ValueOps™
by Broadcom

4 Easy Steps to **Mastering Chaos with Value Stream Management**

ValueOps™
by Broadcom

Introduction

For many of today's software development and delivery teams, chaos reigns.

These teams are contending with countless variables, constant flux, redundant data sets, and many other hurdles. All these factors mean software delivery is noisy, overwhelming, unpredictable.

We're happy to report that there is a better way.

In this ebook, we offer four steps you can take to eliminate the chaos—and establish clarity, order, and efficiency.

STEP 1
Silence the Noise in Software Delivery

Software development and delivery is a complex process that involves multiple teams, tools, and techniques. In this intricate web of activities, noise presents a constant threat, one that can slow down or derail your software development and delivery.

In this context, noise refers to interruptions that prevent your teams from doing the actual work. This creates variation in the flow of information, artifacts, or processes, and these variations can have a negative impact on efficiency and effectiveness. This can include delays, errors, rework, bottlenecks, poor communication, or other inefficiencies.

TYPES OF NOISE

In software development and delivery, noise can take many forms. Here are four types of noise:

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. Fundamentally, this feedback does not provide insights that are actionable or relevant to making improvements.

PROCESS NOISE

This type of noise occurs when there are inefficiencies in the software development process. Manual processes, bottlenecks, technical debt, and unclear requirements can all be the culprit for slow software delivery.

TECHNICAL NOISE

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

EFFECTS OF NOISE

When the tools, people, and processes in software development and delivery 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:

EFFECTS OF NOISE

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.

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.

MISCOMMUNICATION

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

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.

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.

Requirements

It is essential to establish clear workflows, roles, and communication channels to mitigate the effects of noise. Teams should also strive to use integrated toolchains that minimize the need for manual intervention or data transfer. As we'll detail below, approaches like automation, workflow simplification, and the adoption of integrated toolchains are key strategies for eliminating noise and improving overall performance.

ValueOps™
by Broadcom

STEP 2
Apply Value Stream Management

DEFINITION

Within your organization, optimizing your value streams represents a key imperative. Quite simply, value streams represent all the steps your organization has to take to go from the inception of an offering to its delivery to the customer.

Value Stream Management (VSM) represents a strategy for maximizing the efficacy of these efforts. VSM helps your organization maximize efficiency and productivity. By employing VSM, teams can gain improved visibility into the value stream, align teams around common goals, and optimize workflows.

VSM is a 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.

To maximize the value of VSM, teams can leverage these approaches:

LEAN PRINCIPLES

Lean principles are a set of guiding philosophies and practices that have been widely adopted in software development. Through this approach, teams can 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.

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. In software delivery, systems thinking involves understanding the various parts of the software development lifecycle, including stakeholders, processes, technologies, and environments, and how they all affect each other.

Combining these approaches can create an effective VSM 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.

APPLYING SYSTEMS THINKING AND LEAN PRINCIPLES TO VSM

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

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.

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.

WASTE REDUCTION

Lean principles aim to eliminate waste in all forms, including 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.

PULL SYSTEM USAGE

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

CROSS-FUNCTIONAL TEAM COLLABORATION

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.

ValueOps™ by Broadcom

STEP 3

Harness Pattern Thinking

DEFINITION

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.

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. By identifying these patterns, individuals can better understand the problem or situation and develop strategies to address it.

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. 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.

WASTEFUL PATTERNS IN SOFTWARE DEVELOPMENT AND DELIVERY

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

OVERPRODUCTION

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

WAITING

Too often, developers have to wait for work to be completed by other teams or pause while they await feedback or approvals from stakeholders. Waiting can cause delays and can result in idle time for developers.

DEFECTS AND REWORK

When code contains defects or bugs, they have to be fixed. Defects and rework can be time-consuming and cause delays in software release.

HANDOFFS

When work is passed from one team or person to another, these handoffs can create a range of problems, including delays, miscommunications, and errors.

UNUSED TALENT

The skills and expertise of developers are often not fully utilized. This can lead to lower productivity and decreased job satisfaction.

OVERPROCESSING

Teams contend with overprocessing when developers spend too much time on non-value-added activities, such as creating excessive documentation or performing unnecessary testing.

MOTION

Motion arises when developers must move between different systems, tools, or locations to complete their work. This motion can cause delays and reduce efficiency.

ValueOps™
by Broadcom

STEPS FOR UNCOVERING PATTERNS IN THE SOFTWARE DELIVERY VALUE STREAM

Here are some steps that can be taken to uncover patterns in a software delivery value stream:

01

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.

02

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.

03

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.

04

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.

05

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.

STEP 4

Leverage Value Stream Automation

AN INTRODUCTION TO VALUE STREAM AUTOMATION

Value stream automation plays a key role in an organization's success with VSM. By automating repetitive and time-consuming tasks, organizations can optimize the software delivery value stream. Through automation, teams can improve software delivery efficiency by eliminating manual processes, reducing the occurrence of errors, and improving communication and collaboration among team members. Organizations can automate processes from end to end, reducing errors, improving quality control, and freeing employees' time for more strategic work.

