DevOps Foundation®

BLUEPRINT

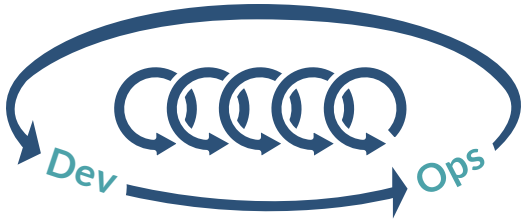
With DevOps, people across the IT organization, working together, enable fast flow, feedback and continuous improvement of planned work into production, while achieving quality, stability, reliability, availability, security and team satisfaction.

CALMS Values

C: Culture - emphasizes shared vision collaboration, communication, learning and continuous improvement.
A: Automation - CI/CD toolchains, and infrastructure-as-code enable automation, consistency, velocity and fast recovery.
L: Lean - Maximize customer value while minimizing waste and improving flow.
M: Measurement - Value-driven metrics for people, process and technology support trust and performance improvement.
S: Sharing - Leaders and teams share ideas, and skills, improve communication, collaboration and performance.

The Three Ways

1st Way: Continuous Flow
2nd Way: Feedback
3rd Way: Continuous Improvement
(Experimenting and Learning)



Principles & Practices

Frequent small releases using continuous integration, testing, delivery, deployments and monitoring reduce lead time, costs and risks.

Organization

Cross-function teams focus on business goals. Ops work with Dev, supporting each other to improve flow towards production and monitor results.

Benefits

Improved release cadence, velocity, throughput, efficiency and stability, quality, security and team satisfaction.

Related Frameworks

Agile: Lean Development
ITSM: Processes
Lean: Reduce Waste
Value Stream Management: End-to-End

Certified Agile Service Manager (CASM)®

BLUEPRINT

Agile Service Management is a framework that ensures service management processes reflect Agile values and are designed with "just enough" control and structure in order to effectively and efficiently deliver services that enable the ability to do something when and how they are needed or desired.

Scrum

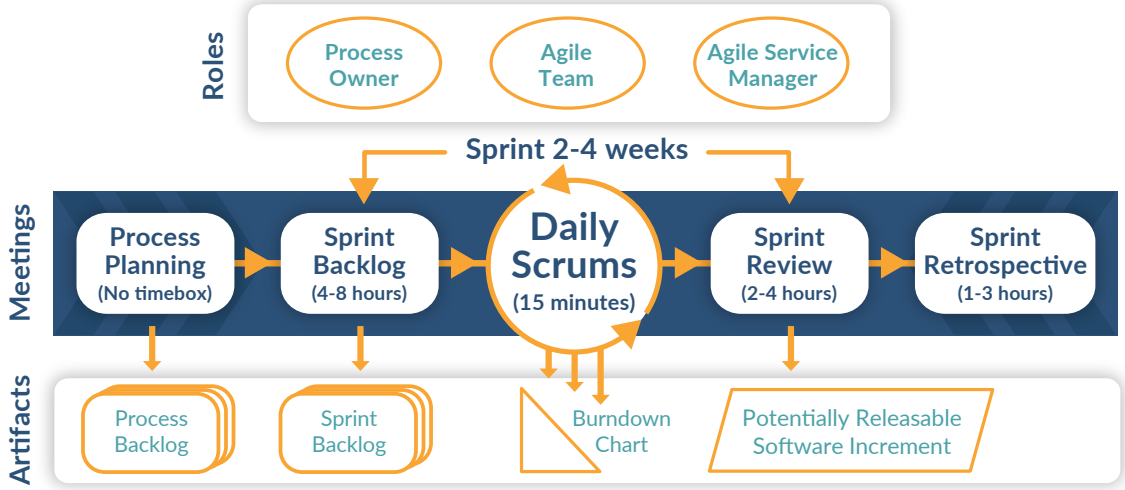
Scrum Masters, Product Owners and Developers perform Sprint Planning, Sprints, Daily Scrums, Sprint Reviews and Sprint Retrospectives to create product backlogs, sprint backlogs and potentially releasable increments.

Agile Service Manager

Is a servant leader, subject matter expert, coach and protector of the Agile Service Management Team, promotes understanding, assists and educates others on the team, assists writing user stories, facilitates events to ensure productivity and success.

Agile Manifesto

We VALUE individuals and interactions, working software, customer collaboration, and responding to change OVER processes and tools, comprehensive documentation, contract negotiations and following a plan.



