bugs
- Rework (fixing bugs)
- Internal approvals
- Toggling between tools
- Extra/unnecessary features

## Essential Non-Value-Added (NVA)

Activities considered non-value-added from a customer perspective but can satisfy a regulatory/compliance issue or other business requirements.

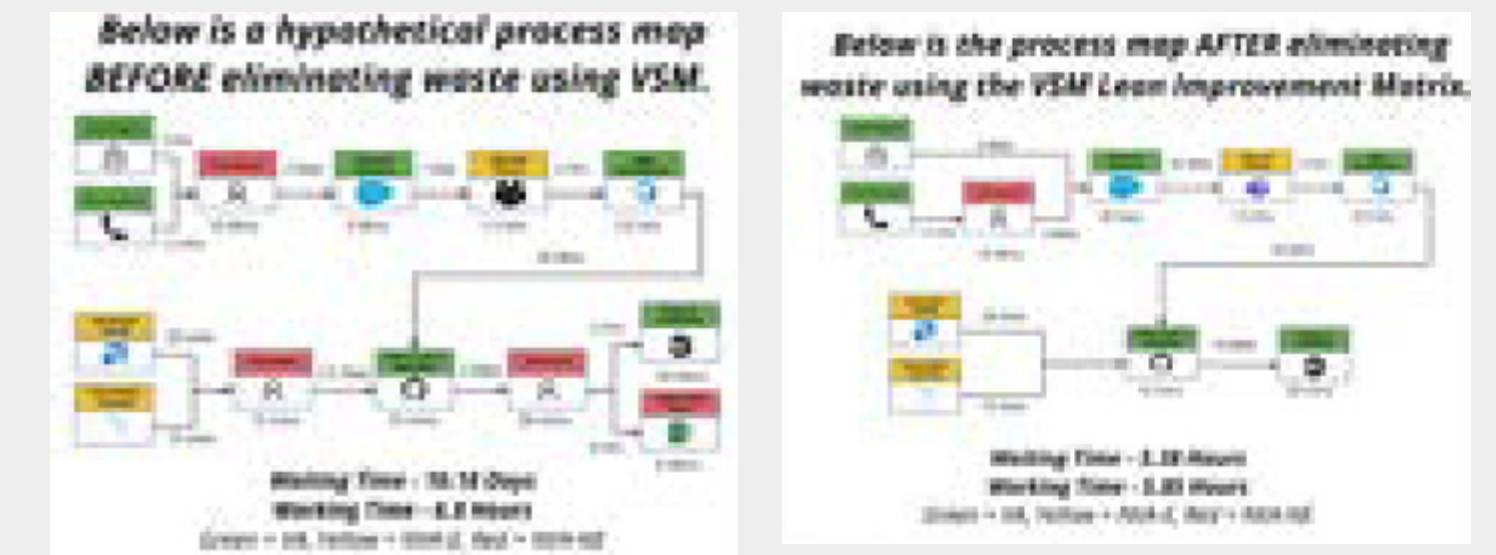
### Examples:

- Regulatory approvals
- Compliance
- Contractual Obligations
- Security

So the question before you, when you start using value stream management and specifically value analysis, is how to identify value-added activities. You can dissect a process and ask the following questions:

- Does the activity transform the form, feature, feeling, and function the customer is willing to pay for?
- Is it being done right the first time?
- Is this something the customer expects to pay for?

Identifying the VA and NVA enables IT business organizations to improve productivity by at least 20-40% with a 10-30% reduction in the delivery time of new applications — ultimately resulting in cost savings and customer satisfaction.



Identifying and optimizing value-added and non-value-added activities helps software development teams understand how long the planned activities will take to complete and how much NVA is present in the scheduled activities. As a result, it helps to improve time to market, quality, cost, efficiency, and effectiveness of overall IT operations.

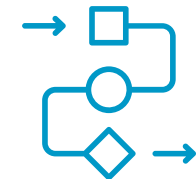
**Waste is often intangible and difficult to identify in IT business processes. For example, delays are a significant category of waste/NVA in the IT services business value chain (i.e., searching for specific information, NVA reviews, complex approval processes, slow response of applications, delays between coding and testing, aging of service tickets, delayed response time from the IT help desk). Productivity and delivery time can be significantly improved by addressing the root causes of delays.**



# How Are Organizations Implementing Value Stream Management to Reduce Noise in Their Value Streams?

VSM is gaining popularity among organizations as a means to effectively manage and enhance their value streams. One critical aspect of VSM involves the identification and reduction of noise within the value stream. Here's how organizations are leveraging VSM to achieve this goal while also improving visibility, alignment, and efficiency:

## Mapping the Value Stream:



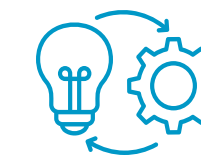
The initial step in implementing VSM is comprehensively mapping the value stream from end to end. This process enables organizations to gain clear visibility into all activities, processes, and systems involved in delivering value to customers. By visualizing the entire value stream, organizations can not only identify noise but also improve alignment across teams and systems.

## Identifying Waste:



Following value stream mapping, organizations focus on identifying and eliminating waste within the value stream. Various types of waste, including overproduction, waiting, defects, and overprocessing, can introduce noise and inefficiencies. By systematically identifying and addressing these wastes, organizations not only reduce noise but also enhance efficiency and alignment with customer value.

