DevOps Definitions of Terms

There is no official "standard" or comprehensive definition of DevOps terms. The DevOps Institute, IEEE, ITIL, ISO and Wikipedia are good sources for some definitions however definitions tend to be different for different sources. This document provides an alphabetized list of popular DevOps terms, abbreviations and acronyms, with a description. The items in this list are used as nouns, unless otherwise indicated.

12-Factor Apps - A methodology for building modern, scalable, maintainable software-as-a-service apps. https://12factor.net/
A/B testing - Deploy different versions of an EUT to different customers and let the customer feedback determine which is best.
Administration testing - The purpose of the test is to determine if an EUT is able to process administration tasks as expected.
Agile Manifesto - A formal proclamation of values and principles to guide an iterative and people centric approach to software development.
Agile Software Development (Agile) - Group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.
Analytics - Results processed and presented in an organized manner in accordance with analysis methods and criterion.
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.
API – see Application Programming Interface
API testing - The purpose of the test is to determine if an API for an EUT functions as expected.
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 Release Automation (ARA) - the process of packaging and deploying an application or update across the complete release lifecycle, ultimately enabling new products and services to be brought to market faster.
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
ARA – see Application Release Automation
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 for each release candidate. E.g. Archiva, Nexus, JFrog
ATDD – see Application Test Driven Development
AUT - see Application Under Test
Auto-scaling - The ability to automatically and elastically scale and de-scale infrastructure depending on traffic and capacity variations while maintaining control of costs.

Backlog - Requirements for a system, expressed as a prioritized list of product backlog items. The product backlog is prioritized by the Product Owner and includes functional, non-functional and technical team-generated requirements.
Behavior-Based Testing - Test cases are created by simulating an EUT's externally observable inputs, and outputs. Example tool: Cucumber
Black-Box Testing - Test case only uses knowledge of externally observable behaviors of an EUT.
Blue/Green testing - Taking software from the final stage of testing to live production using two environments labelled 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.
Bursting - Public cloud resources are added as needed to temporarily increase the total computing capacity of a private cloud.
CAB - Change Advisory Board
Cadence – An indication of the frequency or rhythm of software process cycles or releases.
CALMS - Pillars or values of DevOps: Culture, Automation, Lean, Measurement, Sharing.
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.
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. Eg. Selenium
Cause Analysis (RCA) - Actions take to identify the underlying cause of a problem or incident.
CD – see Continuous Delivery and Continuous Deployment
Change - Addition, modification or removal of anything that could influence IT services.
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 change to delivery of the change.
Change Management - Process that controls all software changes throughout the software lifecycle.
Change Request – A ticket or other form of requirement asking to change something.
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.
Chat-Ops - An approach to managing technical and business operations through a group chat room. E.g. Slack
Chat-Ops tool - The tool user invokes tests from a chat script. Example: Scriptrock
Check-in - Action of submitting a software change into a system version management system.
CI – see Continuous Integration
CI Regression Test - A subset of regression tests that are run immediately after a software component is built. Same as Smoke Test.
CI/CD – see Continuous Integration, Continuous Delivery and Continuous Deployment
Clear-Box Testing - Same as Glass-Box Testing and White-Box Testing.
Cloud-native - Native cloud applications (NCA) are designed for cloud computing.
Clustering - A group of computers (called nodes or members) work together as a cluster connected through a fast network acting as a single system.
CM – see Configuration Management and Continuous Monitoring