Benefits

Requirements & solutions evolve iteratively, incrementally and rapidly through collaboration between self-organizing, cross-functional teams.

Agile Service Self-Managing, Cross-Functional Teams

Including Agile Practice Owner, Agile Service Manager, Customer Practitioner, Process Architect, Tool Engineer, Change Manager and Scribe turn the "What" into "How".

User Stories, Epics and Definition of Done

User Story: "As a (role), I want to (do something) so I can (achieve something)."
Epics are prioritized collections of user stories.
Definition of Done defines what a completed increment needs to be ready to release.

DevOps Leader®

BLUEPRINT

People that lead teams and organizations that are on a DevOps journey sponsor development of generative culture, support their teams and inspire actions to continuously transform their teams to higher levels of organization performance.

Transformational Leadership

Project a vision, provide intellectual stimulation, inspire collaborative communication, support specific behaviors and pro-actively recognize personal behaviors.

Becoming a DevOps Organization

Engage early adopters in small cross-functional teams with shared goals to improve flow of their value streams using small batch sizes, tools and incremental processes.

Measure to Learn

Employ value-stream mapping to visualize flow, determine metrics and current state of value-added tasks and waste to guide improvements.

Measure to Improve

Use metrics and future state value stream mapping to identify improvement opportunities in People, Process and Tools.

Unlearning Behaviors

Fearlessly let go outdated information, work without bias to enthusiastically take in new information that improves effective decision-making and improve flow of work

Models and Organization Designs

Design the organization aligned wtih the vision and improve communication between cross-functional teams using concepts from Target Operating Models, Conway's Law, SAFe, and Spotify.

Articulate and Socialize Vision

Passionately champion a vision with support from top management. Relentlessly promote changes across the organization incrementally to effect changes.

Energy and Momentum

Evangelize measurable business outcomes gained with the improved value stream while honestly contrasting prior performance.

Benefits

Well-led DevOps teams achieve more frequent, secure, quality code deployments, faster lead time from commit to deploy, faster MTTR, lower change failure rates, and team satisfaction.

DevOps Engineering Foundation℠

BLUEPRINT

Learn what DevOps engineering is, why DevOps engineering is important and how DevOps is engineered for success.

DevOps Engineering Introduction

Sound DevOps engineering depends on a foundational understanding of DevOps principles, practices, related frameworks, performance and benefits.

Ephemeral Elastic Infrastructures

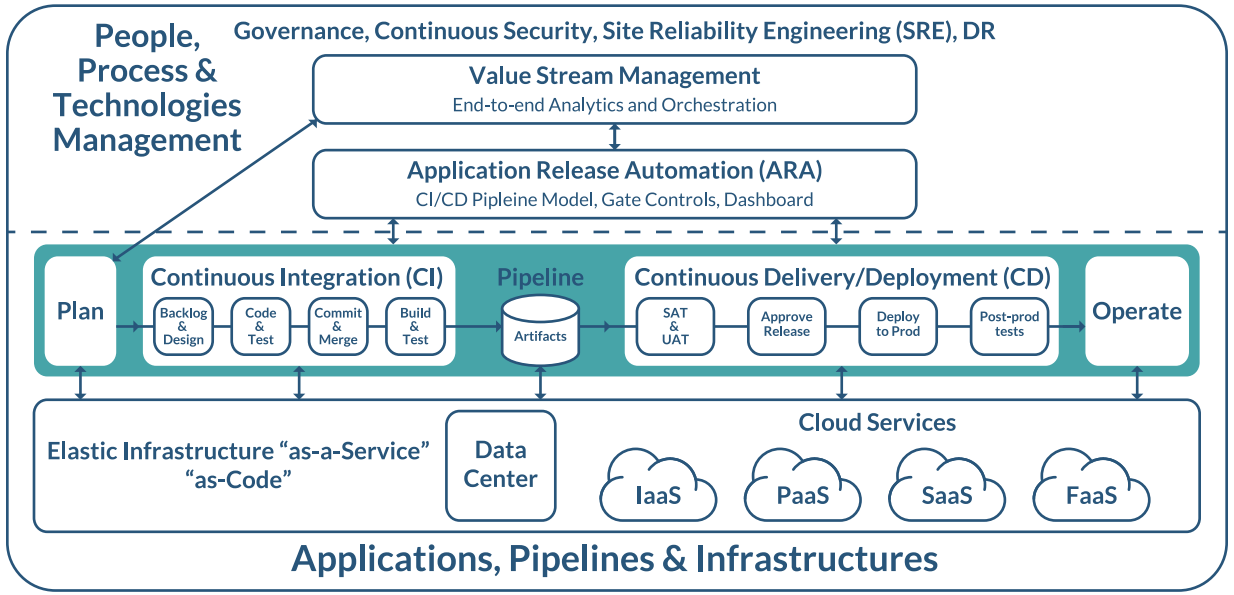
DevOps works best when engineered for virtual and cloud resources using configuration management, infra-as-code, containers orchestration, and GitOps