## Implementing Lean Principles:



VSM is deeply rooted in lean principles, emphasizing continuous improvement and waste reduction. Organizations apply lean principles such as just-in-time production, kanban, and kaizen to streamline their value streams and reduce noise. By aligning processes with customer value and optimizing flow, organizations improve efficiency and eliminate sources of noise.

## Using Automation:



Automation plays a pivotal role in reducing noise within the value stream by minimizing manual processes and mitigating the risk of human error. Organizations leverage automation to streamline repetitive tasks, enhance efficiency, and improve visibility into workflow processes. By automating key aspects of the value stream, organizations not only reduce noise but also increase alignment and efficiency.

Organizations are harnessing Value Stream Management to reduce noise in their value streams while concurrently enhancing visibility, alignment, and efficiency. By strategically mapping the value stream, identifying waste, implementing lean principles, and leveraging automation, organizations can optimize their processes, deliver higher-quality products and services, and drive sustainable growth and profitability.



## CHAPTER 4:

# Pattern Thinking

## An Intro to Pattern Thinking

Pattern thinking is a cognitive approach that involves identifying and understanding recurring patterns in a particular domain and then using that knowledge to improve processes, systems, and outcomes. Pattern thinking can be applied to many domains, including software development, business management, and scientific research.

Pattern thinking focuses on identifying common patterns or structures that underlie a particular problem or situation. These patterns can be visual, such as diagrams or flowcharts, or conceptual, such as mental models or frameworks. By identifying these patterns, individuals can better understand the problem or situation and develop strategies to address it.

Pattern thinking is particularly useful when problems are complex or ill-defined, as it can help individuals make sense of the situation by identifying underlying structures and relationships. It can also help individuals develop innovative solutions to problems, as they can draw on a broader range of knowledge and expertise.

Pattern thinking is a powerful tool for process improvement and automation. It enables organizations to recognize patterns in their processes and use them to identify opportunities for improvement. By recognizing patterns, organizations can make better decisions about how to optimize their value stream performance.

Pattern recognition helps organizations understand the relationships between different process components, allowing them to identify areas for potential improvement. This knowledge can be used to automate certain processes or create more efficient workflows that reduce waste and improve productivity. Additionally, pattern thinking can help organizations anticipate problems before they happen, allowing them to take proactive steps toward avoiding costly disruptions or delays.

Pattern thinking is a valuable cognitive approach that can help individuals improve their problem-solving skills, understand complex problems more deeply, and develop innovative solutions to challenging problems.





Here are some key reasons why pattern thinking is important for software development and delivery today:

### Improved Problem-Solving:

By identifying patterns and structures within software development and delivery processes, developers and teams can better understand the problems they are facing and develop more effective solutions.

### Greater Efficiency and Productivity

By identifying and addressing wasteful patterns in software development and delivery, organizations can improve their teams' efficiency and productivity, leading to faster and more reliable software delivery.

### Better Collaboration and Communication

By using pattern thinking to identify common patterns and structures across teams and departments, organizations can improve collaboration and communication, which can help to prevent misunderstandings, errors, and delays.

### Innovation and Creativity

By identifying patterns and structures, developers and teams can also gain new insights and perspectives that can lead to innovative solutions and software development and delivery approaches.

### Adaptability and Agility

Pattern thinking also helps organizations be more adaptable and agile in changing market conditions and customer needs. By understanding the underlying patterns and structures of software development and delivery, organizations can more easily pivot and adapt to new challenges and opportunities.

Overall, pattern thinking is a valuable cognitive approach that can help organizations to improve their software development and delivery processes, reduce waste, and achieve faster and more reliable software delivery.





## Pattern Thinking in Value Stream Management

As humans, our brains are wired to find patterns in the world around us. From spotting familiar faces in a crowd to recognizing the rhythm of a song, pattern recognition is an essential aspect of our daily lives. But have you ever considered how this skill can be applied to business operations? In value stream management, pattern thinking can play a crucial role in identifying inefficiencies and improving productivity.

When it comes to value stream management, recognizing patterns is an important skill that can help organizations identify inefficiencies and optimize their processes. Patterns are often repeated sequences of events or behaviors that signal a problem or opportunity for improvement.

One way to recognize patterns in a value stream is through data analysis. By collecting and analyzing data on key performance indicators (KPIs), such as lead time, cycle time, and throughput, teams can identify trends and anomalies that may indicate a pattern.

Another approach is through direct observation. Value stream mapping allows teams to visualize the flow of materials, information, and people across the entire process. This helps them identify bottlenecks or other areas where work tends to pile up, indicating a potential pattern.

In addition to these quantitative methods, qualitative approaches like Lean Six Sigma also emphasize the importance of observing behavior patterns among team members. For instance, if certain employees always take longer breaks than others or frequently make mistakes in their work, this could be indicative of deeper underlying issues in the process.

Recognizing patterns requires analytical skills and an understanding of human behavior within the context of workflows. Organizations can improve their efficiency and productivity over time by identifying these patterns early on and addressing them proactively rather than reactively.





# Benefits of Using Pattern Thinking in Value Stream Management to Accelerate Software Delivery

Pattern thinking is an essential skill to have when it comes to value stream management. It allows you to identify trends and patterns in data that may not be immediately apparent, helping you make more informed decisions about your processes.

By recognizing patterns in a value stream, you can quickly pinpoint areas that need improvement and focus on making changes that will have the most significant impact. You'll also be able to identify potential problems before they become major issues and take proactive steps to prevent them from occurring.

