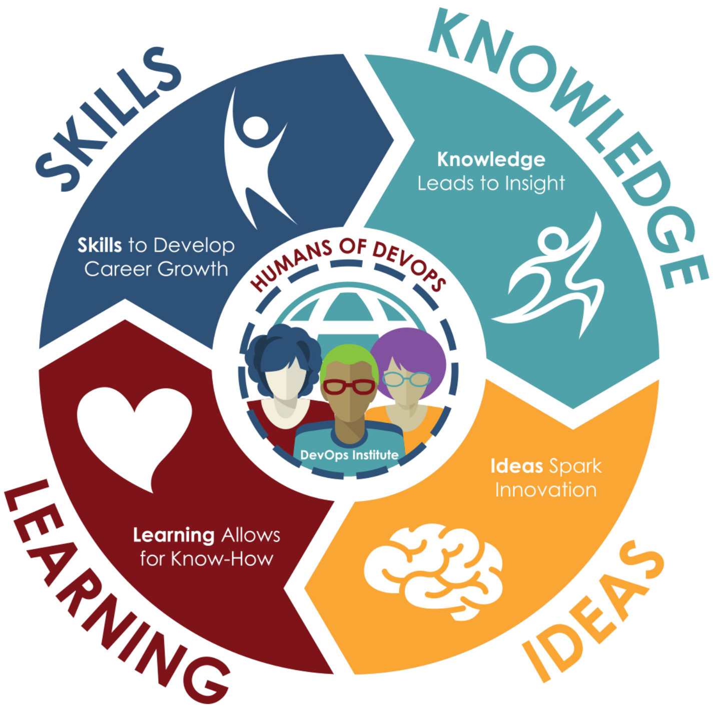
DevOps
INSTITUTE

# VSM Foundation™ Exam Study Guide


SKILLS
KNOWLEDGE
IDEAS
LEARNING
Skills to Develop Career Growth
Knowledge Leads to Insight
Ideas Spark Innovation
Learning Allows for Know-How
HUMANS OF DEVOPS
DevOps Institute

# DevOps Institute's SKIL Framework

DevOps Institute is dedicated to advancing the human elements of DevOps success through our human-centered SKIL framework of Skills, Knowledge, Ideas and Learning.

We develop, accredit and orchestrate SKIL through certifications, research, learning opportunities, events and community connections.

## Join Us!

Become a member and join the fastest growing global community of DevOps practitioners and professionals.

The DevOps Institute continuous learning community is your go-to hub for all things DevOps, so get ready to learn, engage, and inspire.

**devopsinstitute.com/membership**

**You belong.**

DevOps Institute

DevOps Institute is dedicated to advancing the human elements of DevOps success. We fulfill our mission through our SKIL framework of Skills, Knowledge, Ideas and Learning.

Certification is one means of showcasing your skills. While we strongly support formal training as the best learning experience and method for certification preparation, DevOps Institute also recognizes that humans learn in different ways from different resources and experiences. As the defacto certification body for DevOps, DevOps Institute has now removed the barrier to certification by removing formal training prerequisites and opening our testing program to anyone who believes that they have the topical knowledge and experience to pass one or more of our certification exams.

This examination study guide will help test-takers prepare by defining the scope of the exam and includes the following:

- Course Description
- Examination Requirements
- DevOps Glossary of Terms
- Value Added Resources
- Sample Exam(s) with Answer Key

These assets provide a guideline for the topics, concepts, vocabulary and definitions that the exam candidate is expected to know and understand in order to pass the exam. The knowledge itself will need to be gained on its own or through training by one of our Global Education Partners.

Test-takers who successfully pass the exam will also receive a certificate and digital badge from DevOps Institute, acknowledging their achievement, that can be shared with their professional online networks.

If you have any questions, please contact our DevOps Institute Customer Service team at CustomerService@DevOpsInstitute.com.

# Value Stream Management Foundation

# Course Description

**DURATION: 16 hours**

**An introduction to value stream management principles, practices, and tools. Designed to optimize value flow and realization in digital value streams and develop individuals' skills. Digital value streams include the software applications and services in addition to the platforms that support them e.g., cloud infrastructure or DevOps toolchains.**

**OVERVIEW**

The Value Stream Management Foundation course from Value Stream Management Consortium, and offered in partnership with DevOps Institute, is an introductory course taking learners through a value stream management implementation journey. It considers the human, process, and technology aspects of this way of working and explores how optimizing value streams for flow and realization positively impacts organizational performance.

**LEARNING OBJECTIVES**

- Describe the origins of value stream management and key concepts such as flow, value, and delivery
- Describe what value stream management is, why it's needed and the business benefits of its practice
- Describe how lean, agile, DevOps, and ITSM principles contribute to value stream management
- Identify and describe value streams, where they start and end, and how they interconnect
- Identify value stream roles and responsibilities
- Express value streams visually using mapping techniques, define current and target states and hypothesis backlog
- Write value stream flow and realization optimization hypotheses and experiments
- Apply metrics such as touch/processing time, wait/idle time, and cycle time to value streams
- Understand flow metrics and how to access the data to support data-driven conversations and decisions
- Examine value realization metrics and aligning to business outcomes, and how to sense and respond to them (outcomes versus outputs)

- Architect a DevOps toolchain alongside a value stream and data connection points
- Design a continuous inspection and adaptation approach for organizational evolution

## LEARNER MATERIALS

- Sixteen (16) hours of live instructor-led training via the DevOps Institute Education Partner network
- Digital Learner Manual including:
    - Course slideware
    - Value-Added Resources
    - Interactive exercises and discussions designed to apply concepts
    - Case stories
    - Glossary
    - Sample exam(s)
- Access to additional sources of information
- Option to join the DevOps Institute and Value Stream Management Consortium membership communities

## AUDIENCE

People working in an organization aiming to improve performance, especially in response to digital transformation or disruption. Any roles involved in the creation and delivery of products or services:

- Leadership and CXO, especially CIO, CTO, CPO, and CVO
- Transformation and evolution leads and change agents
- Value stream architects, managers, engineers
- Scrum Masters, agile and DevOps coaches and facilitators
- Portfolio, product and project managers, and owners
- Business analysts
- Architects, developers, and engineers
- Release and environment managers
- IT Ops, service and support desk workers
- Customer experience and success professionals

## PREREQUISITES

None, although having a working knowledge of lean, agile, DevOps, and value stream principles will be useful.

## CERTIFICATION EXAM

Successfully passing (65%) the 60-minute examination, consisting of 40 multiple-choice questions, leads to the candidate receiving their Value Stream Management Foundation certification and digital badge. The certification is governed and maintained in partnership with DevOps Institute.

vsm consortium

DevOps Institute

# Value Stream Management Foundation (VSMF)™ v1.0

Examination Requirements

# Value Stream Management Foundation (VSMF)TM Certification

Value Stream Management Foundation is a certification that is accredited by DevOps Institute in partnership with the Value Stream Management Consortium. The purpose of this certification and its associated course is to impart, test, and validate knowledge of Value Stream Management vocabulary, principles, practices, automation, and purpose.

## Eligibility for Examination

Although there are no formal prerequisites for the exam, DevOps Institute highly recommends the following to prepare candidates for the exam leading to the Value Stream Management Foundation certification:

- It is recommended that candidates complete at least 16 contact hours (instruction, labs, and self-study) as part of a formal, approved training course delivered by an accredited Education Partner of DevOps Institute.

## Examination Administration

The Value Stream Management Foundation examination is accredited, managed, and administered under the strict protocols and standards of DevOps Institute.

## Level of Difficulty

The Value Stream Management Foundation certification uses the Bloom Taxonomy of Educational Objectives in the construction of both the learning content and the examination.

- The Value Stream Management Foundation exam contains Bloom 1 questions that test learners' ***knowledge*** of Value Stream Management and DevOps concepts and vocabulary terms.
- The exam also contains Bloom 2 questions that test learners' ***comprehension*** of these concepts in context.

## Format of the Examination

Candidates must achieve a passing score to gain the Value Stream Management Foundation Certification.

| | |
|---|---|
| Exam Type | 40 multiple choice questions |
| Duration | 60 minutes |
| Prerequisites | It is recommended that candidates complete the Value Stream Management Foundation course from an accredited DevOps Institute Education Partner. |
| Supervised | No |
| Open Book | Yes |
| Passing Score | 65% |
| Delivery | Web-based |
| Badge | Value Stream Management Foundation |

## Exam Topic Areas and Question Weighting

The Value Stream Management Foundation exam requires knowledge of the topic areas specified below:

| Topic Area | Description | Max Questions |
|---|---|---|
| VSMF – 1 History and Evolution | The history and evolution of value stream management and its application to digital and technology-led value streams | 4 |
| VSMF - 2 Identifying Value Streams | Setting a long-term vision and identifying value streams. The use of value stream-centric thinking in DevOps and other ways of working frameworks | 4 |
| VSMF - 3 Mapping Value Streams | Mapping value streams as a tool and discipline. When and how to map value streams. Roles in value stream mapping: sponsors, facilitators, teams. When to map | 6 |

| | | |
|---|---|---|
| VSMF - 4 Connecting DevOps Toolchains | Connecting DevOps toolchains to value streams. Traceability, visibility, and value stream management platforms. | 5 |
| VSMF - 5 Value Stream Metrics | Value stream metrics for flow and realization, definitions of value and benefits hypotheses. | 5 |
| VSMF - 6 Inspecting the Value Stream | Inspecting value stream for both flow and realization, making it visible, when to inspect what data, how to use tools to discover insights and trends. | 5 |
| VSMF - 7 Organizing as Value Streams | Organizational design and value stream alignment, transitioning to VSM, who does what, and finance considerations | 6 |
| VSMF - 8 Evolving Value Streams | Measuring VSM capability and evolution, deciding and driving improvements, building hypotheses and observing outcomes, using KPIs and OKRs to adopt VSM | 5 |

### Concept and Terminology List

The candidate is expected to understand and comprehend the following Value Stream Management concepts and vocabulary at Blooms Level 1 and 2.

- Agile
- Agile at Scale
- Chief Value Officer
- Continuous Delivery
- Continuous Delivery Pipeline
- Continuous Flow
- Continuous Integration
- Customer Journey Mapping
- Cycle Time
- DevOps
- DevOps Metrics
- DevOps Pipeline
- DevOps Toolchains
- Digital Product Management
- Digital Transformation
- Digital Value Streams
- Flow
- Flow Metrics
- Fuzzy Front End
- Gemba
- Hypothesis Backlog
- Hypothesis-Driven Development
- Ideation
- Impact-Driven Development
- Insights-Driven Business
- Key Performance Indicators
- Lead Time
- Lean
- Objectives and Key Results
- Observability
- Portfolio and Backlog
- Process Mapping
- Project to Product
- Systems Thinking
- Team Topologies
- The Theory of Constraints
- The Three Pillars of Empiricism
- The Three Ways of DevOps
- Touch Time
- Value
- Value-Adding Activities
- Value Cycle
- Value Realization
- Value Stream Architect
- Value Stream Management
- Value Stream Management Platform
- Value Stream Mapping
- Value Stream Metrics
- Velocity
- Wait Time
- Waste
- Waterfall

DevOps Institute

# DEVOPS GLOSSARY OF TERMS

This glossary is provided for reference only as it contains key terms that may or may not be examinable.

# DevOps Glossary of Terms

| Term | Definition |
|---|---|
| 12-Factor App Design | A methodology for building modern, scalable, maintainable software-as-a-service applications. |
| 2-Factor *or* 2-Step Authentication | Two-Factor Authentication, also known as 2FA or TFA or Two-Step Authentication is when a user provides two authentication factors; usually, firstly a password and then a second layer of verification such as a code texted to their device, shared secret, physical token, or biometrics. |
| A/B Testing | Deploy different versions of an EUT to different customers and let the customer feedback determine which is best. |
| A3 Problem Solving | A structured problem-solving approach that uses a lean tool called the A3 Problem-Solving Report. The term "A3" represents the paper size historically used for the report (a size roughly equivalent to 11" x 17"). |
| Access Management | Granting an authenticated identity access to an authorized resource (e.g., data, service, environment) based on defined criteria (e.g., a mapped role), while preventing unauthorized identity access to a resource. |
| Access Provisioning | Access provisioning is the process of coordinating the creation of user accounts, e-mail authorizations in the form of rules and roles, and other tasks such as provisioning of physical resources associated with enabling new users to systems or environments. |
| Administration Testing | The purpose of the test is to determine if an End User Test (EUT) is able to process administration tasks as expected. |
| Advice Process | Any person making a decision must seek advice from everyone meaningfully affected by the decision and people with expertise in the matter. Advice received must be taken into consideration, though it does not have to be accepted or followed. The objective of the advice process is not to form a consensus, but to inform the decision-maker so that they can make the best decision possible. Failure to follow the advice process undermines trust and unnecessarily introduces risk to the business. |
| Agile | A work management method for complex endeavors that divides tasks into small "sprints" of work with frequent reassessment and adaptation of plans. |

| | |
|---|---|
| Agile (adjective) | Able to move quickly and easily; well-coordinated. Able to think and understand quickly; able to solve problems and have new ideas. |
| Agile Coach | Help teams master Agile development and DevOps practices; enables productive ways of working and collaboration. |
| Agile Enterprise | A fast-moving, flexible, and robust company capable of rapid response to unexpected challenges, events, and opportunities. |
| Agile Manifesto | A formal proclamation of values and principles to guide an iterative and people-centric approach to software development. http://agilemanifesto.org |
| Agile Portfolio Management | Involves evaluating in-flight projects and proposed future initiatives to shape and govern the ongoing investment in projects and discretionary work. CA's Agile Central and VersionOne are examples. |
| Agile Practice Owner | A role accountable for the overall quality of a service management practice and owner of the Practice Backlog. |
| Agile Principles | The twelve principles that underpin the Agile Manifesto. |
| Agile Process | Delivers "just enough" structure and control to enable the organization to achieve its service outcomes in the most expeditious, effective, and efficient way possible. It is easy to understand, easy to follow, and prizes its collaboration and outcomes more than its artifacts. |
| Agile Process Engineering | An iterative and incremental approach to designing a process with short, iterative designs of potentially shippable process increments or microprocesses. |
| Agile Process Improvement | Ensures that IT Service Management agility introduced through Agile Process Engineering is continually reviewed and adjusted as part of IT Service Management's commitment to continual improvement. |
| Agile Service Management | A framework that ensures that ITSM processes reflect Agile values and are designed with "just enough" control and structure in order to effectively and efficiently deliver services that facilitate customer outcomes when and how they are needed. |
| Agile Service Management Artifacts | Practice Backlog, Sprint Backlog, Increment |
| Agile Service Management Events | Practice/microprocess Planning, The Sprint, Sprint Planning, Process Standup, Sprint Review, Sprint Retrospective |