CMDB - Configuration Management Database
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 review - Software engineers inspect each other's source code to detect coding or code formatting errors.
Collaboration - People jointly working with others towards a common goal.
Compatibility Test - Test with the purpose to determine if and EUT interoperates with another EUT such as peer-to-peer applications or protocols.
Configuration Management (CM) - is a system 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 to a standard.
Constraint - Limitation or restriction; something that constrains. See also bottleneck.
Containers - Containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries – anything you can install on a server. This guarantees that it will always run the same, regardless of the environment it is running in. Example: Docker, Valgrant
Continuous Delivery (CD) - A software development discipline where you build software in such a way that the software can be released to production at any time.
Continuous Delivery Architect - A person who is responsible to guide the implementation and best practices for a continuous delivery pipeline.
Continuous Delivery Pipeline - A series of processes which 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 – Process step 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 created changes into a version of a product; Building: produce a version of a product subsystem ; Binding: produce a version of the product system into package artifacts; Delivery: produce a release candidate version of artifacts for deployment; Deployment - release and distribute a version of a product to production
Continuous Deployment (CD) – A process that takes validated software from a delivery staging environment and releases it to a production environment.
Continuous Flow - Smoothly moving software changes from the first step of a process to the last with minimal interruptions between steps.
Continuous Integration (CI) - A development practice that requires developers to merge their code into a common shared repository — ideally, multiple times per day.
Continuous Monitoring (CM) - This is a class of terms relevant to logging, notifications, alerts, displays and analysis of test results information.
Continuous Security – see DevSecOps and Rugged DevOps
Continuous Testing (CT) - This is a class of terms relevant to testing and verification of an EUT in a DevOps environment.

Conway's Law – An observation by Melvin Conway in 1967 that organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.
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.
CS – see Continuous Security
CT – see Continuous Testing
CSF – see Critical Success Factor
Culture - The values and behaviors that contribute to the unique social and psychological environment of an organization."
Cycle Time - A measure of the time to complete one pass of a repeating work task. E.g. build cycle.
Dashboard - Graphical display of summarized test results.
Defect density - The number of faults found in a unit E.g. # defects per KLOC, # defects per change.
Definition of Done (DoD) - Shared understanding of what it means for work to be complete.
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).
Deployment - Distribute and activate new versions of a software for use. For example, the installation of a specified version of software to a given environment such as promoting a new build into production.
Deployment Pipeline - flow of software changes into production via an automated software production line.
Design for Testability - An EUT is designed with features which enable it to be tested.
Developers (Dev) - Individuals involved in software development activities such as application and software engineers.
Development test environment - Ensuring that the developer's test environment is a good representation of the production test environment.
Device Under Test (DUT) - The EUT is a physical device. E.g. Router or switch is being tested.
DevOps - The application of the principles of continuous flow, feedback and improvement to people, process and technology for increasing agility, stability, security, efficiency, quality, availability and satisfaction.
DevOps Infrastructure - The entire set of tools and facilities that make up the DevOps system. Includes CI, CT, CM, CS and CD tools and integrations with orchestration and automation tools as needed to process all changes and prepare delivery packages.
DevOps Pipeline - Same as continuous delivery pipeline.
DevOps Toolchain - a set of software tools that are linked (or chained) together to form a DevOps CI/CD pipeline.
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.
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).
DIY - Do-It-Yourself solution. Same as home-grown solution.

DoD – see Definition of Done
Downtime – same as Mean Time to Restore Service (MTRS)
DUT – see Device Under Test
DVCS – see Distributed Version Control System
Dynamic analysis - The purpose of the test is to determine the performance characteristics of an EUT such as timing of specific paths in the EUT's code. Usually conducted on specific modules, not an entire system, because most of the tools require the code to be instrumented and can intrusively affect the system performance.
Entity Under Test (EUT) – Anything that is being tested. These terms are often abbreviated to the form xUT where "x" represents a type of entity under test. These objects may be physically implemented in hardware (DUT) or may exist as virtual software or services (SUT) that are made up from a combination of systems. An EUT may be a stand-alone independent module or a system of multiple interdependent modules.
ETL - In computing, extract, transform, load is the general procedure of copying data from one or more sources into a destination system which represents the data differently from the source.
EUT – see Entity Under Test
FaaS – see Function-as-a-Service
Fail Early - A CT tenet referring to the preference to find critical problems as early as possible in a DevOps pipeline.
Fail Often - A CT tenet which emphasizes a preference to find critical problems as fast as possible.
Failure rate - # fail verdicts per unit time.
False negative - A test incorrectly reports a verdict of "fail" when the EUT passed the purpose of the test.
False positive - A test incorrectly reports a verdict of "pass" when the EUT 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.
Flow – Software changes moving through a series of process steps.
Framework - Backbone for plugging-in tools, to integrate tool-chains. Launches automated tasks, collects results from automated tasks.
Function-as-a-Service (FaaS) - Category of cloud computing **services** that provides a platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app.
Gated Commits – Assessment criterion for changes to be promoted between all CD pipeline stages such as: Dev to CI stage, CI to packaging or delivery stage, and Delivery to Deployment or Production stage.
Glass-Box Testing - 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 incremental stresses until the EUT reaches a 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 “why we do something” first, then “how we do it” and then “what we do”.
Governance - Establishment of policies, and continuous monitoring of their proper implementation, by the members of a governing body. It includes the mechanisms required to balance the powers of the members (with the associated accountability), and their primary duty of enhancing the prosperity and viability of the organization.