Ultimately, taking a pattern-based approach to value stream management can help organizations achieve greater efficiency, reduce waste, and deliver better results for their customers. So if you're looking for ways to optimize your business processes and improve your bottom line, start thinking in patterns today!

Here are some benefits of pattern thinking in VSM:

## Improved Problem-Solving

---

Pattern thinking can help developers identify recurring patterns and structures in software development processes, leading to better problem-solving and more effective solutions.

## Innovation and Creativity

---

Pattern thinking can help developers gain new insights and perspectives on software development, leading to more innovative and creative solutions.

## Greater Efficiency and Productivity

---

By identifying and addressing wasteful patterns in software development, organizations can improve their teams' efficiency and productivity, leading to faster and more reliable software delivery.

## Improved Quality

---

By identifying and addressing patterns of defects and rework, pattern thinking can help to improve the quality of software development processes and reduce the risk of defects in the final product.

## Better Collaboration and Communication

---

Pattern thinking can help to improve collaboration and communication across teams and departments, by identifying common patterns and structures that can be used as a basis for shared understanding and collaboration.

## Increased Adaptability and Agility

---

Pattern thinking can help organizations to be more adaptable and agile in the face of changing market conditions and customer needs by providing a foundation of common patterns and structures that can be used to pivot and adapt to new challenges and opportunities quickly.



# How to Recognize Patterns in the Software Delivery Value Stream Using Systems Thinking and Lean to Control Flow

Recognizing patterns in the software delivery value stream is an important step toward improving the efficiency and effectiveness of the software development process. It helps with controlling flow. Systems thinking and lean principles can be used to identify these patterns and make targeted improvements to the value stream.

Here are some steps for recognizing patterns in the software delivery value stream using systems thinking and lean:

## 01 | Map the Value Stream

Start by mapping out the full software delivery value stream, from requirements gathering to deployment and maintenance. This will help you understand the various steps involved and identify any potential bottlenecks or inefficiencies.

## 02 | Analyze Feedback Loops

Look at the feedback loops within the value stream to determine how information flows between different parts of the process. Identifying any delays or breakdowns in these feedback loops can help you identify patterns and opportunities for improvement.

## 03 | Use Data to Identify Trends

Collect data on key metrics such as lead time, cycle time, and throughput across the value stream. Use this data to identify trends and patterns over time, including areas where performance is consistently low or high.

## 04 | Apply Lean Principles

Use lean principles to identify sources of waste and inefficiency in the value stream. For example, look for areas where work is being duplicated or where tasks are being repeated unnecessarily.

## 05 | Apply Systems Thinking

Use systems thinking to analyze the interactions between different parts of the value stream. Identify areas where changes in one part of the system have unintended effects on other parts.

By following these steps, organizations can identify patterns in the software delivery value stream and make targeted improvements to optimize the process for efficiency and effectiveness. These improvements can help deliver higher-quality software products faster and cheaper.





# How to Start Thinking Patterns to Reduce Noise That Leads to Non-Value-Added Activities

Software teams should start thinking in patterns for several reasons:

## Identifying Best Practices

Patterns are based on best practices that have been proven to work in a particular context. By identifying and applying these patterns, software teams can benefit from the accumulated knowledge and experience of others.

## Improving Efficiency

By identifying common patterns and structures in software development processes, teams can streamline their workflows, reduce waste, and improve efficiency.

## Enhancing Communication

Patterns provide a common language and framework for discussing and sharing knowledge about software development processes. This can enhance communication and collaboration between team members, as well as with stakeholders outside the team.

## Promoting Consistency

By using patterns, teams can promote consistency and standardization in their software development processes, which can help to reduce errors, improve quality, and increase reliability.

## Enabling Continuous Improvement

Patterns can be a starting point for continuous improvement. By regularly reviewing and refining patterns, teams can identify areas for improvement and implement changes that lead to better outcomes.

Thinking in patterns can help software teams to work more efficiently, communicate more effectively, and consistently deliver high-quality software. By leveraging established patterns and continually refining their own, teams can improve their processes, reduce waste, and continuously improve the value they deliver to their customers.





# Uncovering Patterns in Software Delivery Value Streams

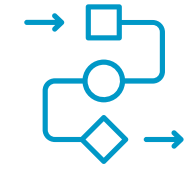
Uncovering patterns in a software delivery value stream involves a combination of analysis, observation, and collaboration. Here are some steps that can be taken to uncover patterns in a software delivery value stream:

## Define the Value Stream



The first step is to define the value stream that you want to analyze. This involves identifying the key stages of the software delivery process, such as planning, development, testing, and deployment.

## Map the Value Stream



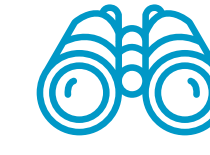
Once you have defined the value stream, you need to map it out in detail. This involves identifying the inputs, outputs, and handoffs between each stage of the value stream, as well as any delays, bottlenecks, or waste that may be present.

## Analyze the Data



Once you have mapped out the value stream, you must analyze the data to identify patterns and trends. This involves looking for common themes, such as recurring delays or bottlenecks, and any correlations between different stages of the value stream.

## Observe the Process



In addition to analyzing the data, it is important to observe the software delivery process in action. This can help identify patterns or structures that may not be immediately apparent from the data alone.

## Collaborate with Stakeholders



It is important to collaborate with stakeholders, such as developers, testers, and project managers, to gain their insights and perspectives on the software delivery process. By working together, you can uncover patterns and structures that may not be visible from a single perspective.