# DevOps Glossary of Terms

| | |
|---|---|
| Agile Service Management Roles | Agile Practice Owner, Agile Service Management Team, Agile Service Manager |
| Agile Service Management Team | A team of at least 3 people (including a customer or practitioner) that is accountable for a single microprocess or a complete service management practice. |
| Agile Service Manager | An Agile Service Management subject matter expert who is the coach and protector of the Agile Service Management Team. |
| Agile Software Development | Group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. Usually applied using the Scrum or Scaled Agile Framework approach. |
| Amazon Web Services (AWS) | Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery, and other functionality to help businesses scale and grow. |
| Analytics | Test results processed and presented in an organized manner in accordance with analysis methods and criteria. |
| Andon | A system gives an assembly line worker the ability, and moreover the empowerment, to stop production when a defect is found, and immediately call for assistance. |
| Anti-pattern | A commonly reinvented but poor solution to a problem. |
| Anti-fragility | Antifragility is a property of systems that increases its capability to thrive as a result of stressors, shocks, volatility, noise, mistakes, faults, attacks, or failures. |
| API Testing | The purpose of the test is to determine if an API for an EUT functions as expected. |
| Application Performance Management (APM) | APM is the monitoring and management of the performance and availability of software applications. APM strives to detect and diagnose complex application performance problems to maintain an expected level of service. |
| Application Programming Interface (API) | A set of protocols used to create applications for a specific OS or as an interface between modules or applications. |

| | |
|---|---|
| Application Programming Interface (API) Testing | The purpose of the test is to determine if an API for an EUT functions as expected. |
| Application Release | Controlled continuous delivery pipeline capabilities including automation (release upon code commit). |
| Application Release Automation (ARA) or Orchestration (ARO) | Controlled continuous delivery pipeline capabilities including automation (release upon code commit), environment modeling (end-to-end pipeline stages, and deploy application binaries, packages, or other artifacts to target environments), and release coordination (project, calendar, and scheduling management, integrate with change control and/or IT service support management). |
| Application Test-Driven Development (ATDD) | Acceptance Test-Driven Development (ATDD) is a practice in which the whole team collaboratively discusses acceptance criteria, with examples, and then distills them into a set of concrete acceptance tests before development begins. |
| Application Testing | The purpose of the test is to determine if an application is performing according to its requirements and expected behaviors. |
| Application Under Test (AUT) | The EUT is a software application. E.g. Business application is being tested. |
| Architecture | The fundamental underlying design of computer hardware, software, or both in combination. |
| Artifact | Any element in a software development project including documentation, test plans, images, data files, and executable modules. |
| Artifact Repository | Store for binaries, reports, and metadata. Example tools include JFrog Artifactory, Sonatype Nexus. |
| Attack path | The chain of weaknesses a threat may exploit to achieve the attacker's objective. For example, an attack path may start by compromising a user's credentials, which are then used in a vulnerable system to escalate privileges, which in turn is used to access a protected database of information, which is copied out to an attacker's own server(s). |
| Audit Management | The use of automated tools to ensure products and services are auditable, including keeping audit logs of build, test and deploy activities, auditing configurations, and users, as well as log files from production operations. |

| | |
|---|---|
| Authentication | The process of verifying an asserted identity. Authentication can be based on what you know (e.g., password or PIN), what you have (token or one-time code), what you are (biometrics), or contextual information. |
| Authorization | The process of granting roles to users to have access to resources. |
| Auto-DevOps | Auto DevOps brings DevOps best practices to your project by automatically configuring software development lifecycles. It automatically detects, builds, tests, deploys, and monitors applications. |
| Auto-scaling | The ability to automatically and elastically scale and de-scale infrastructure depending on traffic and capacity variations while maintaining control of costs. |
| Automated rollback | If a failure is detected during a deployment, an operator (or an automated process) will verify the failure and roll back the failing release to the previous known working state. |
| Availability | Availability is the proportion of time a system is in a functioning condition and therefore available (to users) to be used. |
| Backdoor | A backdoor bypasses the usual authentication used to access a system. Its purpose is to grant the cybercriminals future access to the system even if the organization has remediated the vulnerability initially used to attack the system. |
| Backlog | Requirements for a system expressed as a prioritized list of product backlog items usually in the form of 'User Stories'. The product backlog is prioritized by the Product Owner and should include functional, non-functional, and technical team-generated requirements. |
| Basic Security Hygiene | A common set of minimum-security practices that must be applied to all environments without exception. Practices include basic network security (firewalls and monitoring), hardening, vulnerability and patch management, logging and monitoring, basic policies and enforcement (may be implemented under a "policies as code" approach), and identity and access management. |
| Batch Sizes | Refers to the volume of features involved in a single code release. |
| Bateson Stakeholder Map | A tool for mapping stakeholder's engagement with the initiative in progress. |
| Behavior Driven Development (BDD) | Test cases are created by simulating an EUT's externally observable inputs, and outputs. Example tool: Cucumber. |

| | |
|---|---|
| Beyond Budgeting | A management model that looks beyond command-and-control towards a more empowered and adaptive state. |
| Black-Box | Test case only uses knowledge of externally observable behaviors of an EUT. |
| Blameless post mortems | A process through which engineers whose actions have contributed to a service incident can give a detailed account of what they did without fear of punishment or retribution. |
| Blast Radius | Used for impact analysis of service incidents. When a particular IT service fails, the users, customers, other dependent services that are affected. |
| Blue/Green Testing or Deployments | Taking software from the final stage of testing to live production using two environments labeled Blue and Green. Once the software is working in the green environment, switch the router so that all incoming requests go to the green environment - the blue one is now idle. |
| Bug | An error or defect in software that results in an unexpected or system-degrading condition. |
| Bureaucratic Culture | Bureaucratic organizations are likely to use standard channels or procedures which may be insufficient in a crisis (Westrum). |
| Bursting | Public cloud resources are added as needed to temporarily increase the total computing capacity of a private cloud. |
| Business Case | Justification for a proposed project or undertaking on the basis of its expected commercial benefit. |
| Business Continuity | Business continuity is an organization's ability to ensure operations and core business functions are not severely impacted by a disaster or unplanned incident that takes critical services offline. |
| Business Transformation | Changing how the business functions. Making this a reality means changing culture, processes, and technologies in order to better align everyone around delivering on the organization's mission. |
| Business Value | In management, an informal term that includes all forms of value that determine the health and well-being of the firm in the long run. |
| Cadence | Flow or rhythm of events. |

| | |
|---|---|
| CALMS Model | Considered the pillars or values of DevOps: Culture, Automation, Lean, Measurement, Sharing (as put forth by John Willis, Damon Edwards, and Jez Humble). |
| Canary Testing | A canary (also called a canary test) is a push of code changes to a small number of end-users who have not volunteered to test anything. Similar to incremental rollout, it is where a small portion of the user base is updated to a new version first. This subset, the canaries, then serve as the proverbial "canary in the coal mine". If something goes wrong then a release is rolled back and only a small subset of the users are impacted. |
| Capacity | An estimate of the total amount of engineering time available for a given Sprint. |
| Capacity Test | The purpose of the test is to determine if the EUT can handle expected loads such as number of users, number of sessions, aggregate bandwidth. |
| Capture-Replay | Test cases are created by capturing live interactions with the EUT, in a format that can be replayed by a tool. E.g. Selenium |
| Carrots | Positive incentives, for encouraging and rewarding desired behaviors. |
| Chain of Goals | A method designed by Roman Pichler of ensuring that goals are linked and shared at all levels through the product development process. |
| Change | Addition, modification, or removal of anything that could have an effect on IT services. (ITIL® definition) |
| Change Failure Rate | A measure of the percentage of failed/rolled back changes. |
| Change Fatigue | A general sense of apathy or passive resignation towards organizational changes by individuals or teams. |
| Change Lead Time | A measure of the time from a request for a change to the delivery of the change. |
| Change Leader Development Model | Jim Canterucci's model for five levels of change leader capability. |
| Change Management | The process that controls all changes throughout their lifecycle. (ITIL definition) |

| | |
|---|---|
| Change Management (Organizational) | An approach to shifting or transitioning individuals, teams & organizations from a current state to a desired future state. Includes the process, tools & techniques to manage the people-side of change to achieve the required business outcome(s). |
| Change-based Test Selection Method | Tests are selected according to a criterion that matches attributes of tests to attributes of the code that is changed in a build. |
| Chaos Engineering | The discipline of experimenting on a software system in production in order to build confidence in the system's capability to withstand turbulent and unexpected conditions. |
| Chapter Lead | A squad line manager in the Spotify model who is responsible for traditional people management duties is involved in day-to-day work, and grows individual and chapter competence. |
| Chapters | A small family of people having similar skills and who work within the same general competency area within the same tribe. Chapters meet regularly to discuss challenges and areas of expertise in order to promote sharing, skill development, re-use, and problem-solving. |
| ChatOps | An approach to managing technical and business operations (coined by GitHub) that involves a combination of group chat and integration with DevOps tools. Example tools include Atlassian HipChat/Stride, Microsoft Teams, Slack. |
| Check-in | The action of submitting a software change into a system version management system. |
| CI Regression Test | A subset of regression tests that are run immediately after a software component is built. Same as Smoke Test. |
| Clear-Box | Same as Glass-Box Testing and White-Box Testing. |
| Cloud Computing | The practice of using remote servers hosted on the internet to host applications rather than local servers in a private data center. |
| Cloud-Native | Native cloud applications (NCA) are designed for cloud computing. |

| | |
|---|---|
| Cloudbees | Cloudbees is a commercially supported proprietary automation framework tool that works with and enhances Jenkins by providing enterprise levels support and add-on functionality. |
| Cluster Cost Optimization | Tools like Kubecost, Replex, Cloudability use monitoring to analyze container clusters and optimize the resource deployment model. |
| Cluster Monitoring | Tools that let you know the health of your deployment environments running in clusters such as Kubernetes. |
| Clustering | A group of computers (called nodes or members) work together as a cluster connected through a fast network acting as a single system. |
| Code Coverage | A measure of white box test coverage by counting code units that are executed by a test. The code unit may be a code statement, a code branch, or control path or data path through a code module. |
| Code Quality | See also static code analysis, Sonar and Checkmarks are examples of tools that automatically check the seven main dimensions of code quality – comments, architecture, duplication, unit test coverage, complexity, potential defects, language rules. |
| Code Repository | A repository where developers can commit and collaborate on their code. It also tracks historical versions and potentially identifies conflicting versions of the same code. Also referred to as "repository" or "repo." |
| Code Review | Software engineers inspect each other's source code to detect coding or code formatting errors. |
| Cognitive Bias | Cognitive bias is a limitation in objective thinking that is caused by the tendency for the human brain to perceive information through a filter of personal experience and preferences: a systematic pattern of deviation from norm or rationality in judgment. |
| Collaboration | People jointly working with others towards a common goal. |
| Collaborative Culture | A culture that applies to everyone which incorporates an expected set of behaviors, language, and accepted ways of working with each other reinforcement by leadership. |
| Compatibility Test | Test with the purpose to determine if an EUT interoperates with another EUT such as peer-to-peer applications or protocols. |

| | |
|---|---|
| Configuration Management | Configuration management (CM) is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life. |
| Conformance Test | The purpose of the test is to determine if an EUT complies with a standard. |
| Constraint | Limitation or restriction; something that constrains. See also *bottleneck*. |
| Container | A way of packaging software into lightweight, stand-alone, executable packages including everything needed to run it (code, runtime, system tools, system libraries, settings) for development, shipment, and deployment. |
| Container Network Security | Used to prove that any app that can be run on a container cluster with any other app can be confident that there is no unintended use of the other app or any unintended network traffic between them. |
| Container Registry | Secure and private registry for Container images. Typically allowing for easy upload and download of images from the build tools. Docker Hub, Artifactory, Nexus are examples. |
| Container Scanning | When building a Container image for your application, tools can run a security scan to ensure it does not have any known vulnerability in the environment where your code is shipped. Blackduck, Synopsis, Synk, Claire, and Klar are examples. |
| Continual Service Improvement (CSI) | One of the ITIL Core publications and a stage of the service lifecycle. |
| Continuous Delivery (CD) | A methodology that focuses on making sure software is always in a releasable state throughout its lifecycle. |
| Continuous Delivery (CD) Architect | A person who is responsible to guide the implementation and best practices for a continuous delivery pipeline. |
| Continuous Delivery Pipeline | A continuous delivery pipeline refers to the series of processes that are performed on product changes in stages. A change is injected at the beginning of the pipeline. A change may be new versions of code, data, or images for applications. Each stage processes the artifacts resulting from the prior stage. The last stage results in deployment to production. |

| | |
|---|---|
| Continuous Delivery Pipeline Stage | Each process in a continuous delivery pipeline. These are not standard. Examples are Design: determine implementation changes; Creation: implement an unintegrated version of design changes; Integration: merge |
| Continuous Deployment | A set of practices that enable every change that passes automated tests to be automatically deployed to production. |
| Continuous Flow | Smoothly moving people or products from the first step of a process to the last with minimal (or no) buffers between steps. |
| Continuous Improvement | Based on Deming's Plan-Do-Check-Act, a model for ensuring ongoing efforts to improve products, processes, and services. |
| Continuous Integration (CI) | A development practice that requires developers to merge their code into trunk or master ideally at least daily and perform tests (i.e. unit, integration, and acceptance) at every code commit. |
| Continuous Integration Tools | Tools that provide an immediate feedback loop by regularly merging, building, and testing code. Example tools include Atlassian Bamboo, Jenkins, Microsoft VSTS/Azure DevOps, TeamCity. |
| Continuous Monitoring (CM) | This is a class of terms relevant to logging, notifications, alerts, displays, and analysis of test results information. |
| Continuous Testing (CT) | This is a class of terms relevant to the testing and verification of an EUT in a DevOps environment. |
| Conversation Café | Conversation Cafés are open, hosted conversations in cafés as well as conferences and classrooms—anywhere people gather to make sense of our world. |
| Conway's Law | Organizations that design systems are constrained to produce designs that are copies of the communication structures of these organizations. |
| Cooperation vs. Competition | The key cultural value shift toward being highly collaborative and cooperative, and away from internal competitiveness and divisiveness. |

| | |
|---|---|
| COTS | Commercial-off-the-shelf solution |
| Critical Success Factor (CSF) | Something that must happen for an IT service, process, plan, project or other activity to succeed. |
| Cultural Iceberg | A metaphor that visualizes the difference between observable (above the water) and non-observable (below the waterline) elements of culture. |
| Culture (Organizational Culture) | The values and behaviors that contribute to the unique psychosocial environment of an organization. |
| Cumulative Flow Diagram | A cumulative flow diagram is a tool used in agile software development and lean product development. It is an area graph that depicts the quantity of work in a given state, showing arrivals, time in queue, quantity in a queue, and departure. |
| Current State Map | A form of value stream map that helps you identify how the current process works and where the disconnects are. |
| Customer Reliability Engineer (CRE) | CRE is what you get when you take the principles and lessons of SRE and apply them to customers. |
| Cycle Time | A measure of the time from the start of work to ready for delivery. |
| Daily Scrum | Daily timeboxed event of 15 minutes or less for the Team to replan the next day of work during a Sprint. |
| Dashboard | Graphical display of summarized data e.g., deployment frequency, velocity, test results. |
| DAST (Dynamic Application Security Testing) | Dynamic application security testing (DAST) is a process of testing an application or software product in an operating state. |
| Data Loss Protection (DLP) | Tools that prevent files and content from being removed from within a service environment or organization. |
| Database Reliability Engineer (DBRE) | A person responsible for keeping database systems that support all user-facing services in production running smoothly. |

| | |
|---|---|
| Defect Density | The number of faults found in a unit E.g. # defects per KLOC, # defects per change. |
| Definition of Done | A shared understanding of expectations that an Increment or backlog item must live up to. |
| Delivery Cadence | The frequency of deliveries. E.g. # deliveries per day, per week, etc. |
| Delivery Package | Set of release items (files, images, etc.) that are packaged for deployment. |
| Deming Cycle | A four-stage cycle for process management, attributed to W. Edwards Deming. Also called Plan-Do-Check-Act (PDCA). |
| Dependency Firewall | Many projects depend on packages that may come from unknown or unverified providers, introducing potential security vulnerabilities. There are tools to scan dependencies but that is after they are downloaded. These tools prevent those vulnerabilities from being downloaded to begin with. |
| Dependency Proxy | For many organizations, it is desirable to have a local proxy for frequently used upstream images/packages. In the case of CI/CD, the proxy is responsible for receiving a request and returning the upstream image from a registry, acting as a pull-through cache. |
| Dependency Scanning | Used to automatically find security vulnerabilities in your dependencies while you are developing and testing your applications. Synopsys, Gemnasium, Retire.js, and bundler-audit are popular tools in this area. |
| Deployment | The installation of a specified version of software to a given environment (e.g., promoting a new build into production). |
| Design for Testability | An EUT is designed with features that enable it to be tested. |
| Design Principles | Principles for designing, organizing, and managing a DevOps delivery operating model. |
| Dev | Individuals involved in software development activities such as application and software engineers. |

| | |
|---|---|
| Developer (Dev) | An individual who has the responsibility to develop changes for an EUT. Alternate: Individuals involved in software development activities such as application and software engineers. |
| Development Test | Ensuring that the developer's test environment is a good representation of the production test environment. |
| Device Under Test (DUT) | The DUT is a device (e.g. router or switch) being tested. |
| DevOps | A cultural and professional movement that stresses communication, collaboration, and integration between software developers and IT operations professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably." (Wikipedia) |
| DevOps Coach | Help teams master Agile development and DevOps practices; enables productive ways of working and collaboration. |
| DevOps Infrastructure | The entire set of tools and facilities that make up the DevOps system. Includes CI, CT, CM, and CD tools. |
| DevOps Kaizen | Kaizen is a Japanese word that closely translates to "change for better," the idea of continuous improvement—large or small—involving all employees and crossing organizational boundaries. Damon Edwards' DevOps Kaizen shows how making small, incremental improvements (little J's) has an improved impact on productivity long term. |
| DevOps Pipeline | The entire set of interconnected processes that make up a DevOps Infrastructure. |
| DevOps Score | A metric showing DevOps adoption across an organization and the corresponding impact on delivery velocity. |
| DevOps Toolchain | The tools needed to support a DevOps continuous development and delivery cycle from idea to value realization. |