Graphical User Interface (GUI) - pronounced "gooey." It is a user interface that includes graphical elements, such as windows, icons and buttons. The term was created in the 1970s to distinguish graphical interfaces from text-based ones, such as command line interfaces.
Gray-Box Testing - Test cases use a limited knowledge of the internal design structure of the EUT.
Green-Blue Deployment - see Blue-Green Deployments
GUI – see Graphical User Interface
GUI testing - The purpose of the test is to determine if the graphical user interface operates as expected.
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.
Hybrid Cloud - a cloud computing environment that uses a mix of on-premises, private cloud and third-party, public cloud services with orchestration between the two platforms.
IaaS – see Infrastructure as a Service
IAC – see Infrastructure as Code
Idempotent - the desired state of a server is defined as code or declarations, and the execution of configuration steps are automated to consistently achieve the defined server configuration state time-after-time. Configuration Management tools offer idempotentcy CM tools (e.g., Puppet, Chef, Ansible, and SaltStack)
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.
Immutable infrastructures – a software deployment method in which changes to software modules on a deployment node (E.g. server, container, etc.) are replaced to ensure proper behavior, instead of instantiating an instance with error-prone, time-consuming patches and upgrades or mutations.
Impediment - Anything that prevents from performing work efficiently.
Implementation Under Test (IUT) - 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.
Incident - An unplanned interruption or degradation affecting an IT service.
Incident Management - Process that restores normal service operation as quickly as possible to minimize business impact and to ensure that agreed levels of service quality are maintained.
Information Technology (IT) - anything related to computing technology, such as networking, hardware, software, the Internet, or the people that work with these technologies.
Infrastructure - All hardware, software, networks, facilities and other systems, required to develop, test, deliver, monitor, control or support IT services.
Infrastructure as code (IaC) - is an approach for managing and provisioning computer data centers and cloud services through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools. Infrastructure as code approaches are promoted for cloud computing as infrastructure as a service (IaaS).
Infrastructure test - verify the framework for an EUT operating environment.
Infrastructure-as-a-Service (IaaS) - On-demand access to a shared pool of configurable computing resources. Example service providers are AWS, Azure, GCS, IBM, Oracle.