By following these steps, organizations can uncover patterns in their software delivery value streams and identify areas for improvement. This can help to reduce waste, improve quality, and achieve faster and more reliable software delivery.



## Wasteful Patterns in Software Development and Delivery

There are so many wasteful patterns in software development and delivery that can slow down the flow of value and reduce the efficiency and effectiveness of the development process. Here are some of the most common wasteful patterns:

**Overproduction** occurs when developers produce more features, code, or documentation than is needed. Overproduction can lead to time, resources, and money waste.

**Waiting** occurs when developers are waiting for work to be completed by other teams or waiting for feedback or approvals from stakeholders. Waiting can cause delays and can result in idle time for developers.

**Defects and rework** occur when code or features contain defects or bugs, which must be fixed or reworked before they can be released. Defects and rework can be time-consuming and cause delays in software release.

**Handoffs** occurs when work is passed from one team or person to another, which can lead to delays, miscommunications, and errors.

**Unused talent** occurs when the skills and expertise of developers are not fully utilized, which can lead to lower productivity and decreased job satisfaction.

**Overprocessing** occurs when developers spend too much time on non-value-added activities, such as creating excessive documentation or performing unnecessary testing.

**Motion** occurs when developers must move between different systems, tools, or locations to complete their work, which can cause delays and reduce efficiency.



# Value Stream Automation

## An Intro to Value Stream Automation

Value stream automation is a key part of value stream management that helps organizations to optimize the software delivery value stream by automating repetitive and time-consuming tasks, from planning to when the customer uses the product. It aims to improve the efficiency of the software delivery process by eliminating manual processes, reducing the occurrence of errors, and improving communication and collaboration among team members.

It enables a more agile and efficient value stream, allowing teams to identify and address issues quickly and effectively. It can also help reduce the noise in the value stream by automating repetitive tasks (necessary non-value-added activities) and freeing up team members to focus on value-added activities.

It involves mapping out the entire value stream of a process, identifying areas where waste occurs, and implementing automation solutions to eliminate those inefficiencies.

One key aspect of value stream automation is its focus on adding value for customers. This means that every step in the process should be evaluated to ensure it contributes directly or indirectly to delivering customer satisfaction.

Businesses can automate processes from end to end, reducing errors, improving quality control, and freeing employees' time for more strategic work.

Value stream automation has become increasingly popular in recent years due to its ability to streamline operations while allowing companies to scale their business quickly without hiring additional staff. By embracing this approach, organizations can create efficiencies across all departments – from sales and marketing through supply chain management – optimizing their entire business operation.





# Pattern Thinking and Value Stream Automation

Pattern thinking and value stream automation can be powerful tools for improving flow in software development. Organizations can improve the speed, efficiency, and quality of their software delivery by using pattern thinking to identify recurring patterns and structures in the development process and then applying value stream automation to streamline and optimize those processes.

Here are some key steps to follow when using pattern thinking and value stream automation to improve flow in software development:

**Identify key value streams:** Begin by identifying the key value streams in your software development process. This may include coding, testing, deployment, and support processes such as documentation and training.

**Identify patterns:** Use pattern thinking to recognize recurring patterns and structures within those processes. For example, you may identify patterns of handoffs, delays, or bottlenecks slowing down the flow of value.

**Define process improvements:** Based on your analysis, define process improvements that can help to streamline and optimize your value streams. This may include automating repetitive tasks, reducing handoffs, or improving team communication and collaboration.

**Implement automation tools:** Once you have defined your process improvements, implement automation tools that can help to automate and optimize your value streams. For example, you may use continuous integration and continuous deployment (CI/CD) pipelines to automate the build, testing, and deployment process.

**Monitor and measure progress:** Finally, monitor and measure your progress to ensure your process improvements have the desired effect. Use metrics such as cycle time, lead time, and defect rates to track your progress and identify areas for further improvement.

Pattern thinking is a powerful tool for improving value stream performance. By leveraging data analytics and data visualization, companies can identify patterns in their operations and use them to understand their business processes better and optimize their value streams.

Pattern thinking involves analyzing data to uncover and understand the patterns that exist in your operations. It helps you identify trends, predict outcomes, and make informed decisions about improving your value stream performance. With pattern thinking, you can identify areas of opportunity for improvement, develop strategies for optimizing your processes, and measure the impact of changes on your bottom line. By utilizing pattern thinking in your value streams, you can ensure that you get the maximum return on investment from your efforts.

By using pattern thinking and value stream automation to improve flow in software development, organizations can achieve faster and more reliable software delivery while improving the quality and value of their products and services. The combination of pattern thinking and value stream automation can bring significant value to software development, including improved quality, efficiency, collaboration, visibility, and adaptability. By leveraging these approaches, organizations can streamline their software development processes, reduce waste, and achieve faster, more reliable, and higher-quality software delivery.