| | |
|---|---|
| DevSecOps | A mindset that "everyone is responsible for security" with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required. |
| Digital Transformation | The adoption of digital technology by a company to improve business processes, value for customers, and innovation. |
| Digital Value Stream | A value stream is anything that delivers a product or a service. A digital value stream is one that delivers a digital product or service. |
| Distributed Version Control System (DVCS) | The software revisions are stored in a distributed revision control system (DRCS), also known as a distributed version control system (DVCS). |
| DMZ (De-Militarized Zone) | A DMZ in network security parlance is a network zone in between the public internet and internal protected resources. Any application, server, or service (including APIs) that need to be exposed externally are typically placed in a DMZ. It is not uncommon to have multiple DMZs in parallel. |
| Dynamic Analysis | Dynamic analysis is the testing of an application by executing data in real-time with the objective of detecting defects while it is in operation, rather than by repeatedly examining the code offline. |
| Dynamic Application Security Testing (DAST) | Dynamic application security testing (DAST) is a process of testing an application or software product in an operating state. |
| EggPlant | Automated function and regression testing of enterprise applications. Licensed by Test Plant. |
| Elastic Infrastructure | Elasticity is a term typically used in cloud computing, to describe the ability of an IT infrastructure to quickly expand or cut back capacity and services without hindering or jeopardizing the infrastructure's stability, performance, security, governance, or compliance protocols. |
| eNPS | Employee Net Promoter Score (eNPS) is a way for organizations to measure employee loyalty. The Net Promoter Score, originally a customer service tool, was later used internally on employees instead of customers. |
| Entity Under Test (EUT) | This is a class of terms that refers to the names of types of entities that are being tested. These terms are often abbreviated to the form xUT where "x" represents a type of entity under test. |
| Epic | A collection of related user stories that may need to be worked on across multiple Sprints. |

| | |
|---|---|
| Erickson (Stages of Psychosocial Development) | Erik Erikson (1950, 1963) proposed a psychoanalytic theory of psychosocial development comprising eight stages from infancy to adulthood. During each stage, the person experiences a psychosocial crisis which could have a positive or negative outcome for personality development. |
| Error Budget | The error budget provides a clear, objective metric that determines how unreliable a service is allowed to be within a specific time period. |
| Error Budget Policies | An error budget policy enumerates the activity a team takes when they've exhausted their error budget for a particular service in a particular time period. |
| Error Tracking | Tools to easily discover and show the errors that the application may be generating, along with the associated data. |
| External Automation | Scripts and automation outside of a service that is intended to reduce toil. |
| Fail Early | A DevOps tenet referring to the preference to find critical problems as early as possible in a development and delivery pipeline. |
| Fail Often | A DevOps tenet which emphasizes a preference to find critical problems as fast as possible and therefore frequently. |
| Failure Rate | Fail verdicts per unit of time. |
| False Negative | A test incorrectly reports a verdict of "fail" when the EUT actually passed the purpose of the test. |
| False Positive | A test incorrectly reports a verdict of "pass" when the EUT actually failed the purpose of the test. |
| Feature Toggle | The practice of using software switches to hide or activate features. This enables continuous integration and testing a feature with selected stakeholders. |
| Federated Identity | A central identity used for access to a wide range of applications, systems, and services, but with a particular skew toward web-based applications. Also, often referenced as Identity-as-a-Service (IDaas). Any identity that can be reused across multiple sites, particularly via SAML or OAuth authentication mechanisms. |

| | |
|---|---|
| Fire Drills | A planned failure testing process focussed on the operation of live services including service failure testing as well as communication, documentation, and other human factor testing. |
| Flow | How people, products, or information move through a process. Flow is the first way of The Three Ways. |
| Flow of Value | A form of map that shows the end-to-end value stream. This view is usually not available within the enterprise. |
| Framework | The backbone for plugging in tools. Launches automated tasks, collects results from automated tasks. |
| Freedom and Responsibility | A core cultural value that with the freedom of self-management (such as afforded by DevOps) comes the responsibility to be diligent, to follow the advice process, and to take ownership of both successes and failures. |
| Frequency | How often an application is released. |
| Functional Testing | Tests to determine if the functional operation of the service is as expected. |
| Future State Map | A form of value stream map that helps you develop and communicate what the target end state should look like and how to tackle the necessary changes. |
| Fuzzing | Fuzzing or fuzz testing is an automated software testing practice that inputs invalid, unexpected, or random data into applications. |
| Gated Commits | Define and obtain consensus for the criterion of changes promoted between all CD pipeline stages such as Dev to CI stage / CI to packaging/delivery stage / Delivery to Deployment/Production stage. |
| Generative (DevOps) Culture | In a generative organization, alignment takes place through identification with the mission. The individual "buys into" what he or she is supposed to do and its effect on the outcome. Generative organizations tend to be proactive in getting the information to the right people by any means. necessary. (Westrum) |
| Generativity | A cultural view wherein long-term outcomes are of primary focus, which in turn drives investments and cooperation that enable an organization to achieve those outcomes. |
| Glass-Box | Same as Clear-Box Testing and White-Box Testing. |

| | |
|---|---|
| Goal-seeking tests | The purpose of the test is to determine an EUT's performance boundaries, using incrementally stresses until the EUT reaches peak performance. E.g. Determine the maximum throughput that can be handled without errors. |
| Golden Circle | A model by Simon Sinek that emphasizes an understanding of the business' "why" before focusing on the "what" and "how". |
| Golden Image | A template for a virtual machine (VM), virtual desktop, server, or hard disk drive. (TechTarget) |
| Goleman's Six Styles of Leadership | Daniel Goleman (2002) created the Six Leadership Styles and found, in his research, that leaders used one of these styles at any one time. |
| Governance, Risk Management and Compliance (GRC) | A team or software platform intended for concentrating governance, compliance, and risk management data, including policies, compliance requirements, vulnerability data, and sometimes asset inventory, business continuity plans, etc. In essence, a specialized document and data repository for security governance. Or a team of people who specialize in IT/security governance, risk management, and compliance activities. Most often non-technical business analyst resources. |
| Gray-Box | Test cases use a limited knowledge of the internal design structure of the EUT. |
| GUI testing | The purpose of the test is to determine if the graphical user interface operates as expected. |
| Guilds | A "community of interest" group that welcomes anyone and usually cuts across an entire organization. Similar to a Community of Practice. |
| Hand Offs | The procedure for transferring the responsibility of a particular task from one individual or team to another. |
| Hardening | Securing a server or infrastructure environment by removing or disabling unnecessary software, updating to known good versions of the operating system, restricting network-level access to only that which is needed, configuring logging in order to capture alerts, configuring appropriate access management, and installing appropriate security tools. |
| Helm Chart Registry | Helm charts are what describe related Kubernetes resources. Artifactory and Codefresh support a registry for maintaining master records of Helm Charts. |
| Heritage Reliability Engineer (HRE) | Applying the principles and practices of SRE to legacy applications and environments. |

| | |
|---|---|
| High-Trust Culture | Organizations with a high-trust culture encourage good information flow, cross-functional collaboration, shared responsibilities, learning from failures and new ideas. |
| Horizontal Scaling | Computing resources are scaled wider to increase the volume of processing. E.g. Add more computers and run more tasks in parallel. |
| Hypothesis-Backlog | A collection of requirements expressed as experiments. |
| Hypothesis-Driven Development (HDD) | A prototype methodology that allows product designers to develop, test, and rebuild a product until it's acceptable to the users. |
| Idempotent | CM tools (e.g., Puppet, Chef, Ansible, and Salt) claim that they are 'idempotent' by allowing the desired state of a server to be defined as code or declarations and automate steps necessary to consistently achieve the defined state time-after-time. |
| Identity | The unique name of a person, device, or the combination of both that is recognized by a digital system. Also referred to as an "account" or "user." |
| Identity and Access Management (IAM) | Policies, procedures, and tools for ensuring the right people have the right access to technology resources. |
| Identity as a Service (IDaaS) | Identity and access management services that are offered through the cloud or on a subscription basis. |
| Image-based test selection method | Build images are pre-assigned test cases. Tests cases are selected for a build by matching the image changes resulting from a build. |
| Immersive learning | A learning approach that guides teams with coaching and practice to help them learn to work in a new way. |
| Immutable | An immutable object is an object whose state cannot be modified after it is created. The antonym is a mutable object, which can be modified after it is created. |
| Immutable Infrastructures | Instead of instantiating an instance (server, container, etc.), with error-prone, time-consuming patches and upgrades (i.e. mutations), replace it with another instance to introduce changes or ensure proper behavior. |
| Impact-Driven Development (IDD) | A software development methodology that takes small steps towards achieving both impact and vision. |

| | |
|---|---|
| Implementation Under Test | The EUT is a software implementation. E.g. Embedded program is being tested. |
| Improvement Kata | A structured way to create a culture of continuous learning and improvement. (In Japanese business, Kata is the idea of doing things the "correct" way. An organization's culture can be characterized as its Kata through its consistent role modeling, teaching and coaching.) |
| Incentive model | A system designed to motivate people to complete tasks toward achieving objectives. The system may employ either positive or negative consequences for motivation. |
| Incident | Any unplanned interruption to an IT service or reduction in the quality of an IT service. Includes events that disrupt or could disrupt the service. (ITIL definition) |
| Incident Management | A process that restores normal service operation as quickly as possible to minimize business impact and ensure that agreed levels of service quality are maintained. (ITIL definition). Involves capturing the who, what, when of service incidents and the onward use of this data in ensuring service level objectives are being met. |
| Incident Response | An organized approach to addressing and managing the aftermath of a security breach or attack (also known as an incident). The goal is to handle the situation in a way that limits damage and reduces recovery time and costs. |
| Increment | Potentially shippable completed work that is the outcome of a Sprint. |
| Incremental Rollout | Deploying many small, gradual changes to a service instead of a few large changes. Users are incrementally moved across to the new version of the service until eventually all users are moved across. Sometimes referred to by colored environments e.g. Blue/green deployment. |
| Infrastructure | All of the hardware, software, networks, facilities, etc., required to develop, test, deliver, monitor and control or support IT services. The term IT infrastructure includes all of the information technology but not the associated people, processes, and documentation. (ITIL definition) |
| Infrastructure as Code (IaC) | The practice of using code (scripts) to configure and manage infrastructure. |
| Infrastructure Test | The purpose of the test is to verify the framework for EUT operating. E.g. verify specific operating system utilities function as expected in the target environment. |

| | |
|---|---|
| Infrastructure-as-a-Service (IaaS) | On-demand access to a shared pool of configurable computing resources. |
| Impact-Driven Development | A software development approach that takes small steps towards achieving both impact and vision. |
| Insights Driven | An insight-driven organization embeds analysis, data, and reasoning into the decision-making process, every day. |
| Integrated development environment (IDE) | An integrated development environment (IDE) is a software suite that consolidates the basic tools developers need to write and test software. Typically, an IDE contains a code editor, a compiler or interpreter, and a debugger that the developer accesses through a single graphical user interface (GUI). An IDE may be a standalone application, or it may be included as part of one or more existing and compatible applications. (TechTarget) |
| Integrated development environment (IDE) 'lint' checks | Linting is the process of running a program that will analyze code for potential errors (e.g., formatting discrepancies, non-adherence to coding standards and conventions, logical errors). |
| Internet of Things | A network of physical devices that connect to the internet and potentially to each other through web-based wireless services. |
| Internal Automation | Scripts and automation delivered as part of the service that is intended to reduce toil. |
| INVEST | A mnemonic was created by Bill Wake as a reminder of the characteristics of a quality user story. |
| ISO 31000 | A family of standards that provide principles and generic guidelines on risk management. |
| Issue Management | A process for capturing, tracking, and resolving bugs and issues throughout the software development lifecycle. |
| IT Service Management (ITSM) | Adopting a process approach towards management, focusing on customer needs and IT services for customers rather than IT systems, and stressing continual improvement. (Wikipedia) |
| iTest | Tool licensed by Spirent Communications for creating automated test cases. |

| | |
|---|---|
| ITIL | Provides a best practices framework that organizations can adapt to deliver and maintain IT services to provide optimal value for all stakeholders, including the customer. |
| Jenkins | Jenkins is a freeware tool. It is the most popular master automation framework tool, especially for continuous integration task automation. Jenkins task automation centers around timed processes. Many test tools and other tools offer plugins to simplify integration with Jenkins. |
| Kaizen | The practice of continuous improvement. |
| Kanban | Method of work that pulls the flow of work through a process at a manageable pace. |
| Kanban Board | Tool that helps teams organize, visualize and manage work. |
| Karpman Drama Triangle | The drama triangle is a social model of human interaction. The triangle maps a type of destructive interaction that can occur between people in conflict. |
| Key Metrics | Something that is measured and reported upon to help manage a process, IT service or activity. |
| Key Performance Indicator (KPI) | Key performance indicators are the critical indicators of progress toward an intended result, providing a focus for improvement, and on what matters most. |
| Keywords-Based | Test cases are created using pre-defined names that reference programs useful for testing. |
| Knowledge Management | A process that ensures the right information is delivered to the right place or person at the right time to enable an informed decision. |
| Known Error | Problem with a documented root cause and a workaround. (ITIL definition) |
| Kolb's Learning Styles | David Kolb published his learning styles model in 1984; his experiential learning theory works on two levels: a four-stage cycle of learning and four separate learning styles. |
| Kotter's Dual Operating System | John Kotter describes the need for a dual operating system that combines the entrepreneurial capability of a network with the organizational efficiency of traditional hierarchy. |

| | |
|---|---|
| Kubernetes | Kubernetes is an open-source container-orchestration system for automating application deployment, scaling, and management. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation. |
| Kubler-Ross Change Curve | Describes and predicts the stages of personal and organizational reaction to major changes. |
| Lab-as-a-Service (LaaS) | Category of cloud computing services that provides a laboratory allowing customers to test applications without the complexity of building and maintaining the lab infrastructure. |
| Laloux (Culture Models) | Frederic Laloux created a model for understanding organizational culture. |
| Latency | Latency is the delay incurred in communicating a message, the time a message spends "on the wire" between the initial request being received e.g. by a server, and the response being received e.g. by a client. |
| Laws of Systems Thinking | In his book, 'The Fifth Discipline', Peter Senge outlines eleven laws that will help the understanding of business systems and to identify behaviors for addressing complex business problems. |
| Lean | Production philosophy that focuses on reducing waste and improving the flow of processes to improve overall customer value. |
| Lean (adjective) | Spare, economical. Lacking richness or abundance. |
| Lean Canvas | Lean Canvas is a 1-page business plan template. |
| Lean Enterprise | An organization that strategically applies the key ideas behind lean production across the enterprise. |
| Lean IT | Applying the key ideas behind lean production to the development and management of IT products and services. |
| Lean Manufacturing | Lean production philosophy derived mostly from the Toyota Production System. |

| | |
|---|---|
| Lean Product Development | Lean Product Development, or LPD, utilizes Lean principles to meet the challenges of Product Development. |
| Lean Startup | A system for developing a business or product in the most efficient way possible to reduce the risk of failure. |
| License Scanning | Tools, such as Blackduck and Synopsis, that check that licenses of your dependencies are compatible with your application, and approve or blacklist them. |
| Little's Law | A theorem by John Little that states that the long-term average number $L$ of customers in a stationary system is equal to the long-term average effective arrival rate $\lambda$ multiplied by the average time $W$ that a customer spends in the system. |
| LoadRunner | A tool used to test applications, measuring system behavior, and performance under load. Licensed by HP. |
| Log | Serialized report of details such as test activities and EUT console logs. |
| Log Management | The collective processes and policies used to administer and facilitate the generation, transmission, analysis, storage, archiving, and ultimate disposal of the large volumes of log data created within an information system. |
| Logging | The capture, aggregation, and storage of all logs associated with system performance including, but not limited to, process calls, events, user data, responses, error, and status codes. Logstash and Nagios are popular examples. |
| Logic Bomb (Slag Code) | A string of malicious code used to cause harm to a system when the programmed conditions are met. |
| Longevity Test | The purpose of the test is to determine if a complete system performs as expected over an extended period of time |
| Machine Learning | Data analysis that uses algorithms that learn from data. |
| Malware | A program designed to gain access to computer systems, normally for the benefit of some third party, without the user's permission |
| Many-factor Authentication | The practice of using at least 2 factors for authentication. The two factors can be of the same class. |