DevOps Technology

The technology heart of DevOps is source and artifacts control, CI/CD pipelines, tools and toolchains, application release automation, and value stream management.

Architectures & Continuous Integration

Modular applications architected as microservices, packaged in containers, work best with continuous integration processes.



Continuous Delivery & Deployment

DevOps for Continuous Delivery and Deployment requires engineering release automation and leveraging deployment strategies.

Metrics, Monitoring, Observability, Governance

Visibility and controls for DevOps requires metrics, monitoring, observability, and governance.

Continuous Testing

Fast DevOps lead times require that testing solutions be engineered around key tenets of continuous testing, strategies for test creation, TDD, test acceleration, test results, test management, and test environment management.

DevOps Engineering Humans

DevOps can not succeed by engineering technology and processes alone. The right culture, specific team topologies, commitment to continuous learning, and awareness of future DevOps trends are essential.

Continuous Delivery Ecosystem Foundation℠

BLUEPRINT

Continuous Delivery (CD) is a software engineering approach in which teams produce incremental sotware changes in short cycles ensuring that the software can be released safely at any time. A DevOps toolchain automates a continuous delivery pipeline to deliver software changes faster, more frequently, securely, with reduced cost and risk.

Continuous Delivery Concepts

Collaborative management, design practices, continuous integration, continuous testing, infrastructures, toolchains, security, monitoring and delivery / deployment practices, work on incremental changes frequently using automation and fast feedback loops.

Collaborative Culture

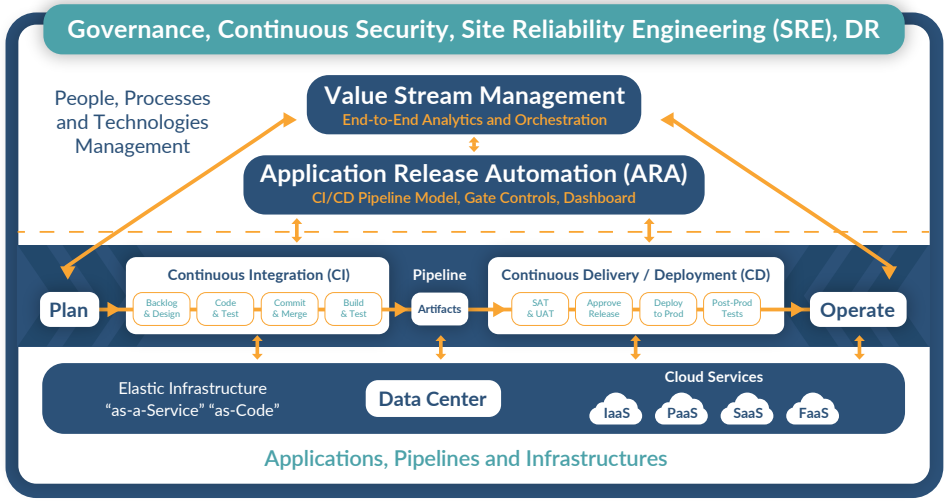
Align cross-functional lean, agile teams around business goals. Embrace "The Three Ways of DevOps". Master collaboration, affinity, effective tooling and organization changes that support increasing scale with quality built-in.

Design Practices for CD

High performance CD ecosystems use loosely coupled API-based modular service-oriented architectures (e.g. microservices) and 12-Factor Apps design practices, enable apps to be separately packaged, processed, tested and delivered in sparate images or containers.

Continuous Integration and Testing

Code changes are committed frequently to a version managed trunk branch. Images built from merged code are saved in an artifact respository. Tests conducted throughout the pipeline catch risky failures before release while completing tests quickly to avoid bottlenecks.