## How Value Stream Automation Helps Accelerate Value-Added Activities

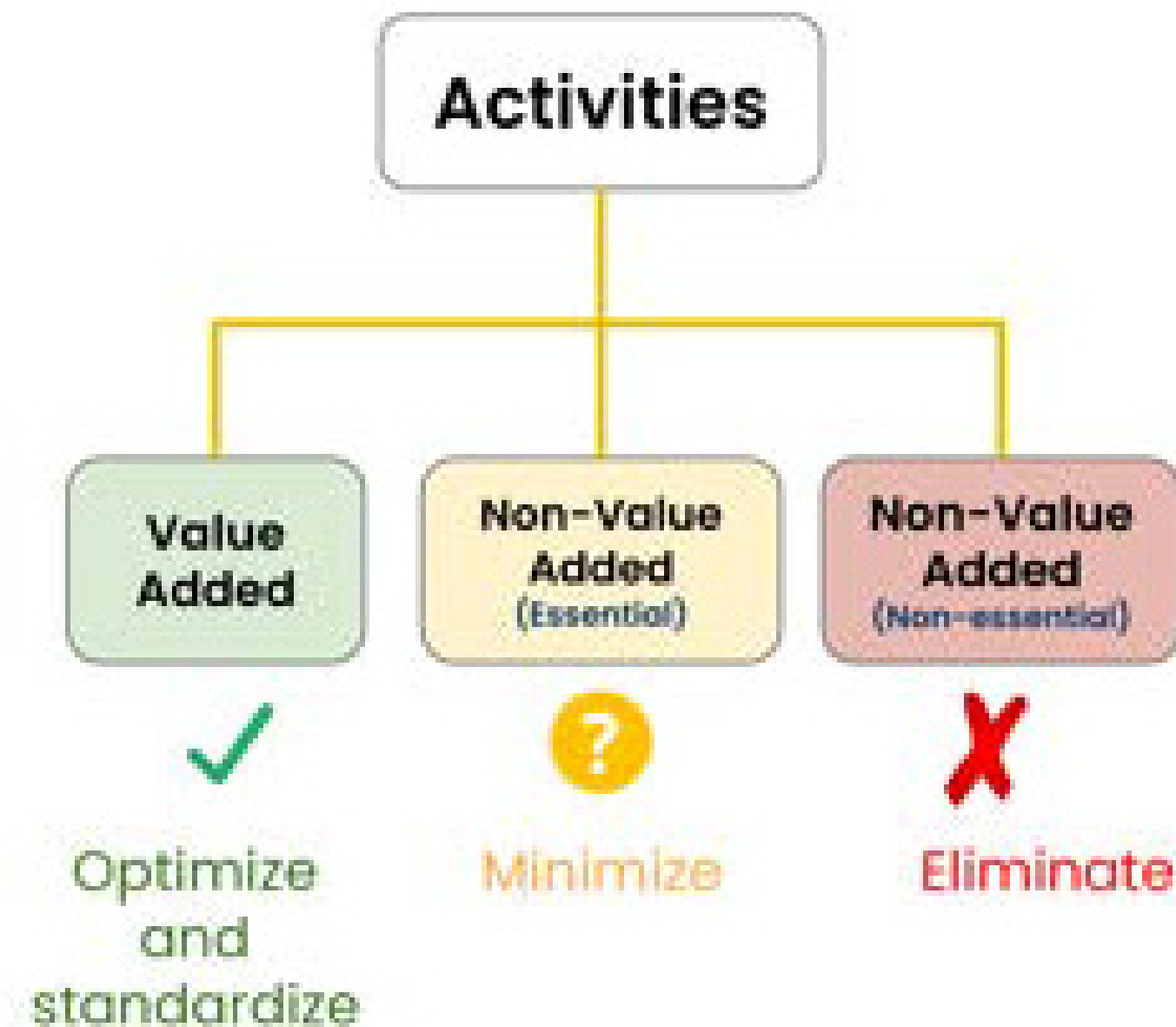
One of the most significant advantages of value stream automation is that it can accelerate value-added activities.

By automating tasks that don't add value, companies can free up resources to focus on what matters most - creating products and delivering services that meet customer needs. This not only improves efficiency but also leads to better outcomes for customers.

Value stream automation achieves this by eliminating manual processes, reducing errors and defects, improving team communication, and creating process transparency. Doing so enables organizations to identify bottlenecks in their systems quickly and make necessary changes to improve flow.

Moreover, by automating mundane tasks like data entry or report generation, employees can spend more time on tasks that require creativity or critical thinking skills. This leads to higher job satisfaction among workers who feel empowered by using their skills effectively.

Value Stream Automation offers significant benefits beyond just efficiency gains; it accelerates important activities while ensuring quality output, giving businesses a competitive advantage in today's fast-paced market environments.