| | |
|---|---|
| Mean Time Between Deploys | Used to measure deployment frequency. |
| Mean Time Between Failures (MTBF) | The average time that a CI or IT service can perform its agreed function without interruption. Often used to measure reliability. Measured from when the CI or service starts working, until the time it fails (uptime). (ITIL definition) |
| Mean Time to Detect Defects (MTTD) | Average time required to detect a failed component or device. |
| Mean Time to Discovery | How long a vulnerability or software bug/defect exists before it's identified. |
| Mean Time to Patch | How long it takes to apply patches to environments once a vulnerability has been identified. |
| Mean Time to Repair/Recover (MTTR) | Average time required to repair/recover a failed component or device. MTTR does not include the time required to recover or restore service. |
| Mean Time to Restore Service (MTRS) | Used to measure time from when the CI or IT service fails until it is fully restored and delivering its normal functionality (downtime). Often used to measure maintainability. (ITIL definition). |
| Mental Models | A mental model is an explanation of someone's thought process about how something works in the real world. |
| Merge | The action of integrating software changes together into a software version management system. |
| Metric | Something that is measured and reported upon to help manage a process, IT service, or activity. |
| Metrics | This is a class of terms relevant to measurements used to monitor the health of a product or infrastructure. |

| | |
|---|---|
| Microprocess | A distinct activity that can be defined, designed, implemented, and managed independently and is generally associated with a primary service management practice.  A microprocess may be integrated with other service management practices. |
| Microprocess Architecture | A collection of integrated microprocesses that collectively perform all of the activities necessary for an end-to-end service management practice to be successful. |
| Microservices | A software architecture that is composed of smaller modules that interact through APIs and can be updated without affecting the entire system. |
| Mindset | A person's usual attitude or mental state is their mindset. |
| Minimum Viable Process | The least amount needed in order for this process or microprocess to meet its Definition of Done. |
| Minimum Viable Product | Most minimal version of a product that can be released and still provide enough value that people are willing to use it. |
| Mock Object | Mock is a method/object that simulates the behavior of a real method/object in controlled ways. Mock objects are used in unit testing. Often a method under a test calls other external services or methods within it. These are called dependencies. |
| Model | Representation of a system, process, IT service, CI, etc. that is used to help understand or predict future behavior. In the context of processes, models represent pre-defined steps for handling specific types of transactions. |
| Model-Based | Test cases are automatically derived from a model of the entity under test. Example tool: Tricentis |
| Monitoring | The use of a hardware or software component to monitor the system resources and performance of a computer service. |
| Monitoring Tools | Tools that allow IT organizations to identify specific issues of specific releases and to understand the impact on end-users. |
| Monolithic | A software system is called "monolithic" if it has a monolithic architecture, in which functionally distinguishable aspects (for example data input and output, data processing, error handling, and the user interface) are all interwoven, rather than containing architecturally separate components. |
| Multi-factor Authentication | The practice of using 2 or more factors for authentication. Often used synonymously with 2-factor Authentication. |

| | |
|---|---|
| Multi-cloud | Multi-cloud DevOps solutions provide on-demand multi-tenant access to development and test environments. |
| Network Reliability Engineer (NRE) | Someone who applies a reliability engineering approach to measure and automate the reliability of networks. |
| Neuroplasticity | Describes the ability of the brain to form and reorganize synaptic connections, especially in response to learning or experience or following injury. |
| Neuroscience | The study of the brain and nervous system. |
| Non-functional requirements | Requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors or functions (e.g., availability, reliability, maintainability, supportability); qualities of a system. |
| Non-functional tests | Defined as a type of service testing intending to check non-functional aspects such as performance, usability, and reliability of a software service. |
| Object Under Test (OUT) | The EUT is a software object or class of objects. |
| Observability | Observability is focused on externalizing as much data as you can about the whole service allowing us to infer what the current state of that service is. |
| Objectives and Key Results (OKRs) | Objectives and key results is a goal-setting framework used by individuals, teams, and organizations to define measurable goals and track their outcomes. |
| On-call | Being on-call means someone being available during a set period of time, and being ready to respond to production incidents during that time with appropriate urgency. |
| Open Source | Software that is distributed with its source code so that end-user organizations and vendors can modify it for their own purposes. |
| Operations (Ops) | Individuals involved in the daily operational activities needed to deploy and manage systems and services such as quality assurance analysts, release managers, system and network administrators, information security officers, IT operations specialists, and service desk analysts. |
| Operations Management | The function that performs the daily activities needed to deliver and support IT services and the supporting IT infrastructure at the agreed levels. (ITIL) |

| | |
|---|---|
| Ops | Individuals involved in the daily operational activities needed to deploy and manage systems and services such as quality assurance analysts, release managers, system and network administrators, information security officers, IT operations specialists, and service desk analysts. |
| Orchestration | An approach to building automation that interfaces or "orchestrates" multiple tools together to form a toolchain. |
| Organization Culture | A system of shared values, assumptions, beliefs, and norms that unite the members of an organization. |
| Organization Model | For DevOps, an approach that models Spotify's Squad approach for organizing IT. |
| Organizational Change | Efforts to adapt the behavior of humans within an organization to meet new structures, processes, or requirements. |
| OS Virtualization | A method for splitting a server into multiple partitions called "containers" or "virtual environments" in order to prevent applications from interfering with each other. |
| Outcome | Intended or actual results. |
| Outcome Mapping | A methodology for planning, monitoring, and evaluating development initiatives in order to bring about sustainable change. |
| Package Registry | A repository for software packages, artifacts, and their corresponding metadata. Can store files produced by an organization itself or for third-party binaries. Artifactory and Nexus are amongst the most popular. |
| Pages | Something for creating supporting web pages automatically as part of a CI/CD pipeline. |
| Patch | A software update designed to address (mitigate/remediate) a bug or weakness. |
| Patch management | The process of identifying and implementing patches. |
| Pathological Culture | Pathological cultures tend to view information as a personal resource, to be used in political power struggles (Westrum). |
| Penetration Testing | An authorized simulated attack on a computer system that looks for security weaknesses, potentially gaining access to the system's features and data. |

| | |
|---|---|
| People Changes | Focuses on changing attitudes, behaviors, skills, or performance of employees. |
| Performance Test | The purpose of the test is to determine an EUT meets its system performance criterion or to determine what a system's performance capabilities are. |
| Plan-Do-Check-Act | A four-stage cycle for process management and improvement attributed to W. Edwards Deming. Sometimes called the Deming Cycle or PDCA. |
| Platform-as-a-Service (PaaS) | Category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure. |
| Plugin | A pre-programmed integration between an orchestration tool and other tools. For example, many tools offer plugins to integrate with Jenkins. |
| Policies | Formal documents that define boundaries in terms of what the organization may or may not do as part of its operations. |
| Policy as Code | The notion that security principles and concepts can be articulated in code (e.g., software, configuration management, automation) to a sufficient degree that the need for an extensive traditional policy framework is greatly reduced. Standards and guidelines should be implemented in code and configuration, automatically enforced, and automatically reported on in terms of compliance, variance, or suspected violations. |
| Practice | A complete end-to-end capability for managing a specific aspect of service delivery (e.g. changes, incidents, service levels). |
| Practice Backlog | A prioritized list of everything that needs to be designed or improved for a practice including current and future requirements. |
| Practice/Microprocess Planning | A high-level event to define the goals, objectives, inputs, outcomes, activities, stakeholders, tools, and other aspects of a practice or microprocess. This meeting is not timeboxed. |
| Pre-Flight | This is a class of terms that refers to names of activities and processes that are conducted on an EUT prior to integration into the trunk branch. |
| Priority | The relative importance of an incident, problem, or change; based on impact and urgency. (ITIL definition) |

| | |
|---|---|
| Privileged Access Management (PAM) | Technologies that help organizations provide secured privileged access to critical assets and meet compliance requirements by securing, managing, and monitoring privileged accounts and access. (Gartner) |
| Problem | The underlying cause of one or more incidents. (ITIL definition) |
| Process | A structured set of activities designed to accomplish a specific objective. A process takes inputs and turns them into defined outputs. Related work activities that take specific inputs and produce specific outputs that are of value to a customer. |
| Process Changes | Focuses on changes to standard IT processes, such as software development practices, ITIL processes, change management, approvals, etc. |
| Process Owner | A role accountable for the overall quality of a process. It may be assigned to the same person who carries out the Process Manager role, but the two roles may be separate in larger organizations. (ITIL definition) |
| Process Standup | A time-boxed event of 15 minutes to inspect progress towards the Sprint Goal and identify impediments as quickly as possible. |
| Processing Time | The period during which one or more inputs are transformed into a finished product by a manufacturing or development procedure. (Business Dictionary) |
| Product Backlog | Prioritized list of functional and non-functional requirements for a system usually expressed as user stories. |
| Product Owner | An individual responsible for maximizing the value of a product and for managing the product backlog. Prioritizes, grooms, and owns the backlog. Gives the squad purpose. |
| Programming-Based | Test cases are created by writing code in a programming language. E.g. JavaScript, Python, TCL, Ruby |
| Project to Product | Changing ways of working from a large batch, waterfall project led approach, to a small batch, agile product (or value stream) approach. |
| Provision Platforms | Tools that provide platforms for provisioning infrastructure (e.g., Puppet, Chef, Salt). |
| Psychological Safety | Psychological safety is a shared belief that the team is safe for interpersonal risk-taking. |

| | |
|---|---|
| QTP | Quick Test Professional is a functional and regression test automation tool for software applications. Licensed by HP. |
| Quality Management | Tools that handle test case planning, test execution, defect tracking (often into backlogs), severity, and priority analysis. CA's Agile Central |
| Ranorex | GUI test automation framework for testing of desktop, web-based and mobile applications. Licensed by Ranorex. |
| Ransomware | Encrypts the files on a user's device or a network's storage devices. To restore access to the encrypted files, the user must pay a "ransom" to the cybercriminals, typically through a tough-to-trace electronic payment method such as Bitcoin. |
| RASP | Runtime Application Self-Protection |
| Regression testing | The purpose of the test is to determine if a new version of an EUT has broken some things that worked previously. |
| Regulatory compliance testing | The purpose of the test is to determine if an EUT conforms to specific regulatory requirements. E.g. verify an EUT satisfies government regulations for consumer credit card processing. |
| Release | Software that is built, tested, and deployed into the production environment. |
| Release Acceptance Criteria | Measurable attributes for a release package that determine whether a release candidate is acceptable for deployment to customers. |
| Release Candidate | A release package that has been prepared for deployment, may or may not have passed the Release. |
| Release Governance | Release Governance is all about the controls and automation (security, compliance, or otherwise) that ensure your releases are managed in an auditable and trackable way, in order to meet the need of the business to understand what is changing. |
| Release Management | The process that manages releases and underpins Continuous Delivery and the Deployment Pipeline. |

| | |
|---|---|
| Release Orchestration | Typically a deployment pipeline used to detect any changes that will lead to problems in production. Orchestrating other tools will identify performance, security, or usability issues. Tools like Jenkins and Gitlab CI can "orchestrate" releases. |
| Relevance | A Continuous Testing tenet which emphasizes a preference to focus on the most important tests and test results |
| Reliability | A measure of how long a service, component, or CI can perform its agreed function without interruption. Usually measured as MTBF or MTBSI. (ITIL definition) |
| Reliability Test | The purpose of the test is to determine if a complete system performs as expected under stressful and loaded conditions over an extended period of time. |
| Remediation | Action to resolve a problem found during DevOps processes. E.g.<br>Roll-back changes for an EUT change that resulted in a CT test case fail verdict. |
| Remediation Plan | A plan that determines the actions to take after a failed change or release. (ITIL definition) |
| Request for Change (RFC) | Formal proposal to make a change. The term RFC is often misused to mean a change record, or the change itself. (ITIL definition) |
| Requirements Management | Tools that handle requirements definition, traceability, hierarchies & dependency. Often also handles code requirements and test cases for requirements. |
| Resilience | Building an environment or organization that is tolerant to change and incidents. |
| Response Time | Response time is the total time it takes from when a user makes a request until they receive a response. |
| REST | Representation State Transfer. The software architecture style of the worldwide web. |
| Restful API | Representational state transfer (REST) or RESTful services on a network, such as HTTP, provide scalable interoperability for requesting systems to quickly and reliably access and manipulate textual representations (XML, HTML, JSON) of resources using stateless operations (GET, POST, PUT, DELETE, etc.). |

| | |
|---|---|
| RESTful interface testing | The purpose of the test is to determine if an API satisfies its design criterion and the expectations of the REST architecture. |
| Return on Investment (ROI) | The difference between the benefit achieved and the cost to achieve that benefit, expressed as a percentage. |
| Review Apps | Allow code to be committed and launched in real-time – environments are spun up to allow developers to review their application. |
| Rework | The time and effort required to correct defects (waste). |
| Risk | A possible event that could cause harm or loss or affect an organization's ability to achieve its objectives. The management of risk consists of three activities: identifying risks, analyzing risks, and managing risks. The probable frequency and probable magnitude of future loss. Pertains to a possible event that could cause harm or loss or affect an organization's ability to execute or achieve its objectives. |
| Risk Event | A possible event that could cause harm or loss or affect an organization's ability to achieve its objectives. The management of risk consists of three activities: identifying risks, analyzing risks, and managing risks. |
| Risk Management Process | The process by which "risk" is contextualized, assessed and treated. From ISO 31000: 1) Establish context, 2) Assess risk, 3) Treat risk (remediate, reduce or accept). |
| Robot Framework | TDD framework created and supported by Google. |
| Role | Set of responsibilities, activities, and authorities granted to a person or team. A role is defined by a process. One person or team may have multiple roles. A set of permissions assigned to a user or group of users to allow a user to perform actions within a system or application. |
| Role-based Access Control (RBAC) | An approach to restricting system access to authorized users. |
| Roll-back | Software changes which have been integrated are removed from the integration. |
| Root Cause Analysis (RCA) | Actions take to identify the underlying cause of a problem or incident. |

| | |
|---|---|
| Rugged Development (DevOps) | Rugged Development (DevOps) is a method that includes security practices as early in the continuous delivery pipeline as possible to increase cybersecurity, speed, and quality of releases beyond what DevOps practices can yield alone. |
| Rugged DevOps | Rugged DevOps is a method that includes security practices as early in the continuous delivery pipeline as possible to increase cybersecurity, speed, and quality of releases beyond what DevOps practices can yield alone. |
| Runbooks | A collection of procedures necessary for the smooth operation of a service. Previously manual in nature they are now usually automated with tools like Ansible. |
| Runtime Application Self Protection (RASP) | Tools that actively monitor and block threats in the production environment before they can exploit vulnerabilities. |
| Sanity Test | A very basic set of tests that determine if a software is functional at all. |
| Scalability | Scalability is a characteristic of a service that describes its capability to cope and perform under an increased or expanding load. |
| Scaled Agile Framework (SAFE) | A proven, publicly available, framework for applying Lean-Agile principles and practices at an enterprise scale. |
| SCARF Model | A summary of important discoveries from neuroscience about the way people interact socially. |
| Scheduling | Scheduling: the process of planning to release changes into production. |
| Scrum | A simple framework for effective team collaboration on complex projects. Scrum provides a small set of rules that create "just enough" structure for teams to be able to focus their innovation on solving what might otherwise be an insurmountable challenge. (Scrum.org) |
| Scrum Pillars | Pillars that uphold the Scrum framework include Transparency, Inspection, and Adaption. |
| Scrum Team | A self-organizing, cross-functional team that uses the Scrum framework to deliver products iteratively and incrementally. The Scrum Team consists of a Product Owner, Developers, and a Scrum Master. |
| Scrum Values | A set of fundamental values and qualities underpinning the Scrum framework: commitment, focus, openness, respect and courage. |