Infrastructure & Toolchains

Resilient, elastic inffrastructures, such as virtual and cloud-based systems defined "as-code" and "as-a-service" support on-demand, auto-scalable, immutable delployment environments. Tests such as Chaos Monkey identify failure points for improving reliability of infrastructure and toolchains.

Security Assurance (DevSecOps

Vulnerabilities are identified and fixed as a part of the pipeline by integrating security practices into team activities, processes and tool chains, including automated security scans and monitoring of code, images and deployments.

Monitoring & Improvement

Real-time active monitoring and analytics make tests, processes, and application performance measures visible for real-time decision analytics at each stage of the pipeline to prevent bottlenecks and to identify improvements.

Continuous Delivery & Deployment

Automated configuration mangement, release automation, modular packages deployed using orchestrated virtualized, containerized, applications enable deployable production-ready artifacts and deployed safely using strategies such as Blue/Green, Feature Flag and Canary.

Continuous Testing Foundation℠

BLUEPRINT

Testing, an essential portion of DevOps, is responsible for continuous assessment of incremental changes and is part of establishing a culture and environment where building, testing, and releasing software can happen rapidly, frequently, reliably and safely.

CT Concepts

All types of tests, that are mostly automated executed in production-equivalent test environments, assess the results of each stage in the end-to-end pipeline to determine if the artifacts are acceptable or need remediation before promotion.

Test Engineering Culture

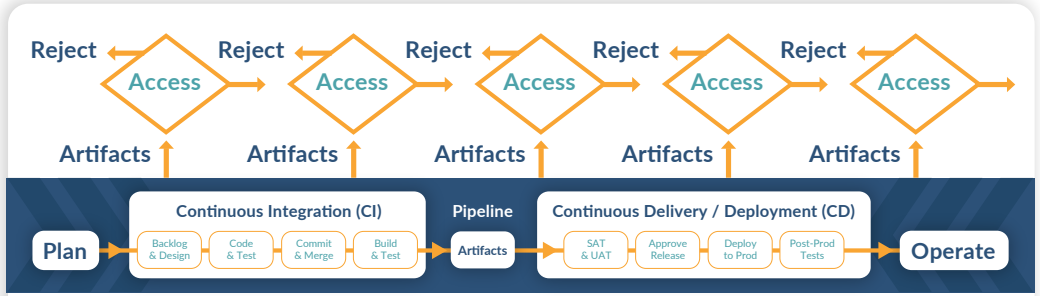
Quality is everyone's responsibility. Leaders sponsor, inspire, fund, and motivate cross-functional teams to collaborate, learn, implement, and improve continuous testing practices.

Test Strategies

Automate and trigger tests at each pipeline stage, orchestrate production-equivalent test environments, shift-left and accelerate testing as early as possible to find customer-relevant verdicts fast and early. Use A/B, Blue-Green and Canary strategies to validate user alternatives.

Test Frameworks and Tools

A test framework acts as a "backbone" for integrating and automating tasks including test plans, all types of test tools, test artifact version management, test resources, test data, tests, test schedules, test environment orchestration, test execution, test results, test reports, and test results analysis.



CT Tenets: Shift-Left, Fail Early, Fail Often, Test Fast, Relevance

Benefits

Reduced time to market, improved quality, reduced cost, improved innovation, team satisfaction.

Test Planning

Test Strategies describe test requirements, management policies, resources, topologies, automation goals and coverage methodology and assumptions for a product or service. Test Plans define test cases and priorities for each product module to meet test strategy requirements.

Test Automation

Automated tests are created per requirements described in test strategies and plans, using a test automation creation strategy such as TDD. Test and test tools are automated, orchestrated, operated and results analyzed through an API by test cases scheduled by a framework.

Test Management and Analysis

Manage resources ($, labor, schedules and flexible scaling) to meet specific CT quality, delivery and response time goals that are determined by key stakeholders.

DevSecOps Foundation (DSOF)℠

BLUEPRINT

Integrating security practices into DevOps, such as Security as Code, is a way for security practitioners to operate and contribute value with less friction. Security practices must adapt dynamically to ensure data security and privacy issues are not left behind in the fast-paced world of DevOps.

Cyber Threat Landscape (CTL)

Tactics, techniques and procedures (TTPs) describe how threat agents orchestrate and manage attacks. Threat Models optimize security by identifying objectives and vulnerabilities such as OWASP top ten, before defining counter-measures. Continuous Delivery practices are engaged to realize continuous governance, risk management and compliance.