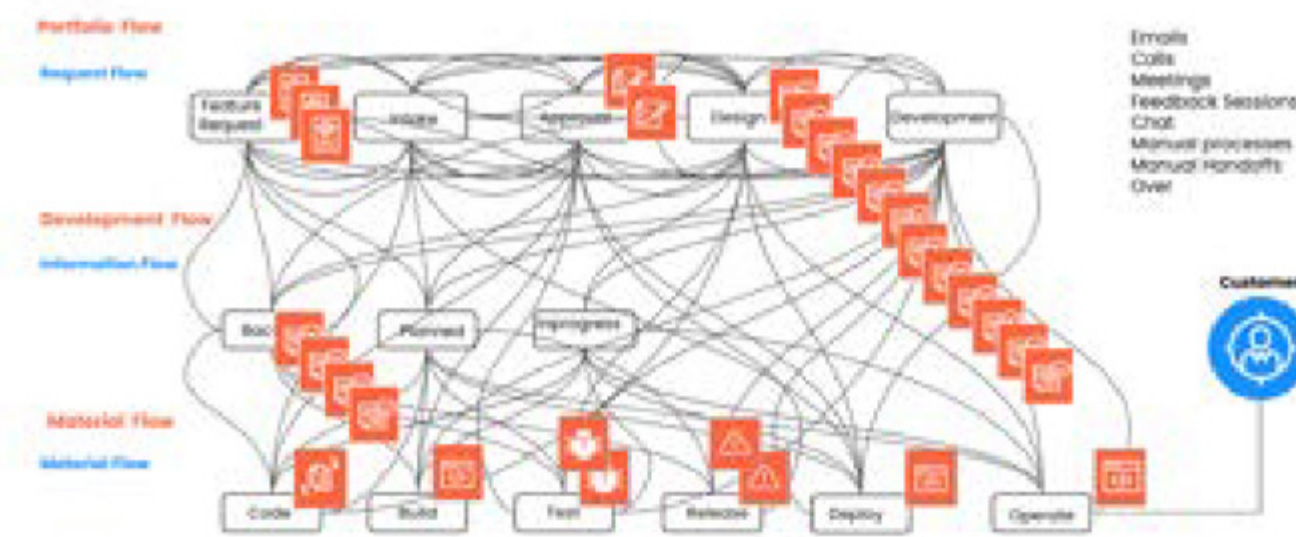




## Automation Patterns and How It Helps to Reduce Noise

Automation patterns are pre-defined solutions that can be used to solve a range of problems in value stream management. These patterns enable software teams to automate repetitive tasks and streamline workflow, allowing faster delivery and higher-quality outputs.

Automation patterns can also reduce noise by providing visibility into the delivery process. By using tools such as dashboards and analytics, teams can gain insights into how their software is being delivered and identify areas for improvement. This can reduce communication overhead and improve team collaboration.



Above is a depiction of a noisy software delivery value stream, excessive feedback, communication, and non-value-added activities, leading to delays, bottlenecks, errors, and inefficiency.



Above is a depiction of a software delivery value stream when necessary non-value-added activities are automated and the noise coming from feedback and communication is reduced.



# How to Start Implementing Automation Patterns

Implementing automation patterns in your software delivery value stream requires careful planning and execution. Here are some steps that you can follow to get started:

**1 | Identify areas of your delivery process that would benefit from automation.** This may include building and testing code, deploying releases, or monitoring application performance.

**2 | Define your workflows and processes.** To ensure that your automation implementation is effective, you must have clearly defined workflows and processes. This includes specifying each task's inputs, outputs, and dependencies of each task in your delivery pipeline.