| | |
|---|---|
| Scrum Master | An individual who provides process leadership for Scrum (i.e., ensures Scrum practices are understood and followed) and who supports the Scrum Team by removing impediments. |
| Secret Detection | Secret Detection aims to prevent that sensitive information, like passwords, authentication tokens, and private keys are unintentionally leaked as part of the repository content. |
| Secrets Management | Secrets management refers to the tools and methods for managing digital authentication credentials (secrets), including passwords, keys, APIs, and tokens for use in applications, services, privileged accounts, and other sensitive parts of the IT ecosystem. |
| Secure Automation | Secure automation removes the chance of human error (and wilful sabotage) by securing the tooling used across the delivery pipeline. |
| Security (Information Security) | Practices intended to protect the confidentiality, integrity, and availability of computer system data from those with malicious intentions. |
| Security as Code | Automating and building security into DevOps tools and practices, making it an essential part of toolchains and workflows. |
| Security tests | The purpose of the test is to determine if an EUT meets its security requirements. An example is a test that determines if an EUT processes login credentials properly. |
| Selenium | Popular open-source tool for software testing GUI and web applications. |
| Self-healing | Self-healing means the ability of services and underlying environments to detect and resolve problems automatically. It eliminates the need for manual human intervention. |
| Serverless | A code execution paradigm where no underlying infrastructure or dependencies are needed, moreover, a piece of code is executed by a service provider (typically cloud) who takes over the creation of the execution environment. Lambda functions in AWS and Azure Functions are examples. |
| Service | Enables the ability to do something when and how it is needed or desired. It enables its customers to achieve their objectives more efficiently and/or more effectively than they could without the service. |
| Service Desk | Single point of contact between the service provider and the users.  Tools like Service Now are used for managing the lifecycle of services as well as internal and external stakeholder engagement. |

| | |
|---|---|
| Service Level Agreement (SLA) | Written agreement between an IT service provider and its customer(s) that defines key service targets and responsibilities of both parties. An SLA may cover multiple services or customers. (ITIL definition) |
| Service Level Indicator (SLI) | SLI's are used to communicate quantitative data about services, typically to measure how the service is performing against an SLO. |
| Service Level Objective (SLO) | An SLO is a goal for how well a product or service should operate. SLO's are set based on what an organization is expecting from a service. |
| Seven Pillars of DevOps | Seven distinct "pillars" provide a foundation for DevOps systems which include Collaborative Culture, Design for DevOps, Continuous Integration, Continuous Testing, Continuous Delivery and Deployment, Continuous Monitoring, and Elastic Infrastructure and Tools. |
| Shift Left | An approach that strives to build quality into the software development process by incorporating testing early and often. This notion extends to security architecture, hardening images, application security testing, and beyond. |
| SilkTest | Automated function and regression testing of enterprise applications. Licensed by Borland. |
| Simian Army | The Simian Army is a suite of failure-inducing tools designed by Netflix. The most famous example is Chaos Monkey which randomly terminates services in production as part of a Chaos Engineering approach. |
| Single Point of Failure (SPOF) | A single point of failure (SPOF) is a part of a system that, if it fails, will stop the entire system from working. |
| Site Reliability Engineering (SRE) | The discipline that incorporates aspects of software engineering and applies them to infrastructure and operations problems. The main goals are to create scalable and highly reliable software systems. |
| Smoke Test | A basic set of functional tests that are run immediately after a software component is built. Same as CI Regression Test. |
| Snapshot | Report of pass/fail results for a specific build. |
| Snippets | Stored and shared code snippets to allow collaboration around specific pieces of code. Also allows code snippets to be used in other code-bases. BitBucket and GitLab allow this. |

| | |
|---|---|
| SOAP | Simple Object Access Protocol (SOAP) is an XML-based messaging protocol for exchanging information among computers. |
| Software Composition Analysis | A tool that checks for libraries or functions in source code that have known vulnerabilities. |
| Software Defined Networking (SDN) | Software-Defined Networking (SDN) is a network architecture approach that enables the network to be intelligently and centrally controlled, or 'programmed,' using software applications. |
| Software Delivery Lifecycle (SDLC) | The process used to design, develop and test high quality software. |
| Software Version Management System | A repository tool which is used to manage software changes. Examples are: Azure DevOps, BitBucket, Git, GitHub, GitLab, VSTS. |
| Software-as-a-Service (SaaS) | Category of cloud computing services in which software is licensed on a subscription basis. |
| Source Code Tools | Repositories for controlling source code for key assets (application and infrastructure) as a single source of truth. |
| Spotify Squad Model | An organizational model that helps teams in large organizations behave like startups and be nimble. |
| Sprint | A period of 2-4 weeks during which an increment of product work is completed. |
| Sprint (Scrum) | A time-boxed iteration of work during which an increment of product functionality is implemented. |
| Sprint Backlog | Subset of the backlog that represents the work that must be completed to realize the Sprint Goal. |
| Sprint Goal | The purpose and objective of a Sprint, often expressed as a business problem that is going to be solved. |

| | |
|---|---|
| Sprint Planning | A 4 to 8-hour time-boxed event that defines the Sprint Goal, the increment of the Product Backlog that will be completed during the Sprint, and how it will be completed. |
| Sprint Retrospective | A 1.5 to 3-hour time-boxed event during which the Team reviews the last Sprint and identifies and prioritizes improvements for the next Sprint. |
| Sprint Review | A time-boxed event of 4 hours or less where the Team and stakeholders inspect the work resulting from the Sprint and update the Product Backlog. |
| Spyware | Software that is installed in a computer without the user's knowledge and transmits information about the user's computer activities over back to the threat agent. |
| Squads | A cross-functional, co-located, autonomous, self-directed team. |
| Stakeholder | Person who has an interest in an organization, project or IT service. Stakeholders may include customers, users and suppliers. (ITIL definition). |
| Stability | The sensitivity a service has to accept changes and the negative impact that may be caused by system changes. Services may have reliability, in that if functions over a long period of time, but may not be easy to change and so does not have stability. |
| Standard Change | Pre-approved, low risk change that follows a procedure or work instruction. (ITIL definition) |
| Static Application Security Testing (SAST) | A type of testing that checks source code for bugs and weaknesses. |
| Static Code Analysis | The purpose of the test is to detect source code logic errors and omissions such as memory leaks, unutilized variables, unutilized pointers. |
| Status Page | Service pages that easily communicate the status of services to customers and users. |
| Sticks | Negative incentives, for discouraging or punishing undesired behaviors. |
| Storage Security | A specialty area of security that is concerned with securing data storage systems and ecosystems and the data that resides on these systems. |
| Stormstack | A commercial orchestration tool based on event triggers instead of time-based. |
| StoStaKee | This stands for stop, start, and keep: this is an interactive time-boxed exercise focused on past events. |

| | |
|---|---|
| Strategic Sprint | A <4 week timeboxed Sprint during which strategic elements that were defined during Practice Planning are completed so that the Team can move on to designing the activities of the process. |
| Stream-Aligned Team | A team aligned to a single, valuable stream of work; this might be a single product or service, a single user story, or a single user persona. |
| Structural Changes | Changes in the hierarchy of authority, goals, structural characteristics, administrative procedures, and management systems. |
| Supplier | External (third party) supplier, manufacturer, or vendor responsible for supplying goods or services that are required to deliver IT services. |
| Synthetic Monitoring | Synthetic monitoring (also known as active monitoring, or semantic monitoring) runs a subset of an application's automated tests against the system on a regular basis. The results are pushed into the monitoring service, which triggers alerts in case of failures. |
| System of Record | A system of record is the authoritative data source for a data element or data entity. |
| System Test | The purpose of the test is to determine if a complete system performs as expected in its intended configurations. |
| System Under Test (SUT) | The EUT is an entire system. E.g. Bank teller machine is being tested. |
| Tag-Based Test Selection Method | Tests and Code modules are pre-assigned tags. Tests are selected for a build matching pre-assigned tags. |
| Target Operating Model | A description of the desired state of the operating model of an organization. |
| Teal Organization | An emerging organizational paradigm that advocates a level of consciousness including all previous world views within the operations of an organization. |
| Team Dynamics | A measurement of how a team works together. Includes team culture, communication styles, decision-making ability, trust between members, and the willingness of the team to change. |
| Team Topologies | An approach to organizing business and technology teams for fast flow, providing a practical, step-by-step, adaptive model for organizational design and team interaction. |

| | |
|---|---|
| Techno-Economic Paradigm Shifts | Techno-economic paradigm shifts are at the core of the general, innovation-based theory of economic and societal development as conceived by Carlota Perez. |
| Telemetry | Telemetry is the collection of measurements or other data at remote or inaccessible points and their automatic transmission to receiving equipment for monitoring. |
| Test Architect | Person who has responsibility for defining the overall end-to-end test strategy for an EUT. |
| Test Artifact Repository | Database of files used for testing. |
| Test Campaign | A test campaign may include one or more test sessions. |
| Test Case | Set of test steps together with data and configuration information. A test case has a specific purpose to test at least one attribute of the EUT. |
| Test Creation Methods | This is a class of test terms that refers to the methodology used to create test cases. |
| Test-Driven Development (TDD) | Test-driven development (TDD) is a software development process in which the developer writes a test before composing code. They then follow this process:<br>1. Write the test<br>2. Run the test and any others that are relevant and see them fail<br>3. Write the code<br>4. Run test(s)<br>5. Refactor code if needed<br>6. Repeat<br>Unit level tests and/or application tests are created ahead of the code that is to be tested. |
| Test Duration | The time it takes to run a test. E.g. # hours per test |

| | |
|---|---|
| Test Environment | The test environment refers to the operating system (e.g. Linus, windows version, etc.), the configuration of software (e.g. parameter options), dynamic conditions (e.g. CPU and memory utilization), and physical environment (e.g. power, cooling) in which the tests are performed. |
| Test Fast | A CT tenet referring to accelerated testing. |
| Test Framework | A set of processes, procedures, abstract concept,s and environments in which automated tests are designed and implemented. |
| Test Harness | A tool which enables the automation of tests. It refers to the system test drivers and other supporting tools that requires to execute tests. It provides stubs and drivers which are small programs that interact with the software under test. |
| Test Hierarchy | This is a class of terms describes the organization of tests into groups. |
| Test Methodology | This class of terms identifies the general methodology used by a test. Examples are White Box, Black Box |
| Test result repository | Database of test results. |
| Test Results Trend-based | A matrix of correlation factors correlates test cases and code modules according to test results (verdict). |
| Test Roles | This class of terms identifies general roles and responsibilities for people relevant to testing. |
| Test Script | Automated test case. A single test script may be implemented with one or more test cases depending on the data. |
| Test Selection Method | This class of terms refers to the method used to select tests to be executed on a version of an EUT. |

| | |
|---|---|
| Test Session | Set of one or more test suites that are run together on a single build at a specific time. |
| Test Suite | Set of test cases that are run together on a single build at a specific time. |
| Test Trend | History of verdicts. |
| Test Type | The class which indicates the purpose of the test. |
| Test Version | The version of files used to test a specific build. |
| Tester | An individual who has the responsibility to test a system or service. |
| Testing Tools | Tools that verify code quality before passing the build. |
| The Advice Process | Any person deciding must seek advice from everyone meaningfully affected by the decision and people with expertise in the matter. Advice received must be taken into consideration, though it does not have to be accepted or followed. The objective of the advice process is not to form a consensus, but to inform the decision-maker so that they can make the best decision possible. Failure to follow the advice process undermines trust and unnecessarily introduces risk to the business. |
| The Checkbox Trap | The situation wherein an audit-centric perspective focuses exclusively on "checking the box" on compliance requirements without consideration for overall security objectives. |
| The Power of TED | The Power of TED* offers an alternative to the Karpman Drama Triangle with its roles of Victim, Persecutor, and Rescuer. The Empowerment Dynamic (TED) provides the antidote roles of Creator, Challenger, and Coach and a more positive approach to life's challenges. |
| The Sprint | A period of <4 weeks during which an increment of work is completed. |

| | |
|---|---|
| The Three Pillars of Empiricism | Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation. |
| The Three Ways | Key principles of DevOps – Flow, Feedback, Continuous experimentation, and learning. |
| Theory of Constraints | Methodology for identifying the most important limiting factor (i.e., constraint) that stands in the way of achieving a goal and then systematically improving that constraint until it is no longer the limiting factor. |
| Thomas Kilmann Inventory (TKI) | Measures a person's behavioral choices under certain conflict situations. |
| Threat Agent | An actor, human or automated, that acts against a system with intent to harm or compromise that system. Sometimes also called a "Threat Actor." |
| Threat Detection | Refers to the ability to detect, report, and support the ability to respond to attacks. Intrusion detection systems and denial-of-service systems allow for some level of threat detection and prevention. |
| Threat Intelligence | Information pertaining to the nature of a threat or the actions a threat may be known to be perpetrating. May also include "indicators of compromise" related to a given threat's actions, as well as a "course of action" describing how to remediate the given threat action. |
| Threat Modeling | A method that ranks and models potential threats so that the risk can be understood and mitigated in the context of the value of the application(s) to which they pertain. |
| Time to Insight Actioned | The time between having an idea, delivering it to the customer, learning and actioning the insight from that learning. |
| Time to Learning | The time between conceiving an idea and learning how it was received based on customer feedback. |
| Time to Market | The period of time between when an idea is conceived and when it is available to customers. |
| Time to Value | The measure of the time it takes for the business to realize value from a feature or service. |
| Time Tracking | Tools that allow for time to be tracked, either against individual issues or other work or project types. |

| | |
|---|---|
| Timebox | The maximum duration of a Scrum event. |
| Toil | A kind of work tied to running a production service that tends to be manual, repetitive, automatable, tactical, devoid of enduring value. |
| Tool | This class describes tools that orchestrate, automate, simulate and monitor EUT's and infrastructures. |
| Toolchain | A philosophy that involves using an integrated set of complimentary task-specific tools to automate an end-to-end process (vs. a single-vendor solution). |
| Touch Time | In a Lean Production system the touch time is the time that the product is actually being worked on, and value is being added. |
| Tracing | Tracing provides insight into the performance and health of a deployed application, tracking each function or microservice which handles a given request. |
| Traffic Volume | The amount of data sent and received by visitors to a service (e.g. a website or API). |
| Training From the Back of the Room | An accelerated learning model in line with agile values and principles using the 4Cs instructional design "map" (Connection, Concept, Concrete Practice, Conclusion). |
| Transformational Leadership | A leadership model in which leaders inspire and motivate followers to achieve higher performance by appealing to their values and sense of purpose, facilitating wide-scale organizational change (State of DevOps Report, 2017). |
| Tribe Lead | A senior technical leader that has broad and deep technical expertise across all the squads' technical areas. A group of squads working together on a common feature set, product, or service is a tribe in Spotify's definitions. |
| Tribes | A collection of squads with a long-term mission that work on/in a related business capability. |
| Trojan (horses) | Malware that carries out malicious operations under the appearance of a desired operation such as playing an online game. A Trojan horse differs from a virus because the Trojan binds itself to non-executable files, such as image files, audio files whereas a virus requires an executable file to operate. |
| Trunk | The primary source code integration repository for a software product. |

| | |
|---|---|
| Unit Test | The purpose of the test is to verify code logic. |
| Usability Test | The purpose of the test is to determine if humans have a satisfactory experience when using an EUT. |
| User | Consumer of IT services. Or, the identity asserted during authentication (aka username). |
| User and Entity Behavior Analytics (UEBA) | A machine learning technique to analyze normal and "abnormal" user behavior with the aim of preventing the latter. |
| User Story | A brief statement used to describe a requirement from a user's perspective. User stories are used to facilitate communication, planning, and negotiation activities between the stakeholders and the Agile Service Management Team. |
| Value Added Time | The amount of time spent on an activity that creates value (e.g., development, testing). |
| Value Cycle | The lifecycle stages of the value stream from ideation to value realization. |
| Value Efficiency | Being able to produce value with the minimum amount of time and resources. |
| Value Stream | All of the activities needed to go from a customer request to a delivered product or service. |
| Value Stream Map | Visually depicts the end-to-end flow of activities from the initial request to value creation for the customer. |
| Value Stream Mapping | A lean tool that depicts the flow of information, materials, and work across functional silos with an emphasis on quantifying waste, including time and quality. |
| Value Stream Management | Value Stream Management is a combination of people, processes, and technology that maps, optimizes, visualizes, measures, and governs business value flow through heterogeneous software delivery pipelines from idea through development and into production. |
| Value Stream Management Platform | Software that manages value streams. |