Internet of Things (IOT) - A network of physical devices that connect to the internet and potentially to each other through web-based or wireless services.
IOT – see Internet of Things
ISO/IEC 20000 - International standard for IT service management. ISO/IEC 20000 is used to audit and certify service management capabilities.
IT – see Information Technology
IT Infrastructure Library (ITIL) - Set of best practice publications for IT service management. Published in a series of five core books representing the stages of the IT service lifecycle which are: Service Strategy, Service Design, Service Transition, Service Operation and Continual Service Improvement.
IT Service - A service provided to a customer from an IT organization.
ITIL – see IT Infrastructure Library
Jenkins - a popular master automation framework tool distributed without commercial cost. Used primarily 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. Reference jenkins.org
Jidoka - sometimes is called autonomation, meaning automation with human intelligence. Jidoka highlights the causes of problems because work stops immediately when a problem first occurs. This leads to improvements in the processes that build-in quality by eliminating the root causes of defects.
Kaizen – Japanese term used in lean manufacturing referring to the practice of continuous improvement.
Kanban - Method of work that pulls the flow of work through a process at a manageable pace.
Kanban Board – Software tool that help teams organize, visualize and manage work.
Kata - A lean management term refers to two linked behaviors: improvement kata and coaching kata. Improvement kata is a repeating four-step routine by which an organization improves and adapts. It makes continuous improvement through the scientific problem-solving method of plan, do, check, act (PDCA) a daily habit
Key Performance Indicator (KPI) - metric used to measure the achievement of critical success factors.
Keywords-Based Testing – A test creation method in which tests are created using pre-defined names that reference programs useful for pre-defined testing applications.
Knowledge Management - 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.
KPI – see Key Performance Indicator.
Kubler-Rose Change Curve – A visualization which describes and predicts the stages of personal and organizational reaction to major changes.
LaaS – see Lab-as-a-Service
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.
Lead Time - Duration of a process or series of processes.
Lean (adjective) - Spare, economical. Lacking richness or abundance.
Lean Enterprise - 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 Production – A process engineering philosophy that focuses on reducing waste and improving the flow of processes to improve overall customer value.
LoadRunner - Tool used to test applications, measuring system behavior and performance under load. Licensed by Hewlett Packard.
Log - In computing, a log file is a file that records either events that occur in an operating system or other software runs, or messages between different users of a communication software.
Longevity Test - The purpose of the test is to determine how a system performs over a long period of time.
Mean Time Between Deploys (MTBD) - Used to measure deployment frequency.
Mean Time Between Failures (MTBF) - Average time that a service can perform its agreed function without unexpected interruption. Often used to measure reliability. Measured from when the service starts working, until the time it fails (uptime).
Mean Time to Detect Incidents (MTTD) - Average time required to detect a failed component or device.
Mean Time to Repair (MTTR) - Average time required to repair a failed component or device. MTTR does not include the time required to recover or restore service.
Mean Time to Restore Service (MTRS) – same as downtime. Time duration measured from when a service fails until it is fully restored and delivering its normal functionality.
Merge - Action of integrating a 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.
Microservices - A software architecture that is composed of smaller modules that interact through APIs and can be updated without affecting the entire system. Microservices is a special case of an implementation approach for service-oriented architectures (SOA) used to build flexible, independently deployable, software systems. Services in a microservice architecture are processes that communicate with each other over a network to fulfill a goal. The microservices approach is a first realization of SOA that followed the introduction of DevOps and is becoming more popular for building continuously deployed systems.
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 Testing – A method for creating tests cases automatically derived from a model of the entity under test. Example tool: Tricentis
Monolithic - A software system with an 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.
MTBD – see Mean Time Between Deploys
MTBF – see Mean Time Between Failures
MTBSI – Mean Time Between Service Interruption, same as MTBF
MTRS - Mean Time to Restore Service
MTTD -Mean Time to Detect Incidents
MTTR -Mean Time to Repair

Muda – Japanese term associated with lean manufacturing systems that means “waste”.
Multi-cloud - the use of multiple cloud computing and storage services in a single heterogeneous architecture. For example, an enterprise may concurrently use separate cloud providers for infrastructure (IaaS) and software (SaaS) services or use multiple infrastructure (IaaS) providers.
Mura - Japanese term associated with lean manufacturing systems that means “Inconsistency or excess variation”.
Muri - Japanese term associated with lean manufacturing systems that means “Overburden, unreasonableness or absurdity”.
NFRs – see Non-functional requirements
Non-functional requirements (NFRs) - 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.
Object Under Test (OUT) - The EUT is a software object or class of objects.
Open Source - denoting software for which the original source code is made freely available and may be redistributed and modified.
Operations (Ops) - Individuals or team involved in the daily operational activities of IT systems and services.
Orchestration - Tasks, usually automated, which setup an environment for a system to operate. Alternative: An approach to building automation that interfaces or “orchestrates” multiple tools together to form a toolchain.
Orchestration (verb) - the automated configuration, coordination, and management of IT systems and software.
Organizational Change - Efforts to adapt the behavior of humans within an organization to meet new structures, processes or requirements.
OUT – see Object Under Test
OS Virtualization - the use of software to allow system hardware to run multiple instances of different operating systems concurrently, allowing you to run different applications requiring different operating systems on one computer system.
Outcome - Intended or actual results.
PaaS – see Platform-as-a-Service
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 (PDCA) – see Deming Cycle.
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 in a toolchain. For example, many tools offer plugins to integrate with Jenkins.
Policies - Formal documents that prescribe what is allowed.
Pre-Flight - This is a class of terms which refers to activities and processes that are conducted on an EUT prior to its integration into the trunk branch.
Priority - The relative importance of an incident, problem or change; based on impact and urgency.
Problem - The underlying cause of one or more incidents.