Responsive DevSecOps Model

Security is made continuously adaptive and auditable by breaking security silos, cultivating a symbiotic relationship between security and other business units. Security specific practices and integrated toolsets as code (such as security scans) enable automated security KPIs and observable security practices into the DevOps value stream.

DevSecOps Stakeholders

Gaps between traditional waterfall security cultures and fast-paced DevOps cultures, are removed by building collaboration and trust. Through improving credibility, reliability and empathy while reducing self-interest. Decisions are based on advice from everyone affected and people with expertise using systems thinking. Shared metrics assure adaptable governance using discipline, with automation, transparency and accountability.

Realizing DevSecOps Outcomes

Security is built into the value stream efficiently with empowered development teams implementing features securely, shift-left security testing, tools for automated feedback. Culture improvements instead of policy enforcements ensure security and software engineers are continuously cross-skilling and collaborating.

Pipelines & Continuous Compliance

Security test and scanning tools are integrated into the CI/CD pipeline to finding known vulnerabilities (published CVEs) and common software weaknesses (CWEs). Repetitive security tasks are automated such as configurations, Fuzz testing and long running security tasks. Compliance as Code helps in automating compliance requirements to foster collaboration, repeatability, and continuous compliance.

DevSecOps Practices

Security is integrated into people, process, technology and governance practices. Continuous security practices for DevSecOps are implemented in onboarding processes for stakeholders. Security practices and outcomes are monitored and improved using data-driven decision making and response patterns. Lean and value stream thinking ensure that security does not cause waste, delays or constraints for flow.

Getting Started

Value Stream Mapping establishes where security activities and bottlenecks currently happen. Collaborative design of a target value state map addresses security requirements, communication and automation improvements. Scope of the design includes practices for Artifact Management, Risk Management, Identity Access Management, Secrets Management, Encryption, Governance, Risk and Compliance, Monitoring and Logging, Incident response and learning.

Learning Using Outcomes

Continuous DevSecOps learning programs are implemented to meet evolving security requirements for the organization and individuals using strategies such as lunch and learns, mentoring, professional education, employee learning plans, structured training classes, Dojos, retrospective learning, gamification, and DevOps Institute SKILup Days.

DevOps Institute

DevSecOps Practitioner℠

BLUEPRINT

DevSecOps Practitioner focuses on advanced practices, methods, techniques and tools to explore DevSecOps in your organization by looking at how people, process and technology can be combined to improve reliability and outcomes.

Advanced Basics

Understanding how to succeed, as highlighted by Malcolm Gladwell, depends on knowing not just the basics but understanding why and how those concepts matter. Exploring the fundamentals with Agile and Lean processes, learning about platforms, and knowing who to hire can make all the difference. Equally important to building teams, one must learn how to communicate among teams.

Applied Metrics

Everyone starts with basic metrics that can sometimes reach goals, but other times one needs a more concrete understanding of how to build appropriate metrics to succeed. Metrics can apply not just to production but to how one manages people and guides their process to improve success. Selecting the appropriate tools can help accelerate metrics collection and application.

Architecting & Planning for DevSecOps

Building a DevSecOps plan depends not just on espousing the right words, but having a solid plan on which one can build an effective architecture. Organizational metrics can be confusing and this area clearly builds on enterprise and API metrics across the architecture while including how to integrate effective security metrics.

Creating DevSecOps Infrastructure

Solid plans lead to solid infrastructure. Knowing how to transform an organization to cloud-native, lift and shift existing models, and integrate infrastructure as code can mean the difference between success and failure. The most effective parts of infrastructure can rely on securing gateways and end-points through analyzing how customer and internal traffic progresses securely through those areas.

Establishing a Pipeline

The key part in any good DevSecOps process relies on integrating a DevSecOps pipeline to support the overall value stream. Pipelines must not just be created by people, but then optimized for their success with good DevSecOps fundamentals using WIP across functional and non-functional areas of the pipeline. Effective pipelines contribute to secure repositories and create telemetry to feed back into the overall metric structure.

Observing DevSecOps Outcomes

The goal of DevSecOps is not just sooner and safer but establishing outcomes that contribute to organizational value. Shifting focus to create value depends on creating observability across all processes, to target metrics and build effective observational tools into the process. Different tools create different views and the observability tools one selects can affect overall outcomes.