| | |
|---|---|
| Variable Speed IT | An approach where traditional and digital processes co-exist within an organization while moving at their own speed. |
| Velocity | The measure of the quantity of work done in a pre-defined interval. The amount of work an individual or team can complete in a given amount of time. |
| Verdict | Test result classified as Fail, Pass, or Inconclusive. |
| Version control tools | Ensure a 'single source of truth' and enable change control and tracking for all production artifacts. |
| Vertical Scaling | Computing resources are scaled higher to increase processing speed e.g. using faster computers to run more tasks faster. |
| Virus (Computer) | Malicious executable code attached to a file that spreads when an infected file is passed from system to system that could be harmless (but annoying) or it could modify or delete data. |
| Voice of the Customer (VOC) | A process that captures and analyzes customer requirements and feedback to understand what the customer wants. |
| Vulnerability | A weakness in a design, system, or application that can be exploited by an attacker. |
| Vulnerability Intelligence | Information describing a known vulnerability, including affected software by version, the relative severity of the vulnerability (for example, does it result in an escalation of privileges for a user role, or does it cause a denial of service), the exploitability of the vulnerability (how easy/hard it is to exploit), and sometimes current rate of exploitation in the wild (is it being actively exploited or is it just theoretical). This information will also often include guidance on what software versions are known to have remediated the described vulnerability. |
| Vulnerability management | The process of identifying and remediating vulnerabilities. |
| Wait Time | The amount of time wasted on waiting for work (e.g., waiting for development and test infrastructure, waiting for resources, waiting for management approval). |
| Waste (Lean Manufacturing) | Any activity that does not add value to a process, product or service. |

| | |
|---|---|
| Water-scrum-fall | A hybrid approach to application lifecycle management that combines waterfall and Scrum development can complete in a given amount of time. |
| Waterfall (Project Management) | A linear and sequential approach to managing software design and development projects in which progress is seen as flowing steadily (and sequentially) downwards (like a waterfall). |
| Weakness | An error in software that can be exploited by an attacker to compromise the application, system, or the data contained therein. Also called a vulnerability. |
| Web Application Firewall (WAF) | Tools that examine traffic being sent to an application and can block anything that looks malicious. |
| Web IDE | Tools that have a web client integrated development environment. Enables developer productivity without having to use a local development tool. |
| Westrum (Organization Types) | Ron Westrum developed a typology of organizational cultures that includes three types of organizations: Pathological (power-oriented), Bureaucratic (rule-oriented) and Generative (performance-oriented). |
| White-Box Testing (or Clear-, Glass-, Transparent-Box Testing or Structural Testing) | Test cases use extensive knowledge of the internal design structure or workings of an application, as opposed to its functionality (i.e. Black-Box Testing). |
| Whitelisting | Application whitelisting is the practice of specifying an index of approved software applications that are permitted to be present and active on a computer system. |
| Wicked Questions | Wicked questions are used to expose the assumptions which shape our actions and choices. They are questions that articulate the embedded, and often contradictory assumptions, we hold about an issue, a problem or a context. |
| Wiki | Knowledge sharing can be enabled by using tools like Confluence which create a rich Wiki of content |
| Wilber's Quadrants | A model that recognises four modes of general approach for human beings. Two axes are used: on one axis people tend towards individuality OR collectivity. |
| Work in Progress (WIP) | Any work that has been started but has not been completed. |

| | |
|---|---|
| Workaround | A temporary way to reduce or eliminate the impact of incidents or problems. May be logged as a known error in the Known Error Database. (ITIL definition). |
| World Café | Is a structured conversational process for knowledge sharing in which groups of people discuss a topic at several tables, with individuals switching tables periodically and getting introduced to the previous discussion at their new table by a "table host". |
| Worms (Computer) | Worms replicate themselves on a system by attaching themselves to different files and looking for pathways between computers. They usually slow down networks and can run by themselves (where viruses need a host program to run). |

# Value Stream Management Foundation

## Learning Material: Value Added Resources

This document provides links to articles, case stories, videos, research, books, and websites related to the Value Stream Management Foundation Course from the Value Stream Management Consortium.

This information includes all links to all of the resources included in the course (and more!) and is provided to enhance your understanding of VSM-related concepts and terms and is not examinable. Of course, there is a wealth of other videos, blogs, and case studies on the web. We welcome suggestions for additions.

| Articles and Papers | |
|---|---|
| Title/Link | Author |
| 10 Tips for Value Stream Identification | Adam Mattis |
| A Brief History of Lean | Lean Enterprise Institute |
| Adding Purpose to Scrum with Holacracy | Ben Linders |
| AARRR vs RARRA: Pirate Metrics Explained | Ben Crouch |
| Accounting for Value Creation and Encouraging the Rise of the Chief Value Officer | Charles Tilley and Kevin Dancey |
| Being a Value Stream Architect: A Q&A with Bryan Finster | Helen Beal |
| Boost Enterprise DevOps with Value Stream Management | Gil Cattelain |
| Business Marketing: Understanding What Customers Value | Harvard Business Review |
| Buyer Utility Map: Templates, Definition, and Usage | Creately |
| Calculated Web Application Metrics | Dynatrace |
| Common Mistakes in Applying Value Stream Mapping to Software | Gary Gruver |
| Completing the OODA Loop of DevOps | Adam Bowen |
| Data-Driven Value Stream Management | Tasktop |
| Decentralised Development: Common Pitfalls and how Value Stream Management can Avoid Them | Jeff Keyes |
| Defining Value Stream Management for SEO Business Owners | Connie Benton |
| Defining Value: The Most Ambiguous Term in Product Development | Jeff Gothelf |
| Delivering Business Nirvana with Value Stream Management | ServiceNow |
| Departments versus Value Streams | Nick Katko |
| DevOps and Theory of Constraints | Gilles Gravier |
| DevOps Culture (Part 1) | John Willis |

| | |
|---|---|
| DevOps for the Business: Enter 'Value Stream Management' for Enterprise Software Woes | Joe McKendrick |
| DevOps Handbook Series Part 2: Defining DevOps Teams | Stefan Thorpe |
| DevOps Insights Standard Dashboard | ServiceNow |
| DevOps is a Trojan Horse. Of the Indispensable Kind | Jan de Vries |
| DevOps Process: Visibility of Work in the Value Stream | Google Cloud |
| Empathy Map Templates | Creately |
| Enabling Teams to Take Responsibility of Their Value Stream | Bryant Schuck |
| From Darwin to DevOps:John Willis and Gene Kim Talk About Life After the Phoenix Project | Helen Beal |
| From Friction to Flow through Focus | Steve Pereira |
| Five Features of Successful Value Streams | Laureen Knudsen |
| From CFO to Chief Value Officer | Delphine Gibassier |
| Fuzzy Front End | Wikipedia |
| Getting More Out of Dependency Mapping | |
| Here's Why We Need Value Strearm Management | Romnick Acabado |
| History of Lean | Ted Hessing |
| How Change Managers Become Change Enablers | ServiceNow |
| How to Identify Your Value Streams: 6 Strategies for Getting Started | Emily Peterson |
| How Value Stream Management Uses Observability to Optimize Flow | Helen Beal |
| Hypothesis Development Canvas Version 0.4 | Bill Schmarzo |
| Hypothesis Driven Development (Practitioner's Guide) | Alexander Cowan |
| Impact Driven Development | Andrew Butel |
| Improve Value Streams with Insights | Micro Focus |
| Introducing the Next Generation Target Operating Model | McKinsey |

| | |
|---|---|
| ISO Standard 37000 ‘Guidance for the Governance of Organizations’ (draft) | ISO |
| IT Value Chain and IT4IT Reference Architecture | The Open Group |
| It’s Time to Learn the Missing Metric | Peter Koning |
| It’s Time to Retire the Net Promoter Score (And Here’s What to Replace it With) | Ron Shevlin |
| ITIL 4 – Value Streams: doing the right things for customers | Axelos |
| Lean Metrics | Ted Hessing |
| Lean Thinking in LeSS | LeSS |
| Mastering Software Delivery with Value Stream Management | Jeff Keyes |
| Mik+One Podcast Episode 17 John Willis (transcription) | Mik Kersten and John Willis |
| More Developers, More Problems: Don’t Let History Repeat Itself | Lee Reid and Patrick Anderson |
| Muda Be Gone! Value Stream Management Reduces DevOps Waste | Michael Baldani |
| Myth: If You Can’t Measure It, You Can’t Manage it | The Deming Institute |
| Net Promoter Score in a Nutshell | 4WeekMBA |
| Organizational Culture is Like an Iceberg | Torben Rick |
| Organizing Around Value Streams | Marcio Sete |
| Outcome Economy | Mary K Pratt |
| Outcome Mapping - How to Collaborate with Clarity | Steve Pereira and Shaaron Alvares |
| Portfolio Management: CSPs Financial Agility | Stephan Schneider, Thomas Knuchel and Dominik Zurbuchen |
| Product Roadmaps | Bree Davies |
| Q&A on the Book EDGE: Value-Driven Digital Transformation | Shane Hastie |
| Q&A with Jeff Keyes of Plutora on the Transformative Impact of Value Stream Management | Rupert Field |
| Qualitative versus Quantitative Data: What’s the Difference? | Devin Pickell |

| | |
|---|---|
| The 14 Principles of The Toyota Way | Jeffrey Liker |
| The Age of Software Needs Value Stream Architects | Mik Kersten |
| The Case for Value Stream Management | Jason Nelson |
| The Deming Cycle | Paul Arveson |
| The Dynamics of Technological Revolutions | Ewa A Treitz |
| The Emergence of the Chief Value Officer | Andrew Harding |
| The Emerging Market Category of Value Stream Management | Bob Davis |
| The Emerging Role of the Value Stream Manager: A Job Description | Scott White |
| The Five Principles of Lean | Doanh Do |
| The Future of DevOps Toolchains is Already Here, It's Just Not Evenly Distributed | Manjunath Bhat |
| The Hypothesis Prioritization Canvas | Jeff Gothelf |
| The Improvement Kata in 4 Simple Steps | The Leadership Network |
| The Lean Profit and Loss Statement - The Future of Accounting | David Blain |
| The Three Pillars of Empiricism | Hiren Doshi |
| The Role of Executives and Managers in Value Stream Management: Insight from Al Shalloway | Shaaron Alvares |
| The Value Stream Manager | Lean Enterprise Institute |
| Theory of Constraints 106: The Five Focusing Steps | Tiago Forte |
| Theory of Constraints: Using a Current Reality Tree | Stuart Rance |
| Thinking Environments | Mark Schwartz and Jason Cox |
| TOGAF Series Guide: Value Streams | The Open Group |
| Toyota Philosophy | Toyota |
| Understanding Value Streams with Karen Martin | Corey Brown |
| Value Stream | Wikipedia |
| Value Stream Accounting | John Parker |
| Value Stream Architecture | Mik Kersten and Nanda Kumar |

| | |
|---|---|
| Value Stream Definitions from the Lean Lexicon | Lean.org |
| Value Stream Engineer: Job Description and Salary | Study.com |
| Value Stream Identification: Lessons from the Wild | Carl Starendal |
| Value Stream Management: The good, bad, and ugly for Agile and DevOps leaders | Eveline Oehrlich |
| Value Stream Management | The International Journal of Logistics Management |
| Value Stream Management and Its Importance | Ksenija Kolomiiets and Alexandra Rostovtseva |
| Value Stream Management Essentials | Bryant Schuck and Steve Boone |
| Value Stream Management: Fact or Fiction? | Lisa Morgan |
| Value Stream Management for the Digital Age | Digital.ai |
| Value Stream Management: The Good, Bad, and Ugly for Agile and DevOps Leaders | Eveline Oehrlich |
| Value Stream Mapping Versus Value Stream Management: What's the Difference? | Peter Maddison |
| Value Stream Maps | Copado |
| Value Streams in Disciplined Agile | PMI |
| Value Streams in SAFe | Scaled Agile |
| Value Stream Management: The Key to Balancing Innovation and Operational Excellence | Tasktop |
| Value Streams — the Starting Point of Your Agile Transformation | Anca Onuta |
| Vision and Visibility: The Risk of Flying Blind | Helen Beal, Steve Pereira |
| What Do Digital Transformation Leaders Do Differently? | Ashok Reddy |
| What is a Leading and a Lagging Indicator? And Why You Need to Understand the Difference | Bernard Marr |
| What is a Key Performance Indicator? (KPI) | KPI.org |
| What is Real User Monitoring? | Sematext |
| What is the Kano Model? | ProductPlan |

| | |
|---|---|
| What is the Novel Economy and Why is it Important in a Post-Pandemic Economy | Brian Solis |
| What is Value Add Versus Non-Value-Add | Six Sigma Daily |
| What Is Value Stream Management? A Primer For Enterprise Leadership | Flint Brenton |
| What is Value Stream Management and Why It’s So Important | Scott White |
| What is Value Stream Mapping? | Kanbanize |
| What Value Stream Management Isn’t | Andrew Fuqua |
| Why Value Stream Management Focuses the Software Development Lifecycle | Jeff Keyes |
| Why Value Stream Management is a Must-Do Not a Nice-to-Do | Gaurav Rewari |
| Why Your Organization Needs a Value Stream Architect Today | Jeff Downs |

| Case Stories | |
|---|---|
| Organization | Articles |
| Adidas<br>MODULE 7 | - Adidas Annual Report 2019<br>- Adidas Annual Report 2020 |
| Bank of New Zealand<br>MODULE 2 | - Overcoming the Enterprise Great Disconnect (DevOps Institute) |
| Barclays | - State of DevOps in Banking (InfoQ) |
| Boeing | - Boeing: Connecting PMOs and Teams in Value Streams (Broadcom) |
| BMW | - Project to Product: How Value Stream Networks Will Transform IT & Business with Mik Kersten (IT Revolution)<br>- Connecting IT to the Business at BMW Group (IT Revolution) |
| Capital One<br>MODULE 1 | - SAFe Case Study: Capital One: Reimagining Product and Delivery through Agile (Scaled Agile) |
| Chipotle | - Chipotle: Using Value Streams to Manage Rapid Change<br>- Lessons from Chipotle's Winning Value Stream Management Approach (Broadcom) |
| Claro | - Micro Focus PPM and ALM Octane combine to secure continued market leadership with value-driven and customer-focused project delivery (Micro Focus) |
| DNB<br>MODULE 5 | - Leading Nordic bank DNB implemented DevOps development methodology with the help of ServiceNow automation |
| Ericsson | - Automation and Lean: Scaling up the Lean Value Chain (InfoQ) |
| FedEx | - Getting Business Results with FedEx DevSecOps Fast Lane (IT Revolution) |
| GSK | - Transforming from a Project-Centric to a Product-Centric Organization (GSK Tech) |
| HealthFirst<br>MODULE 6 | - How Value Stream Management Helped Healthfirst Overcome the Challenges in Changing Through the COVID19 Crisis (Plutora) |
| Hearst | - Value Stream Mapping: Streamline Processes and Release Faster (IT Revolution)<br>- Agile: After the Value Stream (IT Revolution) |

| | |
|---|---|
| Hershey's | ● Hershey's: Deliver Sweeter Outcomes With Value Streams (Broadcom) |
| State Revenue Service (Latvia) | ● Latvia's State Revenue Service Masters Software Delivery with Tasktop and AA Projekts (Tasktop) |
| Lloyds Banking Group | ● A Reflection on Scaling Agile (Robert Eriksson on LinkedIn) |
| Microsoft | ● How we used Value Stream Mapping to accelerate DevOps adoption (DevOps Barcelona) |
| Nationwide Insurance | ● DevOps Value Stream Mapping Process Plots Course at Nationwide (TechTarget) |
| Nike | ● Value Stream Mapping & Shared Outcomes - The Path To Collaboration & Winning As A Team (ADDO) |
| NIO<br>MODULE 4 | ● NIO Case Story (Tasktop) |
| Northrop Grumman<br>MODULE 3 | ● Adding Value to Value Stream Mapping (IISE)<br>● The Value Stream: Explained and Promoted by Northrop Grumman (APQC)<br>● Macro Value Stream Mapping at Northrop Grumman and Lockheed Martin (Lean Advisors)<br>● Expanding DevOps Principles to Large Complex Solutions - Northrop Grumman and Lockheed Martin (IT Revolution) |
| Schneider<br>MODULE 8 | ● Schneider delivers better value to the business with Digital.ai (Digital.ai) |
| Worldpay | ● Value Stream Integration (IT Revolution) |