Process - Structured set of activities designed to accomplish a specific objective. A process takes inputs and turns them into defined outputs.
Process Owner - Role accountable for the overall quality of a process. May be assigned to the same person who carries out the Process Manager role, but the two roles may be separate in larger organizations.
Product Backlog - Requirements for a system, expressed as a prioritized list of product backlog items. The product backlog is prioritized by the Product Owner and includes functional, non-functional and technical team-generated requirements.
Product Owner - An individual responsible for maximizing the value of a product and for managing the product backlog.
Programming-Based Testing – A test creation method in which tests are created by writing code in a programming language. E.g. JavaScript, Python, TCL, Ruby
Regression testing – Tests with the purpose to determine if a new version of an EUT has broken or degraded the performance of an EUT feature that worked previously.
Regulatory compliance testing – Tests with the purpose 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. Or as a verb – to make a release.
Release Acceptance Criterion - Measurable attributes which determine whether a release candidate is acceptable for deployment to production.
Release Candidate - A release package that has been prepared for deployment, may or may not have passed the Release Acceptance Criterion, has not been approved for release and has not been rejected.
Release Management - Process that manages releases and underpins Continuous Delivery and the Deployment Pipeline.
Relevance - A Continuous Testing tenet which emphasizes a preference to focus on the most important tests and test results.
Reliability - Measure of how long a service, component or CI can perform its agreed function without interruption. Usually measured as MTBF or MTBSI.
Reliability Test - A test with the purpose 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. Roll-back changes for an EUT change that resulted in a test case fail verdict.
Remediation Plan - Plan that determines the actions to take after a failed change or release.
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.
REST - Representation State Transfer. Software architecture style of the world-wide 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) - Difference between the benefit achieved and the cost to achieve that benefit, expressed as a percentage.
Revert - same as Roll-back
RFC – see Request for Change
Risk - Possible event that could cause harm or loss or affect an organization's ability to achieve its objectives.
Robot Framework - TDD framework created and supported by Google.
ROI – see Return on Investment
Role - Set of responsibilities, activities and authorities granted to a person or team. One person or team may have multiple roles.
R**oll-back** - Software changes which have been integrated are removed from the integration.
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.
SaaS – see Software-as-a-Service
SAFE – see Scaled Agile Framework
Sanity Test - A very basic set of tests that determine if a software is functional at all.
SAT – see System Acceptance Testing
Scaled Agile Framework (SAFE) - A proven, publicly available, framework for applying Lean-Agile principles and practices at an enterprise scale.
Scrum - A framework for effective team collaboration on software projects.
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.
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, the Development Team, and a Scrum Master.
Security (Information Security) - Practices intended to protect the confidentiality, integrity and availability of IT systems from those with malicious intentions.
Security tests – Tests with the purpose 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.
Serverless Computing - is a cloud-computing execution model in which the cloud provider runs the server, and dynamically manages the allocation of machine resources. Pricing is based on the actual amount of resources consumed by an application, rather than on pre-purchased units of capacity.
Service - Means of delivering value to clients by facilitating outcomes clients want to achieve. As a verb – to provide service.
Service Catalog - Subset of the Service Portfolio that consists of services that are live or available for deployment. Has two aspects: Business/Customer Service Catalog (visible to customers) and the Technical/Supporting Service Catalog.
Service Desk - Single point of contact between the service provider and the users.
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.
Service Oriented Architecture (SOA) - a style of software design where services are provided to the other components by application components, through a communication protocol over a network.