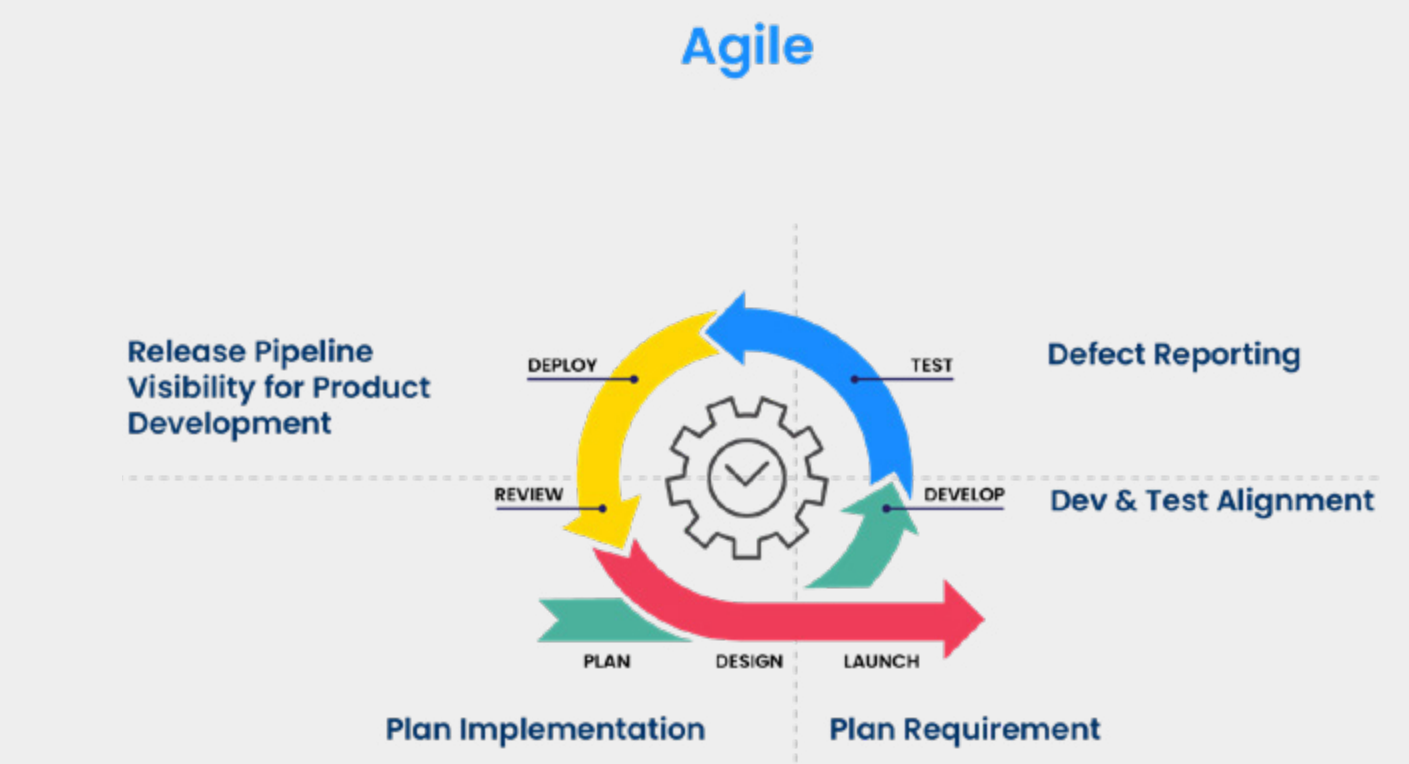
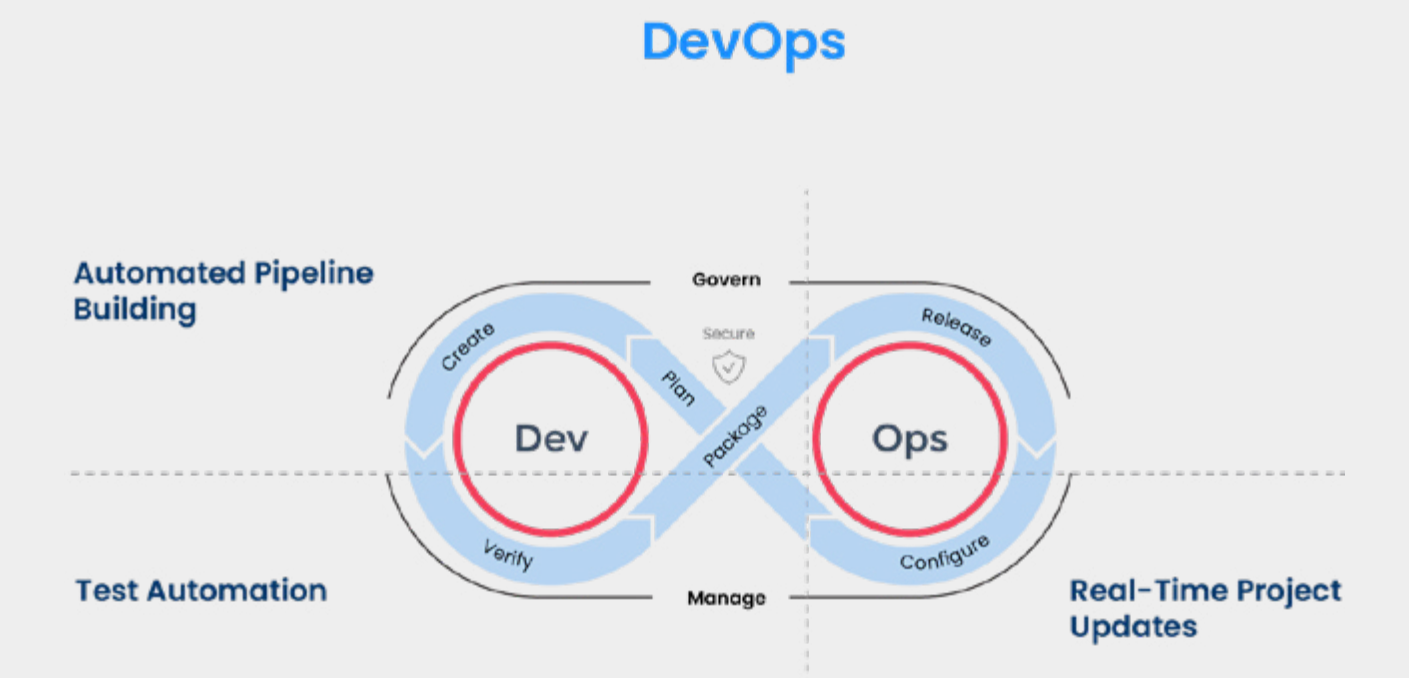
**3 | Automate a small part of your delivery process first.** Decide what kind of automation you'd want to implement. Start by automating a single task or function and gradually build up to more complex workflows. This allows you to find your automation approach before scaling up.

**4 | Monitor and optimize your automation implementation.** Once your automation solution is up and running, you should regularly monitor its performance and make adjustments as necessary. This includes optimizing workflows, resolving issues, and utilizing new automation features and capabilities.

By following these steps, you can begin implementing automation patterns in your software delivery value stream and enjoy the benefits of reduced noise, increased efficiency, and improved software quality.