| Videos, Podcasts, and Presentations | |
|---|---|
| Title/Link | Speaker(s) |
| 4 Maps of DevOps | Peter Maddison |
| 5 Keys of DevOps + Value Stream Assessment | Steve Pereira |
| 5 Reasons Value Stream Thinking is the Future of DevOps | Steve Pereira |
| Accelerating Value Flow: Moving Beyond Story Points | Jeff Keyes |
| Applied VSM Practices: The Flow of Value in Software Delivery | Neelan Choksi |
| Bryan Finster - Value Stream Architect | Don White and Bryan Finster |
| Creating a DevOps Toolchain | Brian Ashcraft |
| Dean Leffingwell on What is a Value Stream | Digital.ai |
| DevOps Agility Requires Shift in Focus to Product Value Streams | Mike Kavis and Mik Kersten |
| DevOps and the Enterprise Value Stream | Mike Kavis and Jayne Groll |
| DevOps Value Stream issues? Check for Time Thieves and Invisible Work | Mike Kavis and Dominica DeGrandis |
| How to Do Value Stream Mapping | Helen Beal |
| How to Implement Value Stream Management | Helen Beal |
| Humans are Hard, Code is Easy | Tom Henricksen |
| Integrated Toolchain Demonstration | Plutora |
| Making Connections Visible: How to Defrag Your Value Stream | Dominica DeGrandis |
| Managing the Process: Fundamentals in Lean Thinking | Tiffany Jachja |
| Next Generation DevOps: Value Stream Management | Helen Beal |
| Overcoming the Enterprise Great Disconnect | BMK Lakshminarayanan |
| Remove Bottlenecks and Optimize Workstreams with Value Stream Mapping | Mike Kavis and Steve Pereira |
| TOGAF Business Architecture: Value Stream | The Open Group |

| | |
|---|---|
| Guide | |
| Value Stream Management - A Retrospective | Hope Lynch |
| Value Stream Identification - Lessons from the Wild | Carl Starendal |
| Value Stream Management and the Next Decade of DevOps | Steve Boone and Chris Nowak |
| Value Stream Management: Friend or Foe? | Mark Rix |
| Value Stream Management: Inspecting Flow | Jonathan Harding |
| Value Stream Management: Metrics | Yaniv Sayers |
| Value Stream Management: Organizing Around Value | Richard Knaster |
| Value Stream Management versus Value Stream Mapping: What You Need to Know | Dominica DeGrandis and Carmen DeArdo |
| Value Stream Thinking | Steve Pereira |
| Value Trumps Flow When Managing Value Streams | Lance Knight |
| What is a Value Stream | Dean Leffingwell |
| What is a Value Stream Manager? | Nick Mathison |
| What is Value Stream Management | Eric Minick |
| ServiceNow DevOps Change Automation and Insights | ServiceNow |

| Research | | |
|---|---|---|
| Title/Link | Date | Analyst |
| DevOps 2021: Evaluating High Growth Markets in the DevOps Pipeline | April 29, 2020 | EMA |
| Elevate Agile-Plus-DevOps With Value Stream Management | February 19, 2020 | Forrester |
| Enable an Insights-Driven Business | April 29, 2021 | Forrester |
| GigaOm Radar for Value Stream Management | September 29, 2020 | GigaOm |
| Integrated Value Stream Management Benefits Business and Development Leaders | February 24, 2021 | Forrester |
| Market Guide for DevOps Value Stream Management Platforms | October 28, 2020 | Gartner |
| Market Insight: Value Stream Management Vendors' Ability to Provide Insights and Visibility in Value Streams | May 19, 2021 | Quadrant Knowledge Solutions |
| Predicts 2021: Value Streams Will Define the Future of DevOps | October 5, 2020 | Gartner |
| State of Agile Reports | Annual | Digital.ai |
| State of DevOps Reports (Puppet) | Annual | Puppet Labs |
| State of DevOps Reports (Google Cloud) | Annual | Google Cloud |
| State of Value Stream Management Reports (VSMC) | Annual | Value Stream Management Consortium |
| The Forrester New Wave™: Value Stream Management Tools, Q3 2018 | August 6, 2018 | Forrester |
| The Forrester Wave™: Value Stream Management Solutions, Q3 2020 | July 15, 2020 | Forrester |
| The Future of DevOps Toolchains Will Involve Maximizing Flow in IT Value Streams | January 14, 2020 | Gartner |
| The Value Stream Management Market: A Customer-Driven Evaluation of Products | October 29, 2020 | EMA |
| Upskilling Reports | Annual | DevOps Institute |
| Value Stream Management | January, 1998 | The International |

| | | |
|---|---|---|
| | | Journal of Logistics Management |
| Vendor Selection Matrix™ – Artificial Intelligence Predictive Analytics | | Research in Action |
| Vendor Selection Matrix$^{TM}$ Value Stream Management | July, 2020 | Research in Action |

| Books | |
|---|---|
| **Title/Link** | **Author(s)** |
| Accelerate | Dr. Nicole Forsgren, Jez Humble, Gene Kim |
| Beyond the Phoenix Project | John Willis, Gene Kim |
| Driving DevOps with Value Stream Management | Gary Rupp |
| Lean Software Development | Mary and Tom Poppendieck |
| Lean Thinking: Banish Waste And Create Wealth In Your Corporation | James P Womack, Daniel T Jones |
| Learning to See | Mike Rother, John Shook |
| Making Work Visible | Dominica DeGrandis |
| Measure What Matters | John Doerr |
| Outcomes Over Output | Joshua Seiden |
| Project to Product | Dr. Mik Kersten |
| Seeing the Whole Value Stream | James P Womack, Daniel T Jones |
| The Art of Business Value | Mark Schwartz |
| The Art of Thinking Like a Data Scientist | Bill Schmarzo |
| The DevOps Handbook | Gene Kim, John Willis, Jez Humble, Patrick Debois |
| The Fifth Discipline | Peter Senge |
| The Great Transition | James Martin |
| The Machine that Changed the World | James P Womack, Daniel T Jones, Daniel Roos |
| Thinking in Systems | Donella H. Meadows |
| The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer | Jeffrey Liker |
| Value Stream Mapping | Karen Martin |
| Value Stream Management: Eight Steps to Planning, Mapping, and Sustaining Lean Improvements | Don Tapping, Tom Luyster, Tom Shuker |
| Value Stream Management. Strategy and Excellence in the Supply Chain | Peter Hines |

| | |
|---|---|
| Value Stream Management For Lean Healthcare | Don Tapping, Sue Kozlowski, Laura Archbold, and Todd Sper |
| Value Stream Management for the Lean Office | Don Tapping, Tom Shuker |
| Making Work Visible | Dominica DeGrandis |
| Standing on Shoulders: A Leader's Guide to Digital Transformation | Jack Maher and Carmen DeArdo |

| Websites | |
|---|---|
| Title/Link | Description |
| Broadcom | VSM platform vendor |
| CloudBees | VSM platform vendor |
| Digital.ai | VSM platform vendor |
| Flow Framework | The Flow Framework® from Tasktop provides the blueprint for implementing Value Stream Management, connecting IT and the business, and transforming your organization into a high-performing tech company. |
| Gemba Academy | Training |
| GitLab | VSM platform vendor |
| HCL Software | VSM platform vendor |
| ISO | |
| IT4IT | |
| TKMG | Value Stream Consulting, Training |
| Micro Focus | VSM platform vendor |
| Plutora | VSM platform vendor |
| PMI | |
| Poppendieck | Value Stream Consulting, Training |
| SAFe | Agile at scale framework |
| Sealights | |
| ServiceNow | VSM platform vendor |
| Target Process | VSM platform vendor |

| | |
|---|---|
| Tasktop | VSM platform vendor |
| Team Topolologies | Organizational framework to optimize flow |
| The Flow Collective | A value stream peer community |
| valuestream.link | Value stream resource directory |
| Visible Value Stream Consulting | Value Stream Consulting, Training |
| What Matters | John Doerr and OKRs |

# Value Stream Management Foundation v1.0

Sample Examination 1

1. Which BEST represents the goals of Value Stream Management?
    a. End-to-end visibility and control
    b. Deciding as late as possible
    c. Implementing change and DevOps
    d. Basing decisions on a long-term philosophy

2. Which of the following is NOT a typical silo challenge?
    a. Finger-pointing
    b. Few handoffs
    c. Rework
    d. Late-night pushes

3. Who is the primary focus for value creation in a value stream?
    a. Business stakeholders
    b. Customer
    c. Shareholders
    d. All of the above

4. From which industry did value stream mapping techniques emerge?
    a. Manufacturing
    b. Finance
    c. Healthcare
    d. Information technology

5. Which team topology is most concerned with the continuous flow of value?
    a. Stream-aligned
    b. Platform
    c. Enabling
    d. Complicated-subsystem

6. Which of the following is NOT a component of a value-based digital transformation?
    a. Outcome-based strategy
    b. Lightweight planning and governance
    c. Value-based prioritization
    d. Annual budgeting and project funding

7. Which term refers to the period from when an idea or opportunity is first revealed, to when a team is assembled to start work on realizing the idea?
    a. Event storming
    b. Customer journey map
    c. Fuzzy front end
    d. Value stream map

8. Which metric is the BEST example of a FLOW metric?
    a. Cycle time
    b. Conversion rate
    c. Value stream ROI
    d. Customer retention

9. What are the Three Pillars of Empiricism?
    a. Flow, feedback, and experimentation
    b. Value, outputs, and outcomes
    c. Transparency, inspection, and adaptation
    d. Mapping, measurement, and metrics

10. Which is the BEST way to describe what value stream management seeks to accelerate?
    a. Time to market
    b. Time to value
    c. Time to learning
    d. Time to customer

11. Which of the following is one of the 14 Management Principles from the Toyota Way?
    a. Leaders make decisions fast, considering the top three options; implement decisions slowly.
    b. Move toward flow; move to ever-smaller batch sizes and cycle times to deliver value fast and to expose weakness and hidden problems.
    c. Use push systems to avoid overproduction of WIP or inventory; decide as early as possible.
    d. Base management decisions on a short-term philosophy, even at the expense of long-term financial goals.

12. Which of the following BEST describes where a value stream map starts and stops?
    a. From process start to process stop
    b. From engagement to relationship
    c. From desired outcome to experiment
    d. From idea to value realization

13. Which of the following best describes the value stream management approach over the traditional way of working?
    a. Systems are tightly coupled and monolithic
    b. Teams are dependent on each other
    c. Most conversations are opinion-driven
    d. Small, multifunctional, autonomous teams are dedicated to long-lived products

14. How can we best describe how an organization operates as an organism?
    a. Organic, adaptive
    b. Hierarchical, static
    c. Siloed, layered
    d. Pyramidical, reporting lines

15. Which of the following is NOT a function of a Value Stream Management Platform (VSMP)?
    a. Visualization of value streams and maps
    b. Governance and continuous compliance
    c. Project tracking and management
    d. Workflow orchestration

16. How does a Value Stream Management Platform (VSMP) collate data from the DevOps toolchain?
    a. By dashboard
    b. Via a common data model
    c. Through a data lake
    d. Point to point integrations

17. Which best describes Little's Law ($L = \lambda * W$)?
    a. ROI = Flow Time * ACR
    b. WIP = ACR * Lead Time
    c. Velocity = Flow Time * ACR
    d. WIP = Flow Time * ACR

18. Which of the following is listed in the 'Business Metrics' category in Gartner's Flow Metrics?
    a. Code change size
    b. Throughput
    c. Product value
    d. Work profile

19. Which is an example of a qualitative (not a quantitative) measure?
    a. Volatility
    b. Wait time
    c. Quality
    d. Value outcomes

20. Which is NOT a step in the VSM Implementation Roadmap?
    a. Analyze
    b. Adapt
    c. Inspect
    d. Map

21. What is the benefit of **measuring** the value stream?
    a. Increases in delivery speed and capacity
    b. The demonstration of business outcomes and optimization
    c. Improved communication and prioritization
    d. Traceability and data collection

22. According to 'The Great Transition', who searches for ways of achieving "outrageous" improvements in critical measures such as speed, cost, quality, and service?
    a. The value stream team
    b. Value stream designers
    c. The value stream architect
    d. The value stream owner

23. What is the definition of ‘touch time’ in value stream mapping?
    a. The elapsed time an activity is worked on from start to finish
    b. How quickly you can deliver work
    c. The required product assembly duration that is needed to match the demand
    d. The elapsed time between one activity finishing and another starting

24. Which of these characteristics are more aligned to value stream mapping than value stream management?
    a. Automates for continuous compliance
    b. Insights and data-driven, quantitative
    c. Ongoing incremental optimization
    d. Human-centric, storytelling, and narrative

25. Which roles are part of a value stream team?
    a. Product managers and owners
    b. Service desk
    c. Customer success
    d. All of the above

26. Which of the following describes an organization at the ‘evolving’ capability level of value stream management for the DevOps toolchain dimension?
    a. Tools have been integrated
    b. Breaking dependencies
    c. Work is traceable around entire cycle - automated value stream map
    d. Using continuous delivery

27. Which best describes The First Way of DevOps?
    a. Emphasizes the performance of the entire system
    b. Creates the right to left feedback loops
    c. A culture that fosters continual experimentation and learning from failure
    d. Understanding that repetition and practice is the prerequisite to mastery

28. In a digital or software-focused value stream, how much time is typically spent on design and development?
    a. 10%
    b. 25%
    c. 50%
    d. 90%

29. What should you aim to eliminate from a current state value stream map?
    a. All meetings
    b. Automated handoffs
    c. Toil
    d. Value-adding activities

30. For application delivery, what do Value Stream Management Platforms (VSMPs) serve as, in Gartner's definitions?
    a. The execution plane
    b. The control plane
    c. The data plane
    d. The user plane

31. The Hypothesis Prioritization Canvas tells us we should do what with hypotheses that are both high risk and of high perceived value?
    a. Ship
    b. Test
    c. Measure
    d. Discard

32. Which statement describes how 'overproduction' waste manifests in technology value streams?
    a. Time spent fixing bugs and paying down technical debt
    b. When capacity is spare in teams and they are starved of requirements
    c. Teams are siloed and handoffs cause delays
    d. People are not empowered to decide on their work and make decisions

33. Which describes the 'Insights and Analysis' section of the value stream cycle?
    a. The changes are approved, released, and operated in the live environment
    b. Code is created, artifacts incorporated, versions controlled, code is built in a trunk-based manner
    c. Vision and goals are set and aligned to epics, features, PBIs, and user stories
    d. Monitoring and observability provide insights into customer reaction to changes and report on value realization

34. Which is a leading indicator for value realization?
    a. Revenue
    b. Net Promoter Score (NPS)
    c. Journey time
    d. Customer lifetime value

35. Which are examples of key results for business?
    a. Usage, feedback, NPS/referrals, reviews, bounce rate, session length
    b. Income, % adoption, churn, conversion
    c. Days/hours/minutes, efficiency, WIP, distribution
    d. WIP, work outcome, system capacity, uptime, MTTD/R, MTBF

36. Which is a TECHNICAL trigger to inspect value realization?
    a. Not having data telling us how our customers are responding to what we deliver
    b. Acting like a "feature factory"
    c. Organizational performance is not great
    d. Difficult to prioritize new activities

37. Which value stream role is a 'champion' (rather than a coordinator or contributor)?
    a. Value Stream Architect (VSA)
    b. Value Stream Manager (VSM)
    c. Value Stream Engineer (VSE)
    d. Chief Value Officer (CVO)

38. Which is NOT a Flow Item in the Flow Framework?
    a. Quality
    b. Defects
    c. Debts
    d. Features

39. Which is an anti-pattern when transitioning into value stream roles?
    a. Reimagine roles from outcomes backwards
    b. Measurement before method
    c. Large scale, sudden rollout
    d. Map the role in relation to the existing structure

