

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)

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.

Principles & Practices

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

Related Frameworks

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

DevSecOps Foundation (DSOF)SM

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 automatin, 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.

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.

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.



Site Reliability Engineering (SRE) FoundationSM

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

Continuous Integration (CI)

Backlog
& Design

Code
& Test

Commit
& Merge

Build
& Test

Pipeline

Artifacts

Continuous Delivery / Deployment (CD)

SAT
& UAT

Approve
Release

Deploy
to Prod

Post-Prod
Tests

Operate

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 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.

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

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.



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.

Continuous Testing FoundationSM

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 Frameworks and Tools

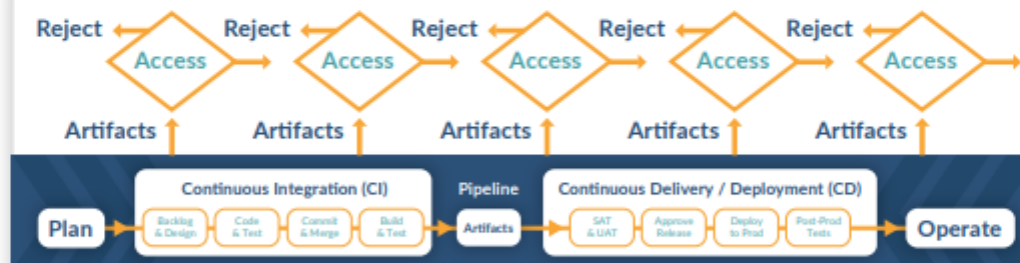
A test framework acts as a "backbone" for integrating and automating tasks including test plans, all types of test tools, test artifact version management, test resources, test data, tests, test schedules, test environment orchestration, test execution, test results, test reports, and test results analysis.

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 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.



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

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 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.

Benefits

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

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.



Continuous Delivery Ecosystem FoundationSM

BLUEPRINT

Continuous Delivery (CD) is a software engineering approach in which teams produce incremental software 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 separate 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 repository. Tests conducted throughout the pipeline catch risky failures before release while completing tests quickly to avoid bottlenecks.

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

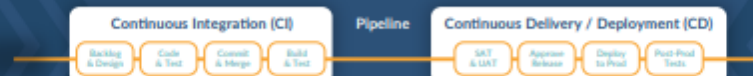
Real-time active monitoring and analytics make tests, processes, and application performance measures visible for real-time decision analytics at each stage of the pipeline to prevent bottlenecks and to identify improvements.

Continuous Delivery & Deployment

Automated configuration management, 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.

Governance, Continuous Security, Site Reliability Engineering (SRE), DR

People, Processes
and Technologies
Management



Infrastructure & Toolchains

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

Important Agile SM Terms

Agile (adjective)

1. Able to move quickly and easily; well-coordinated
2. Able to think and understand quickly; able to solve problems and have new ideas

Agile (software development)

Methods (e.g., Scrum) in which requirements & solutions evolve iteratively & incrementally through collaboration between self-organizing, cross-functional teams

Agile Manifesto

Formal proclamation of four key values and 12 principles to guide an iterative and people-centric approach to development

We value:

Individuals and interactions
Working software
Customer collaboration
Responding to change

Over:

Processes and tools
Comprehensive documentation
Contract negotiations
Following a plan

Agile Service Management

Ensuring ITSM processes reflect Agile values and are designed with "just enough" control and structure to effectively and efficiently deliver services that facilitate customer outcomes when and how they are needed

Agile Process Design

Applying Agile development approach to process design

Agile Process Improvement

Aligning ITSM processes with Agile values as part of Continual Service Improvement (CSI)

Agile Frameworks and Methods

Continuous Integration

Development practice that requires developers to integrate code into a shared repository multiple times a day

Continuous Delivery

Methodology that focuses on making sure software is always in a releasable state throughout its lifecycle

Scrum

Agile framework for effective team collaboration on complex projects

Scrum values include: courage, focus, openness, respect, commitment

Kanban

Method of work that makes work visible, limits work in progress (WIP) and measures velocity (quantity of work done in an iteration)

Lean

Philosophy that focuses on creating more value for customers with fewer resources and less waste

IT Infrastructure Library® (ITIL®)

Set of best practice publications for IT service management

IT Service Management (ITSM)

Set of specialized organizational capabilities for providing value to customers in the form of services

DevOps

Cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals

IT Infrastructure Library® and ITIL® are registered trademarks of AXELOS Limited



Certified Agile Service Manager (CASM)® Reference Card

Important Agile SM Terms

Process

Interrelated work activities that specific inputs and produce specific outputs that are of value to a customer

Process Owner

Role accountable for the overall quality of a process and owner of the Process Backlog

Process Backlog

Prioritized list of everything that needs to be designed or improved for a process including current and future requirements

Minimum Viable Product

The most pared down version of a product (or process) that can still be released

Increment

Potentially shippable completed work that is the outcome of a Sprint

Iteration

A single timeboxed development cycle (i.e., a Sprint)

Timebox

The maximum duration of an event (e.g., a meeting or Sprint)

Velocity

How much product backlog effort a team can handle in a single Sprint

Impediment

Anything that prevents a team member from performing work as efficiently as possible

User Story

A statement written from the user's perspective that describes what a user wants to do with a feature of the software

As a...

I want to...

So I can...

Agile Service Management in a Nutshell

Agile Service Management (like Scrum) is based on time-boxed iterations.



Agile Service Management = 3 Roles, 4 Artifacts and 5 Meetings.

Sprint

A period of 2-4 weeks during which an increment of product work is completed

Definition of Done

Shared understanding of what it means for work to be complete

Burndown Chart

Chart showing the evolution of remaining effort against time

Daily Scrum

A fifteen-minute daily meeting that synchronizes work completed since the prior meeting and forecasts the work to be done before the next one

Sprint Planning Meeting

A 4-8 hour timeboxed event that defines the Sprint Goal, the increment of the Product Backlog that will be done during the Sprint and how it will be done

Sprint Retrospective

A 1.5-3 hour timeboxed event during which the Team reviews the last Sprint and identifies and prioritizes improvements for the next Sprint

Sprint Review

A timeboxed event of 4 hours or less where the Team and stakeholders inspect the work resulting from the Sprint and updates the Product Backlog