# Examples of Patterns in DevOps and Agile and How to Use Automation to Accelerate Software Delivery





# How a Value Stream Management Platform Can Help with Value Stream Automation and Pattern Recognition

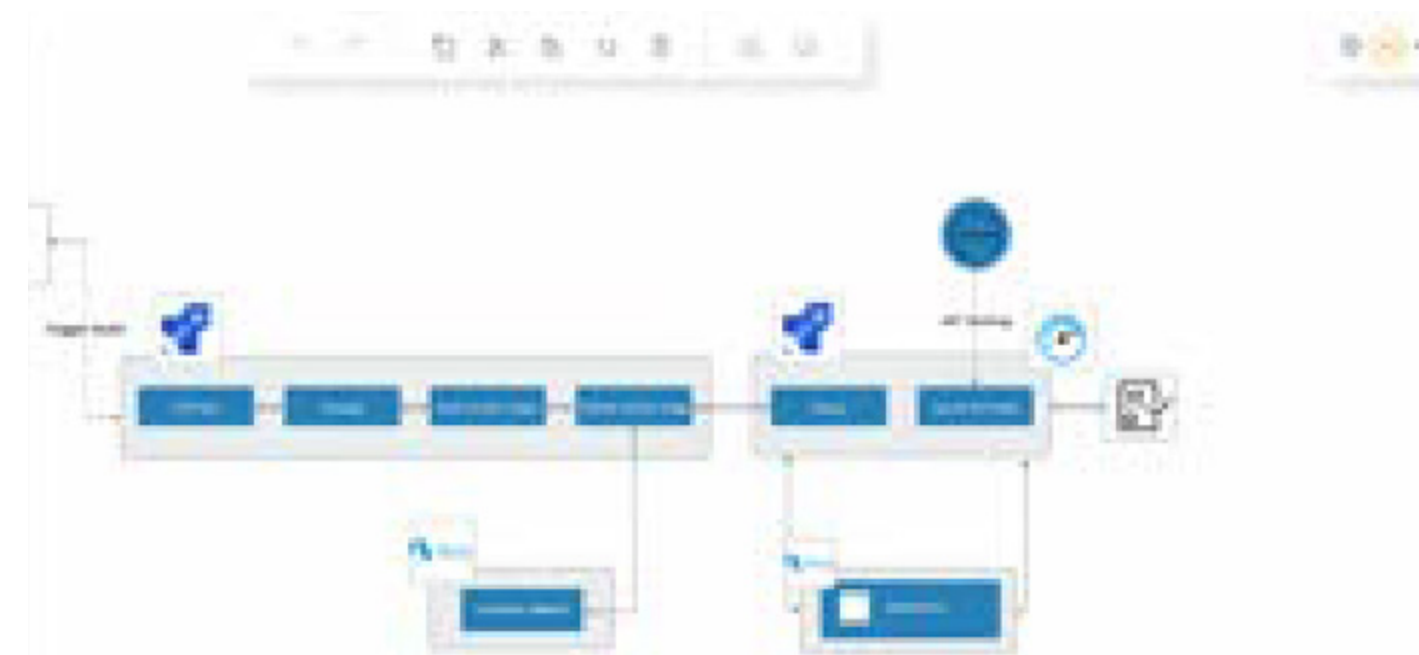
A Value Stream Management Platform (VSMP) serves as a central hub for optimizing software delivery processes, empowering organizations to deliver valuable software efficiently. By connecting various software development tools, VSMPs break down silos and enhance cross-functional visibility. They enable users to visualize the flow of work, measure key metrics, automate workflows, and identify patterns for continuous improvement.

## Key Features of an Ideal VSM Platform:

### Gain Visibility into Your Value Stream

Why is it crucial to have complete visibility into your software delivery value stream? You can't manage what you can't see. Effectively managing your software delivery value stream requires comprehensive visualization. Leveraging the core capabilities of a VSM platform, organizations can identify and visualize each step within the value stream, facilitating effective planning and management of value-added and non-value-added work.

**Visual Layout:** Mapping the value stream is foundational in VSM. An ideal VSM platform facilitates mapping every step within the value stream along with associated people, processes, and tools, fostering visibility and understanding across individual value streams and the entire portfolio.



- Identify people, processes, and technology
- Visualize the flow of activities
- Understand tool connections and manual processes
- Facilitate communication and collaboration
- Commence planning for future state



## Align Your Value Stream

Measurements and analytics allow tracking of interconnectedness up and down the software delivery value stream. A VSM platform enables capturing and interpreting real-time data, organizing it for display on interactive metrics dashboards, empowering effective task prioritization, evaluation of necessary changes, and impact assessment.

**Common Data Model:** A standardized data model collects and organizes disparate data from tools across the value stream, providing a comprehensive view.

**Measurement and Analytics:** Real-time metrics collection aids in implementing necessary changes, continuously improving processes, and enhancing efficiency. Insights Analytics feature offers a range of metrics covering Agile, DevOps, and value metrics, with pre-loaded Quickstart metrics to expedite software delivery.



## Optimize Your Value Stream

A VSM platform streamlines operations by automating processes, ensuring efficiency throughout software development and delivery pipelines. Empowering automation of synchronization, triggering, monitoring, and data collection eliminates waste and maximizes efficiency. Additionally, it facilitates pattern recognition for identifying recurring inefficiencies and opportunities for improvement.

- Connect tools for collaboration, governance, and monitoring
- Eliminate waste in both value-added and non-value-added processes
- Establish feedback loops or integrations for synchronization
- Facilitate interoperability by creating approval processes and enabling seamless tool interactions





## CHAPTER 6:

# Conclusion

As previously discussed and emphasized, leveraging pattern thinking in value stream management serves as a powerful tool in addressing the persistent challenges and inefficiencies within the software delivery value stream. By identifying recurring patterns of waste and inefficiencies, organizations can proactively eliminate such noise, thereby optimizing their processes and enhancing the overall flow of work from conception to production.

It's evident now more than ever that value stream management offers end-to-end visibility and control over multiple interconnected value streams across an organization's portfolio. This heightened visibility enables organizations to foster cross-team collaboration, enforce governance, and implement automated workflows. Through this holistic approach, organizations can expedite the software delivery process while concurrently elevating software quality.

By adopting a pattern-centric mindset in value stream management and leveraging value stream metrics, organizations can effectively measure and track work within end-to-end product value streams. This enables both technology and business leaders to gain valuable insights, enhance communication, and drive continuous improvement initiatives.

Overall, implementing value stream management, particularly through the utilization of value stream automation patterns, enables organizations to align business objectives with IT work seamlessly. This alignment facilitates the delivery of high-quality software to customers at an accelerated pace, all while mitigating risk and uncertainty.

To find out more about automation patterns and how that can reduce noise and make software delivery more predictable, [contact us](#).