40. When should flow be inspected?
    a. Continuously
    b. Every sprint
    c. Monthly
    d. Intermittently

# VSM Foundation v1.0 Sample Exam 1 Answer Key

| Question | Answer | Topic |
|---|---|---|
| 1 | A | VSMF-1 HISTORY AND EVOLUTION |
| 2 | B | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 3 | D | VSMF-2 IDENTIFYING VALUE STREAMS |
| 4 | A | VSMF-3 MAPPING VALUE STREAMS |
| 5 | A | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 6 | D | VSMF-8 EVOLVING VALUE STREAMS |
| 7 | C | VSMF-3 MAPPING VALUE STREAMS |
| 8 | A | VSMF-5 VALUE STREAM METRICS |
| 9 | C | VSMF-6 INSPECTING THE VALUE STREAM |
| 10 | C | VSMF-4 CONNECTING DEVOPS TOOLCHAINS |
| 11 | B | VSMF-1 HISTORY AND EVOLUTION |
| 12 | D | VSMF-3 MAPPING VALUE STREAMS |
| 13 | D | VSMF-8 EVOLVING VALUE STREAMS |
| 14 | A | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 15 | C | VSMF-4 CONNECTING DEVOPS TOOLCHAINS |
| 16 | B | VSMF-4 CONNECTING DEVOPS TOOLCHAINS |
| 17 | D | VSMF-5 VALUE STREAM METRICS |
| 18 | C | VSMF-5 VALUE STREAM METRICS |
| 19 | A | VSMF-6 INSPECTING THE VALUE STREAM |
| 20 | A | VSMF-2 IDENTIFYING VALUE STREAMS |
| 21 | B | VSMF-1 HISTORY AND EVOLUTION |
| 22 | B | VSMF-2 IDENTIFYING VALUE STREAMS |
| 23 | A | VSMF-3 MAPPING VALUE STREAMS |
| 24 | D | VSMF-3 MAPPING VALUE STREAMS |
| 25 | D | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 26 | C | VSMF-8 EVOLVING VALUE STREAMS |
| 27 | A | VSMF-1 HISTORY AND EVOLUTION |
| 28 | D | VSMF-2 IDENTIFYING VALUE STREAMS |
| 29 | C | VSMF-3 MAPPING VALUE STREAMS |
| 30 | B | VSMF-4 CONNECTING DEVOPS TOOLCHAINS |
| 31 | B | VSMF-8 EVOLVING VALUE STREAMS |
| 32 | B | VSMF-5 VALUE STREAM METRICS |
| 33 | D | VSMF-4 CONNECTING DEVOPS TOOLCHAINS |
| 34 | C | VSMF-6 INSPECTING THE VALUE STREAM |
| 35 | B | VSMF-8 EVOLVING VALUE STREAMS |
| 36 | A | VSMF-6 INSPECTING THE VALUE STREAM |
| 37 | D | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 38 | A | VSMF-5 VALUE STREAM METRICS |

|  |  |  |
|---|---|---|
| 39 | C | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 40 | A | VSMF-6 INSPECTING THE VALUE STREAM |

vsm consortium

DevOps Institute

# Value Stream Management Foundation v1.0

## Sample Examination 2

1. What is the purpose of the other (not stream-aligned) fundamental team types in Team Topologies?
    a. To not require hand-offs
    b. To be empowered to deliver customer value
    c. To reduce the burden on the stream-aligned teams
    d. To exist alone and apart

2. A customer journey map is primarily concerned with the flow of…?
    a. Experience
    b. Value
    c. Work
    d. Materials

3. What is value realization?
    a. The journey of work
    b. Friction-free travel
    c. A continuous steady stream
    d. The fulfillment of desired outcomes

4. In McKinsey's Next-Generation Target Operating Model, what does 'Advanced Analytics' intend to achieve?
    a. Provide intelligence to facilitate decisions
    b. Streamline processes and minimize waste
    c. Digitize customer experience and day-to-day operations
    d. Use intelligent automation to replace human tasks

5. Which is an example of a 'surface' or 'manifest' aspect of the VSM Culture Iceberg?
    a. Behavior
    b. Mindset
    c. Psychological safety
    d. Beliefs

6. Which is one of the 14 Management Principles from The Toyota Way?
    a. Today’s problems come from yesterday’s ”solutions”
    b. Small changes can produce big results – areas of highest leverage are often the least obvious
    c. There is no blame
    d. Become and sustain a learning organization through relentless reflection and kaizen

7. ‘Identify’ precedes which step in the VSM Implementation Roadmap?
    a. Map
    b. Connect
    c. Adapt
    d. Organize

8. In the VSM Capability Matrix, what describes the dimension ‘Dependencies’ at the Learning level?
    a. Aware
    b. Managing
    c. Breaking
    d. Loosely-coupled

9. What is a more advanced metric than time to value?
    a. Time to market
    b. Cycle time
    c. Time to learning
    d. Delivery time

10. Which pillar of empiricism says that it is ok to change direction?
    a. Transparency
    b. Inspection
    c. Adaptation
    d. All of the above

11. What is different about Value Stream Management in the digital world?
    a. Nothing
    b. People are less important now than tools
    c. Flow can be tracked through software delivery pipelines
    d. Value stream maps are no longer needed

12. Which statement describes value stream management rather than value stream mapping?
    a. Opinion-driven, qualitative
    b. Visual collaboration
    c. Human-centric, storytelling, and narrative
    d. Automates for continuous compliance

13. Which of the following value stream roles is classed as a 'Contributor'?
    a. Chief Value Officer
    b. Value Stream Manager
    c. Value Stream Architect
    d. Flow Engineer

14. Which is a good pattern for transitioning into value stream roles?
    a. Measurement before method
    b. Reimagine roles from outcomes forwards
    c. Renaming existing roles
    d. Large scale, rapid rollout

15. When people are not empowered to decide on their work and make decisions, this is an example of which waste type manifestation in digital value streams?
    a. Defects
    b. Overproduction
    c. Waiting
    d. Non-utilized talent

16. Which use case suggests that AIOps could be renamed AIPA (Artificial Intelligence Predictive Analytics)?
    a. IT Operations
    b. IT Service Management
    c. Customer experience
    d. Continuous deployment

17. Charlie is CIO of a global consumer packaged conglomerate. The CEO has set a goal to deliver fast and to expose weaknesses and hidden problems in the organizations' digital value streams. Which of the 14 Management Principles from The Toyota Way should Charlie look to for guidance?
    a. Use pull systems to avoid overproduction of WIP
    b. Become and sustain a learning organization
    c. Move toward flow; move to ever-smaller batch sizes and cycle times
    d. Level the work - reduce variability and overburden

18. Practicing visualizing the value stream results in illustration of the value stream. What's the benefit of this practice?
    a. Improved communication and prioritization
    b. Increased speed of delivery
    c. Demonstration of business outcomes
    d. Optimization

19. Which activities in the value stream concern the value stream team?
    a. The start
    b. The delivery of the results
    c. Confirmation of satisfaction
    d. All the activities

20. Value flow metrics are outputs that also indicate...?
    a. Value stream health
    b. Customer experience
    c. Return on investment
    d. Conversion rates

21. Which BEST describes one of the purposes of value stream mapping?
    a. Diagramming of workflow
    b. Visual representation of customer experience
    c. Visual collaboration
    d. Domain discovery and modelling

22. Which of the following describes a VSM use case for development teams?
    a. Get instant feedback on their day-to-day processes to identify and understand bottlenecks as they occur
    b. Accurately communicate to the rest of the business and customers about delivery dates to better manage expectations
    c. Shift governance concerns left to identify risk, minimize rework, and improve quality
    d. Ensure that engineering activities are in alignment with business goals

23. Which of the following best describes work in the 'Design and validate feature' side of product development value flow in digital value streams?
    a. Defined, automated steps
    b. Funding may not be secure
    c. Costs understood
    d. Has been prioritized in the work queue

24. Where did CICD initially focus in the value stream?
    a. At the start
    b. In the middle
    c. At the end
    d. End-to-end

25. Where do all value streams start?
    a. Ideation
    b. Development
    c. Delivery
    d. Realization

26. When waste is removed from a value stream, it results in...?
    a. More time for innovation
    b. More context switching
    c. More time waiting
    d. Less time for learning

27. Which statement describes ‘wait time’ in the context of value stream mapping?
    a. The period from when an idea or opportunity is first revealed, to when a team is assembled to start work on realizing the idea
    b. The required product assembly duration that is needed to match the demand
    c. The elapsed time an activity is worked on from start to finish
    d. The elapsed time between one activity finishing and another starting

28. The statement “The changes are approved, released and operated in the live environment” describes which step in the value cycle?
    a. Portfolio and backlog
    b. Continuous integration
    c. Continuous delivery
    d. Continuous testing

29. In which stage of the value cycle are the logging and monitoring, observability and AIOps, and analytics and dashboard parts of the DevOps toolchain?
    a. Portfolio and backlog
    b. Continuous delivery
    c. Insights and analysis
    d. Continuous integration

30. In the Flow Framework, which of the following is a business result?
    a. Flow velocity
    b. Happiness
    c. Debts
    d. Features

31. Which of the following is an example of a qualitative metric?
    a. Variability
    b. Value outcomes
    c. Information flow
    d. Value-added time

32. Which of the following is NOT a good practice for flow inspection?
    a. Do it continuously
    b. Automate the map using a Value Stream Management Platform (VSMP)
    c. Compare improvements with hypothesis backlog
    d. Revisit ideal state annually

33. An organization is in a state where some value streams are experimenting with DevOps practices and tooling. They are making local discoveries and are aiming to make these global improvements. Their transformation lead has started value stream mapping. At what evolutionary stage is this organization?
    a. Starting
    b. Piloting
    c. Scaling
    d. Optimizing

34. Gartner identified the key capabilities for Value Stream Delivery Platforms (VSDPs). Among them are APIs. What is their purpose?
    a. A basis for security and compliance
    b. Driving team collaboration
    c. Providing extensibility
    d. Governance and access

35. How does value stream inspection using a Value Stream Management Platform (VSMP), help teams during the Scrum event Sprint Review?
    a. Setting the Sprint Goal may involve reviewing team performance and OKRs. Seeing how factors such as cycle time and change fail rate are trending over time helps prioritize features
    b. The team gets together daily to review progress and impediments - VSMP provides real-time data on delays in the value stream helping to guide the ScrumMaster and provide evidence to help remove impediments
    c. VSMPs show how the team performed against their Sprint Goal and whether they are meeting their target cycle times. It can also provide insights into value realized
    d. Having real and immediate data that shows where PBIs may have been stuck provides insights into delays. This tells the teams where they may need to adjust their processes to improve

36. Which statement describes funding in value stream management?
    a. Fixed scope projects, long-term planning and rigid budget
    b. Centralized decision making
    c. Functional silos and project setups guide budgeting
    d. Decentralised decision making ensures buy-in of all relevant parties

37. What type of value stream roles are a natural fit for product owners and agile coaches to evolve to?
    a. Contributor roles like value stream or flow engineers
    b. Coordinator roles such a value stream architect or manager
    c. Champion roles such as chief value officer
    d. There are no roles for these people

38. Which is NOT a principle for insights-driven business?
    a. Learning and experimentation are continuous
    b. Data collection and implementation conducted by specialists
    c. Data is pervasive and anticipatory, delivered at the edge
    d. Data leads to insights, to actions, to outcomes

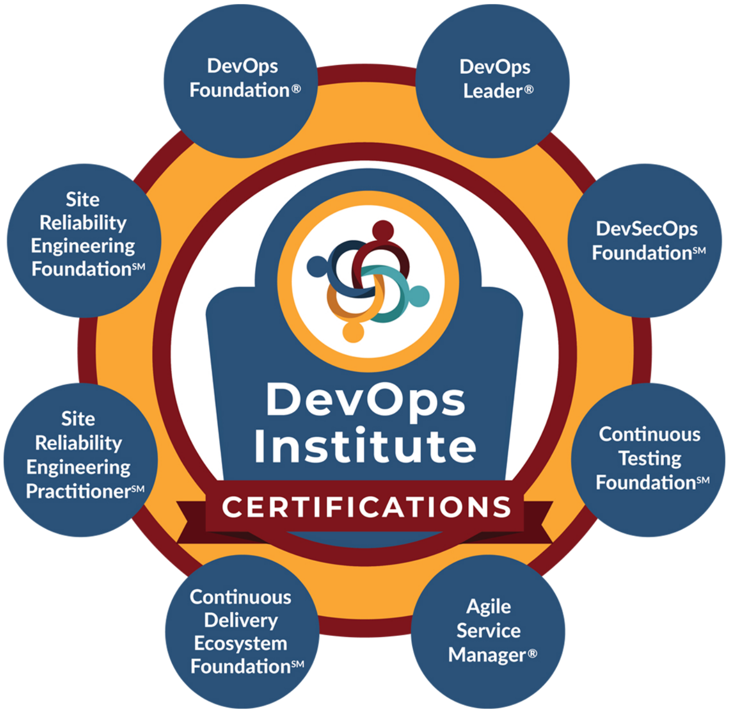
39. The CTO of a bank has asked her team leaders to learn to practice an agile software development process that encourages collaboration among developers, quality assurance testers, and customer representatives in a digital value stream. She is hoping to encourage teams to use conversation and concrete examples to formalize a shared understanding of how their application should behave. What is it she is asking them to master?
    a. Test-driven development (TDD)
    b. Behavior-driven development (BDD)
    c. Hypothesis-driven development (HDD)
    d. Impact-driven development (IDD)

40. A value stream team is aggregating data, managing dependencies, using continuous delivery, measuring via deployment frequency and MTTR metrics and naming some value stream roles. At what stage of the VSM Capability Matrix are they?
    a. Emerging
    b. Learning
    c. Practicing
    d. Evolving

# VSM Foundation v1.0 Sample Exam 2 Answer Key

| Question | Answer | Topic |
|---|---|---|
| 1 | C | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 2 | A | VSMF-3 MAPPING VALUE STREAMS |
| 3 | D | VSMF-5 VALUE STREAM METRICS |
| 4 | A | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 5 | A | VSMF-8 EVOLVING VALUE STREAMS |
| 6 | D | VSMF-1 HISTORY AND EVOLUTION |
| 7 | D | VSMF-2 IDENTIFYING VALUE STREAMS |
| 8 | B | VSMF-8 EVOLVING VALUE STREAMS |
| 9 | C | VSMF-4 CONNECTING DEVOPS TOOLCHAINS |
| 10 | C | VSMF-6 INSPECTING THE VALUE STREAM |
| 11 | C | VSMF-1 HISTORY AND EVOLUTION |
| 12 | D | VSMF-3 MAPPING VALUE STREAMS |
| 13 | D | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 14 | A | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 15 | D | VSMF-5 VALUE STREAM METRICS |
| 16 | C | VSMF-6 INSPECTING THE VALUE STREAM |
| 17 | C | VSMF-1 HISTORY AND EVOLUTION |
| 18 | A | VSMF-1 HISTORY AND EVOLUTION |
| 19 | D | VSMF-2 IDENTIFYING VALUE STREAMS |
| 20 | A | VSMF-5 VALUE STREAM METRICS |
| 21 | C | VSMF-3 MAPPING VALUE STREAMS |
| 22 | A | VSMF-5 VALUE STREAM METRICS |
| 23 | B | VSMF-3 MAPPING VALUE STREAMS |
| 24 | B | VSMF-4 CONNECTING DEVOPS TOOLCHAINS |
| 25 | A | VSMF-2 IDENTIFYING VALUE STREAMS |
| 26 | A | VSMF-2 IDENTIFYING VALUE STREAMS |
| 27 | D | VSMF-3 MAPPING VALUE STREAMS |
| 28 | C | VSMF-4 CONNECTING DEVOPS TOOLCHAINS |
| 29 | C | VSMF-4 CONNECTING DEVOPS TOOLCHAINS |
| 30 | B | VSMF-5 VALUE STREAM METRICS |
| 31 | A | VSMF-6 INSPECTING THE VALUE STREAM |
| 32 | D | VSMF-6 INSPECTING THE VALUE STREAM |
| 33 | B | VSMF-3 MAPPING VALUE STREAMS |
| 34 | C | VSMF-4 CONNECTING DEVOPS TOOLCHAINS |
| 35 | C | VSMF-6 INSPECTING THE VALUE STREAM |
| 36 | D | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 37 | B | VSMF-7 ORGANIZING AS VALUE STREAMS |
| 38 | B | VSMF-8 EVOLVING VALUE STREAMS |

|  |  |  |
|---|---|---|
| 39 | B | VSMF-8 EVOLVING VALUE STREAMS |
| 40 | B | VSMF-8 EVOLVING VALUE STREAMS |


DevOps Foundation®
DevOps Leader®
Site Reliability Engineering Foundation℠
DevSecOps Foundation℠
DevOps Institute
CERTIFICATIONS
Site Reliability Engineering Practitioner℠
Continuous Testing Foundation℠
Continuous Delivery Ecosystem Foundation℠
Agile Service Manager®

## Your Path to
# DevOps Success

DevOps Institute is dedicated to advancing the human elements of DevOps success. Our goal is to help advance careers and support emerging practices using a role-based approach to certification which focuses on the most modern competencies and hireable skills required by today's organizations adopting DevOps.

Take the next steps in your learning and certification journey to DevOps success.

Click on a certification or visit www.devopsinstitute.com/certifications to learn more.

### Become a Member

Join the fastest growing global community of DevOps practitioners and professionals and gain access to invaluable learning content, the latest news, events, emerging practices, develop your network and advance your career.

## You belong.

( devopsinstitute.com/membership )