ValueOps™ by Broadcom

HOW TO START IMPLEMENTING AUTOMATION

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

VALUE STREAM AUTOMATION

IDENTIFY AREAS OF YOUR DELIVERY PROCESS THAT WOULD BENEFIT FROM AUTOMATION

This may include building and testing code, deploying releases, or monitoring application performance.

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.

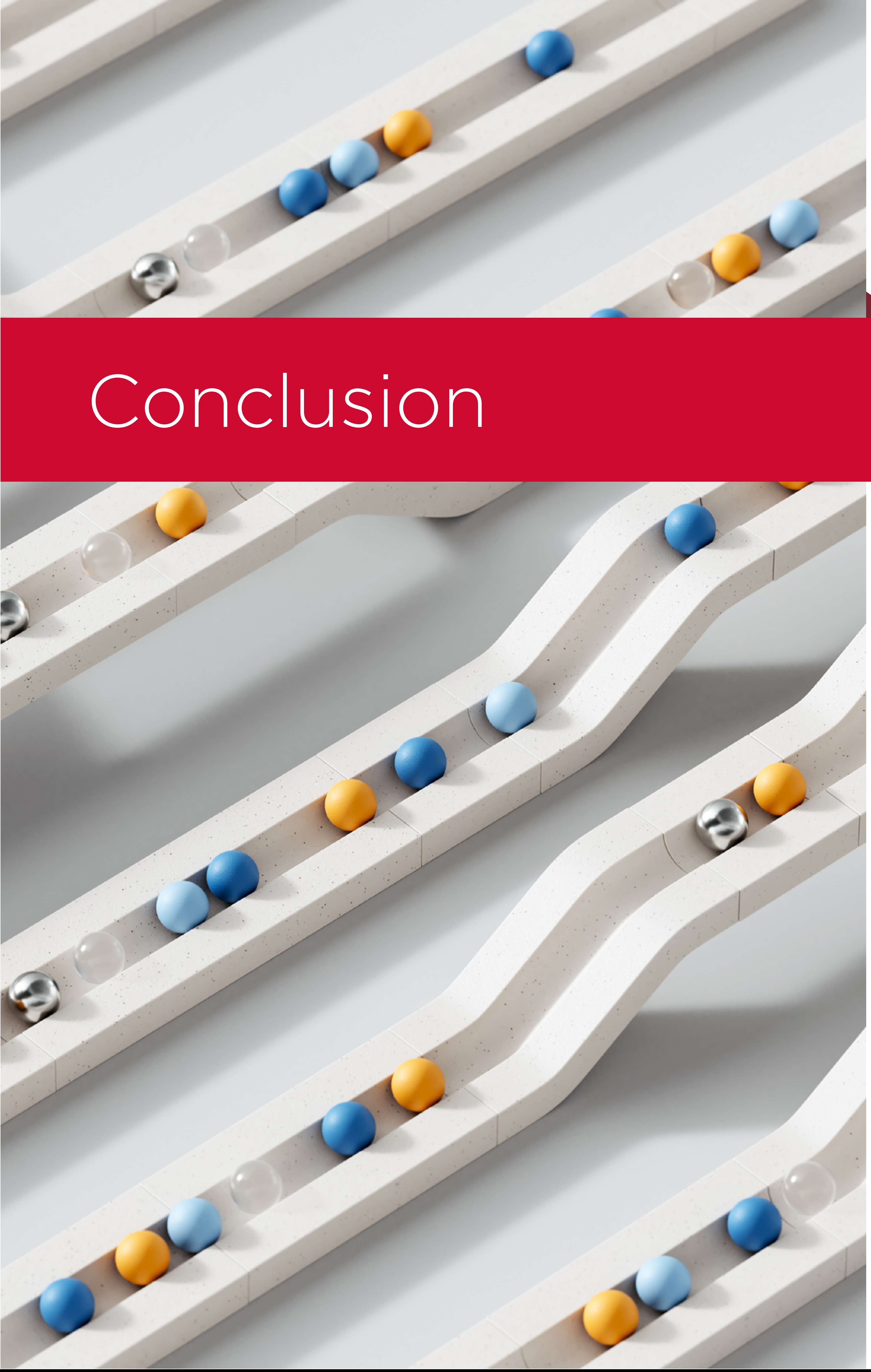
AUTOMATE A SMALL PART OF YOUR DELIVERY PROCESS FIRST

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.

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.



Conclusion

For many software development and delivery teams today, noise, chaos, and unpredictability are an all-too-familiar part of the daily landscape. These realities exact a heavy penalty from organizations, stifling speed, quality, and value. That's the bad news. The good news is that there are proven strategies that can address these hurdles. By eliminating noise, leveraging VSM and pattern thinking, and employing VSM automation, software development and delivery teams can optimize their processes and enhance the overall flow of work from conception to production. In this way, they can both speed delivery and boost software quality.

To learn more, contact us.