Practical 3rd Way Applications

DevSecOps depends on not just flow and feedback but creating a continuous learning path to always improve. Knowing which areas can be improved relies on effectively collecting quantitative and qualitative data to support improvement efforts. At the same time, external events like hackathons and group sensing sessions can contribute to understanding learning effectiveness.

The Future of DevSecOps

As a cultural process, DevSecOps is here to stay. Keeping in line with DevSecOps means comprehending technical advancements such as quantum computing, bio-design and artificial intelligence. Each of these may contribute to future pipelines, but only if one applies sound innovation practices into continual learning cycles.

Site Reliability Engineering (SRE) Foundation℠

BLUEPRINT

Site Reliability Engineering (SRE) is a discipline and a role that incorporates aspects of software engineering and applies them to infrastructure and operations problems to create ultra scalable and highly reliable distributed software systems.

Culture

Reliability @ Scale, Shift-Left "Wisdom of Production", and Continuous Improvement

Toil Reduction

Reduce Non-Value Add Work using Tooling and Automation

SLAs/SLOs/SLIs

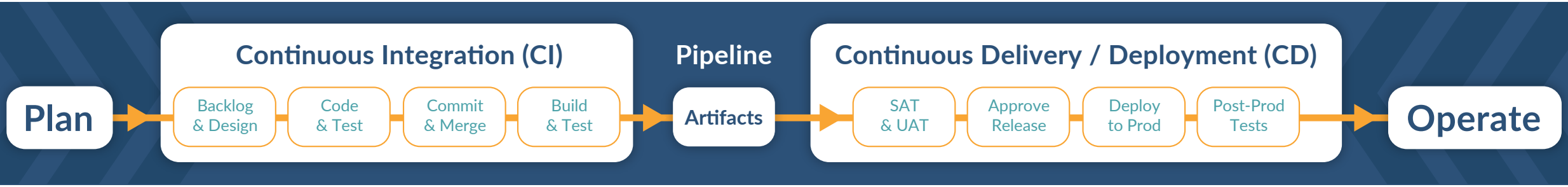
Metrics such as Availability, Latency, and Response Time with Error Budgets

Measurements

Observability, Monitoring, Telemetry, and Instrumentation

Anti-Fragility

Improve Resilience using Fire Drills, Chaos Monkey, Security and Automation



Work Sharing

Work Technical Debt in Small Increments

Manage Load % for Ops, Dev and On-Call Work

Deployments

Gradual Releases using Green/Blue, A/B, Canary Deployments, Automation Scripts, Testing and Monitoring

Performance Management

Monitoring, APM, Capacity Testing & Auto-Scaling

Incident Management

Emergency Response, 50% Ops/Dev Load, 25% On-Call Load, and Blameless Retrospectives

DevOps INSTITUTE

Site Reliability Engineering Practitioner℠

BLUEPRINT

SRE Practitioners deliver business value to customers through collaboration with DevOps teams and engineering of reliable, secure application environments and software systems.

SRE Anti-Patterns

SRE is not a rebranding of Ops.
Alert for user, not system issues.
SLOs are for perceiving user experiences.
False positives are worse than no alerts.
Change by replacement, not updating.
Incidents have plans, not mob reactions.

Full Stack Observability

Provides high-level overviews of system health and granular insights into failure modes of the system, informs context about its inner workings to uncover systemic issues.

Platform SRE and AIOps

Platform SRE solves organizational scalability challenges by applying product management to promoting unified SRE and DevOps culture.
AIOps combines big data and machine learning to automate operations including event correlation, anomaly detection and causality determination.

SLOs For Customer Happiness

Identify system boundaries, define capabilities for each system, define SLI for each capability, define SLO targets, measure baseline.

Benefits

Organization: Stable, reliable services, improved customer experience, culture of collaboration between development and operations.

Individuals: Knowledge and skills for implementing secure and reliable, fault-tolerant systems, observability, intelligent operations and human skills.

Secure and Reliable Systems

Non-abstract large-scale design, intentional architecture, design for changing landscape, design for changing security, multi-grained services architecture, container management, Kubernetes, reactive systems and deployment strategies such as Canary and Blue-Green.

Chaos Engineering

Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system's ability to withstand turbulent conditions.