Service Provider - Organization that supplies services to one or more internal or external customers.
Service Request (SR) - User request for a standard service from an IT service provider.
Shift Left - A Continuous Testing tenet which emphasizes the preference to execute tests as early as possible in the DevOps pipeline. An approach that strives to build quality into the software development process by incorporating testing early and often.
Site Reliability Engineering (**SRE**) is a discipline that incorporates aspects of software engineering and applies them to infrastructure and operations problems. The main goals are to create ultra-scalable and highly reliable software systems.
SLA – see Service Level Agreement
SMART Goals - Specific, measurable, achievable, relevant and time-bound goals.
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.
SOA – see Service Oriented Architecture
SOAP - Simple Object-Oriented Transfer. Protocol for exchanging structured information in the implementation of web services in computer networks.
SOAPUI – popular tool for testing APIs.
Software Version Management System - A repository tool which is used to manage software changes. Examples are Git, GitHub, GitLab, Perforce.
Software-as-a-Service (SaaS) - Category of cloud computing services in which software is licensed on a subscription basis.
Sprint - A time-boxed iteration of work during which an increment of product functionality is implemented.
SR – see Service Request
Stakeholder - Person who has an interest in an organization, project or IT service. Stakeholders may include customers, users and suppliers.
Static code analysis – A methodology to detect source code logic errors and omissions such as memory leaks, un-initialized variables, and un-initialized pointers.
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 Acceptance Testing (SAT) – Testing from the point of view of a complete system.
System of Record - Authorized data source stored in a version management system. E.g. source code, test code, test results. Single source of truth.
System Test - The purpose of the test is to determine if a complete system performs as expected in its intended configurations.
Software Under Test (SUT) – the EUT is software.
System Under Test (SUT) - The EUT is an entire system. E.g. Bank teller machine is being tested.
SUT - see System Under Test or Software Under Test
Tag-based test selection method - Tests and Code modules are pre-assigned tags. Tests are selected for a build matching pre-assigned tags.
TDD – see Test Driven Development

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 which refers to the methodology used to create test cases.
Test Driven Development (TDD) - Test Driven Development. 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. Linux, Windows version etc.), 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 and environment 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 test selection method - A matrix of correlation factors correlates test cases and code modules according to test result (verdict) trends. Tests cases are selected for a build by matching the highest correlating test cases to the code changes in a build.
Test Roles and Responsibilities - 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 implement one or more test cases depending on the data and configuration used to run it.
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 during 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 - Class that indicates what the purpose of the test is.
Test version - The version of files used to test a specific build.
Tester – Role of a person who has responsibility to test a system or service.
The Three Ways – Maturity levels of DevOps described in The Phoenix Project. – Continuous Flow, Continuous Feedback, and Continuous improvement.

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.
Time to Value - Measure of the time it takes for the business to realize value from a feature or service.
Tool - This class describes tools that orchestrate, automate, simulate and monitor EUTs and infrastructures.
Tool chain - set of distinct software development tools that are linked (or chained) together by specific stages to automate an end to end CI/CD pipeline.
Trunk - The primary source code integration repository for a software product.
UAT – see User Acceptance Testing
Unit Test - a level of software testing where individual units/ components of a software are tested. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.
Uptime – see Mean Time Between Failures (MTBF)
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.
User Acceptance Testing (UAT) – end users testing from the point of view of usability.
Value Stream - All the activities to go from a customer request to a delivered product or service.
Value Stream Management - Mapping, optimizing, visualizing, and governing business value flow (including epics, stories, and work items) through heterogeneous enterprise software delivery pipelines to operations.
Value Stream Mapping - Visualization that depicts the flow of information, materials and work across functional silos with an emphasis on identifying and quantifying waste, including time and quality.
Variable Speed IT - An approach where traditional and digital processes co-exist within an organization while moving at their own speed.
Velocity - 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 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.
Version control tools - Same as software version management tool
Yokoten – Japanese term used in lean manufacturing referring to sharing of best practices
Waste (Lean Manufacturing) – A thing or process that does not add value to a product.
Waterfall Process - Linear and sequential approach to managing software design and development.
Water-scrum-fall - A hybrid approach to application lifecycle management that combines waterfall and Scrum development methodologies.
White-Box - Test cases use extensive knowledge of the internal design structure of an EUT Eg. software unit test that is designed to test specific code logic. Same as Clear-Box Testing and Glass-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.
WIP – see Work in Progress

Work in Progress (WIP) - Any work that has been started but has not been completed.
Workaround - Temporary way to reduce or eliminate the impact of incidents or problems.