Glossary

This handout provides definitions of terms used in the course.

Andon Cord
A device used to stop a production line in case of a significant issue

Blameless Postmortem
Retrospective of an incident that avoids blaming individuals for errors and is focused on learning about your system and organization; it typically contains:

1. A description of the incident
2. A description of contributing causes, including participant assumptions and perceptions
3. How the incident was stabilized or fixed
4. A timeline of events, including all actions taken to resolve the incident
5. How the incident affected customers
6. Lessons learned, remediations, and corrective actions

Build, Measure, Learn
A feedback loop that is a core component of the Lean Startup methodology

- Build – minimum viable product
- Measure – outcome and internal metrics
- Learn – your problem and your solution
- Repeat – go deeper where it's needed

CAMS
A set of core DevOps values: culture, automation, measurement, sharing; sometimes known as CALMS, with the addition of lean

Cloud Computing
"A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"

—NIST Special Publication 800-145

Configuration Management
Management of change control for system configuration after initial provision; maintaining and upgrading the application and its dependencies

Continuous Delivery
A DevOps methodology in which engineers code, test, and release software into a running environment frequently, in really small batches, so that you can improve the overall quality and velocity; it is continuous integration plus continuous deployment

Continuous Deployment
After testing every code change, automatically deploying it to a running environment, including production

Continuous Integration
Automatically building and unit testing an entire application whenever a code change is checked into source control, in small increments

Declarative (Functional)
A desired state is defined in configuration and relies on the tool to configure a system to match that declared state

Deployment
Automatically deploying and upgrading applications on a server

Developers
Engineers principally tasked with writing applications

DevOps
"The practice of operations and development engineers participating together through the entire service lifecycle; from the design and development process all the way to production support. . . .
DevOps is also characterized by operations staff making use of many of the same techniques as developers for their systems work."

—https://theagileadmin.com/what-is-devops/

DevSecOps
The extension of DevOps culture for the benefit and inclusion of security

Doublespeak
Deliberately obscured language full of euphemisms designed to spin or avoid responsibility for events

Five Whys
A kaizen continuous improvement technique used to get to the root of a problem by iteratively asking "why" something happened to uncover successive layers of detail and discover underlying factors that contributed

Gemba (現場)
The "real place" where work happens

Idempotent
The ability to execute a function repeatedly, resulting in the same outcome

Imperative (Procedural)
Commands necessary to produce the desired state are defined and executed

Infrastructure as a Service (IaaS)
A type of cloud computing in which you use infrastructure made available over the internet, without having to manage the underlying hardware

Infrastructure as Code (IaC)
A DevOps methodology in which we treat systems the same way as code; system specifications should be checked into source control, go through a code review, and then automatically create real systems from the spec and manage them programmatically

Kaizen (改善)
“Improvement”; in a business context, continuous improvement

Kaizen’s Guiding Principles
- Good processes bring good results
- Go see for yourself to grasp the current situation (gemba)
- Speak with data and manage by facts
- Take action to contain and correct root causes of problems
- Work as a team
- Kaizen is everybody’s business

Lean Management
A DevOps methodology focusing on using small batches of work, work-in-progress limits, feedback loops, and visualization

Lean Software Development: Seven Principles
1. Eliminate waste.
2. Amplify learning.
3. Decide as late as possible.
4. Deliver as fast as possible.
5. Empower the team.
6. Build integrity in.
7. See the whole.

Muda (無駄)
Work that absorbs resources, but adds no value

The Seven Wastes of Lean Software

1. Partially done work
2. Extra features
3. Relearning
4. Handoffs
5. Delays
6. Task switching
7. Defects

Mura (斑)
Work coming in unevenly instead of a constant or regular flow

Muri (無理)
Unreasonable work imposed on workers and machines

Operations Engineers
Engineers principally tasked with build, deployment, maintenance, and monitoring of running computer systems

Orchestration
Performing coordinated operations across multiple systems

PaaS (Platform as a Service)
A type of cloud computing in which you use an application hosting platform made available over the internet, without having to manage the underlying systems

People over Process over Tools
A DevOps methodology that focuses on identifying who's responsible for a job function first, defining the process that needs to happen around those people, and then selecting a tool to perform that process —in that order

Production Environment
The live end-user-facing installation of a service, as opposed to a development or test environment

Provisioning
Making a server ready for operation, including hardware, operating system, system services, and network connectivity

SaaS (Software as a Service)
A type of cloud computing in which you use an application made available over the internet, without having to manage the infrastructure or software

Self-Service
The ability for an end user to initiate a process without having to go through other people

Shift Left
Performing functions earlier in the software delivery lifecycle as part of the development phase (testing, security, reliability engineering), instead of leaving them until after the software is delivered

Serverless Architecture
"Serverless . . . is run in stateless compute containers that are event-triggered, ephemeral (may only last for one invocation), and fully managed by a third party."

— Martin Fowler

Site Reliability Engineer (SRE)
A job title for an engineer tasked with operational support but who performs that role by taking a software engineering approach

The Three Ways
The principles of DevOps

1. Systems Thinking
2. Amplify Feedback Loops
3. A Culture of Continuous Experimentation and Learning

—Gene Kim and Mike Orzen

Transparent Uptime
Communicating frankly with your customers about system outages; the four prerequisites for doing this successfully:

1. Admit failure.
2. Sound like a human.
3. Have a communication channel.
4. Above all else, be authentic.

Visible Ops–Style Change Control

A DevOps methodology that implements change control with an emphasis on eliminating fragile artifacts, creating a repeatable build process, and then building an environment of continual improvement

Well-Behaved Tools

Tools that can be manipulated in a developer-friendly way—they or their config can be checked into source control, tested, automatically deployed, and integrate with other tools in a toolchain