SRE is the Purest Form of DevOps

SREs code infrastructure and tools; set SLOs, alerts, and report against the SLI such as availability; workload is capped; use tracing and APM tools to understand applications performance and do on-call and postmortems.

Value Stream Management Foundation™

BLUEPRINT

Value stream management (VSM) optimizes value flow and realization to customers resulting in higher organizational performance. VSM accelerates DevOps evolution.

A Value Stream

A value stream is an end-to-end set of activities that collectively creates value for a customer. Anything that delivers a product or a service.

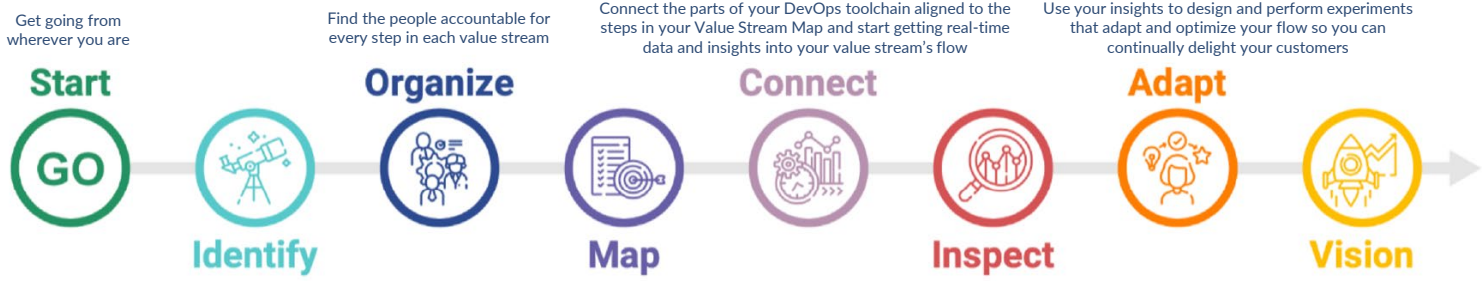
Flow

Flow is the journey of work from idea to realization. Its travel should be friction-free. It's a continuous steady stream of value for customers.

Realization

Realization is the fulfilment of desired outcomes. It's when a customer experiences the value intended. Then the value stream team learn and action insights.

Value Stream Management Implementation Roadmap



Organization

DAO: Distributed Autonomous Organization where teams have value stream oriented roles and own and run their value stream as a business unit.

Principles

Project to product, center around flow, insights driven, continuous compliance, break dependencies, build and measure benefits hypotheses aligned to OKRs.

DevOps Toolchains

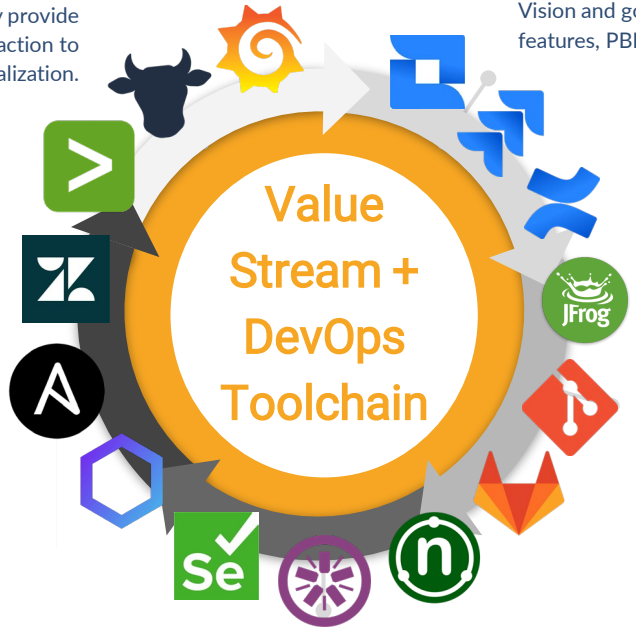
Use a Value Stream Management Platform to (VSMP) to surface actionable insights for continuous inspection and adaptation.

Insights & Analysis

Monitoring and observability provide insights into customer reaction to changes and report on value realization.

Portfolio & Backlog

Vision and goals are set and aligned to epics, features, PBIs and user stories.



Continuous Delivery

The changes are approved, released and operated in the live environment.

Continuous Integration

Code is created, artifacts incorporated, versions controlled, code is built in a trunk based manner.

Continuous Testing

Functional and non-functional testing takes place at every commit at every step or gate through route to live.