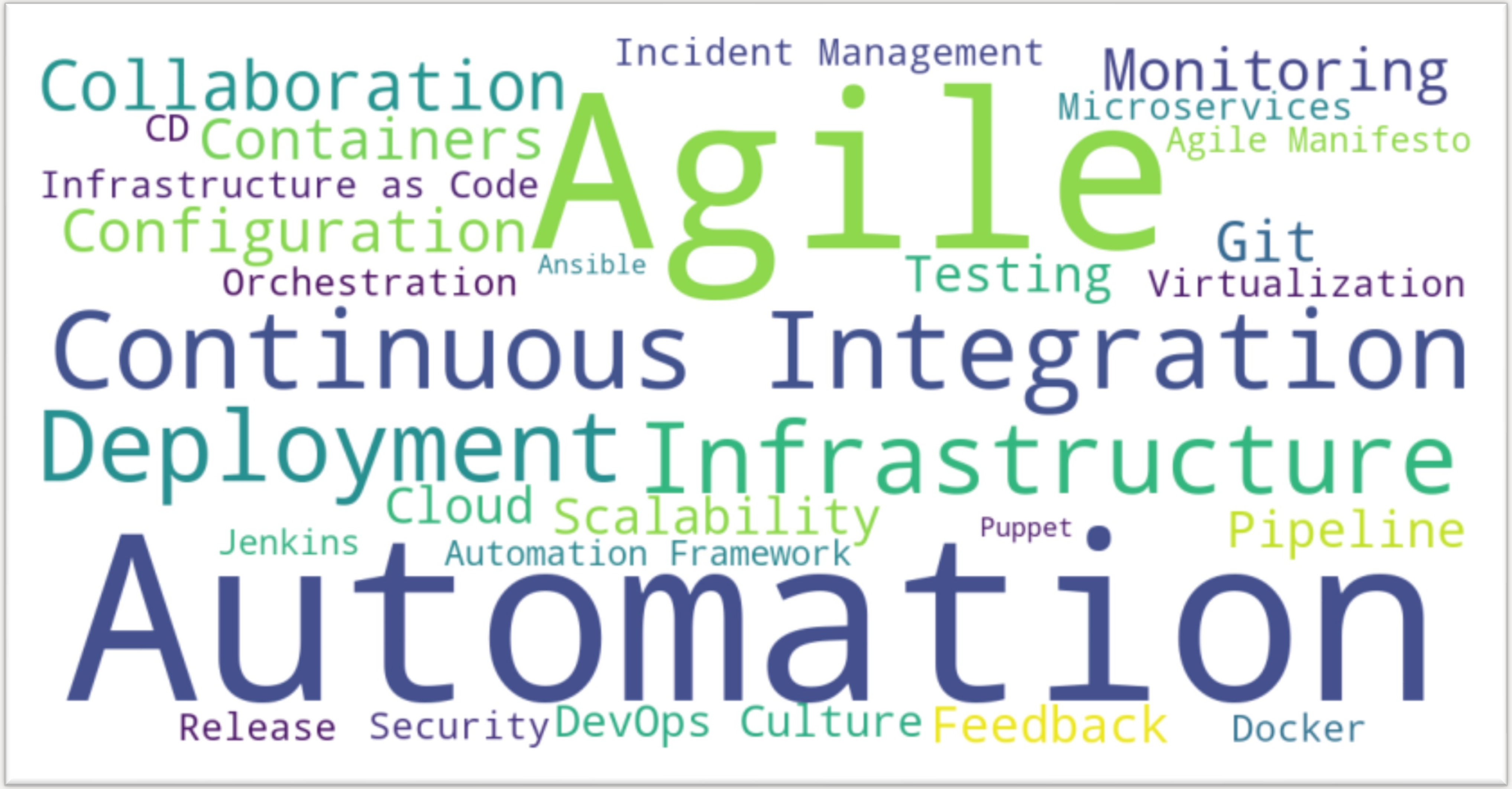
Intro to DevOps

COMPSCI 308 – 2022-04-11

Luke Moffett, Duke CS

Duke

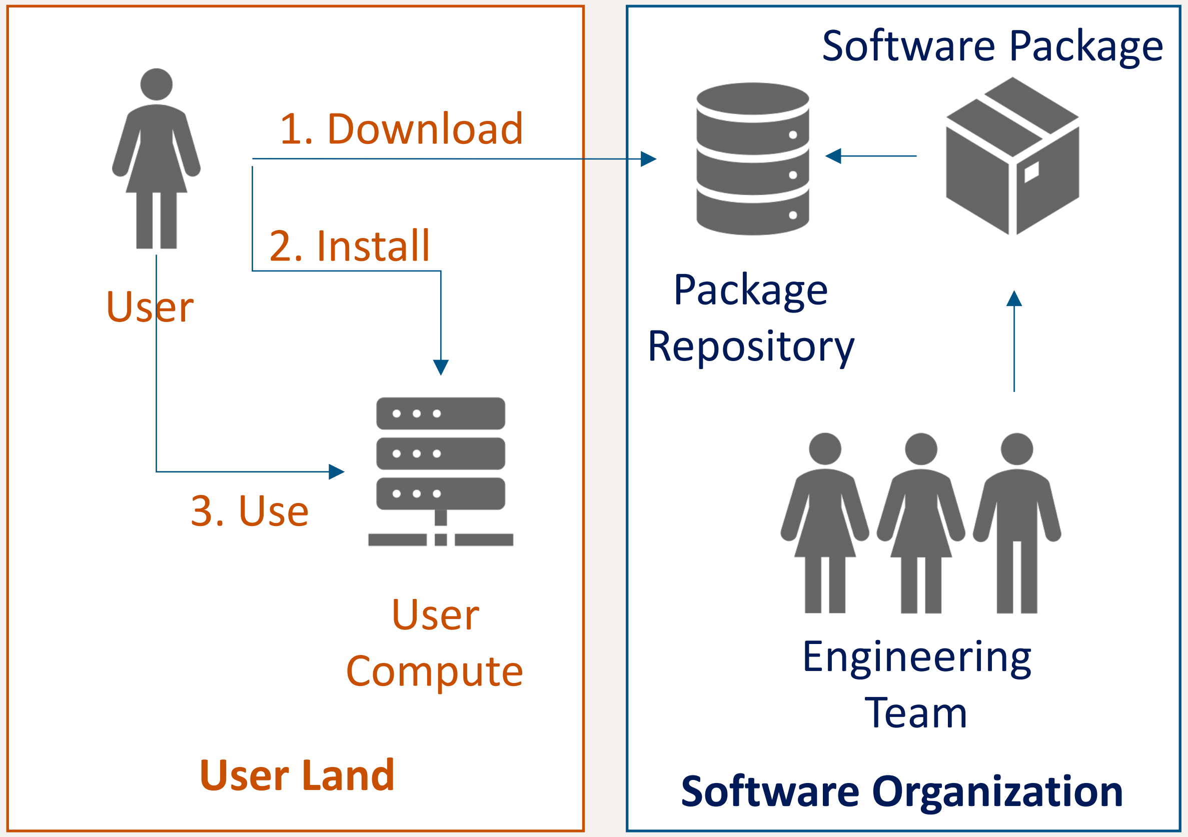
Related Ideas



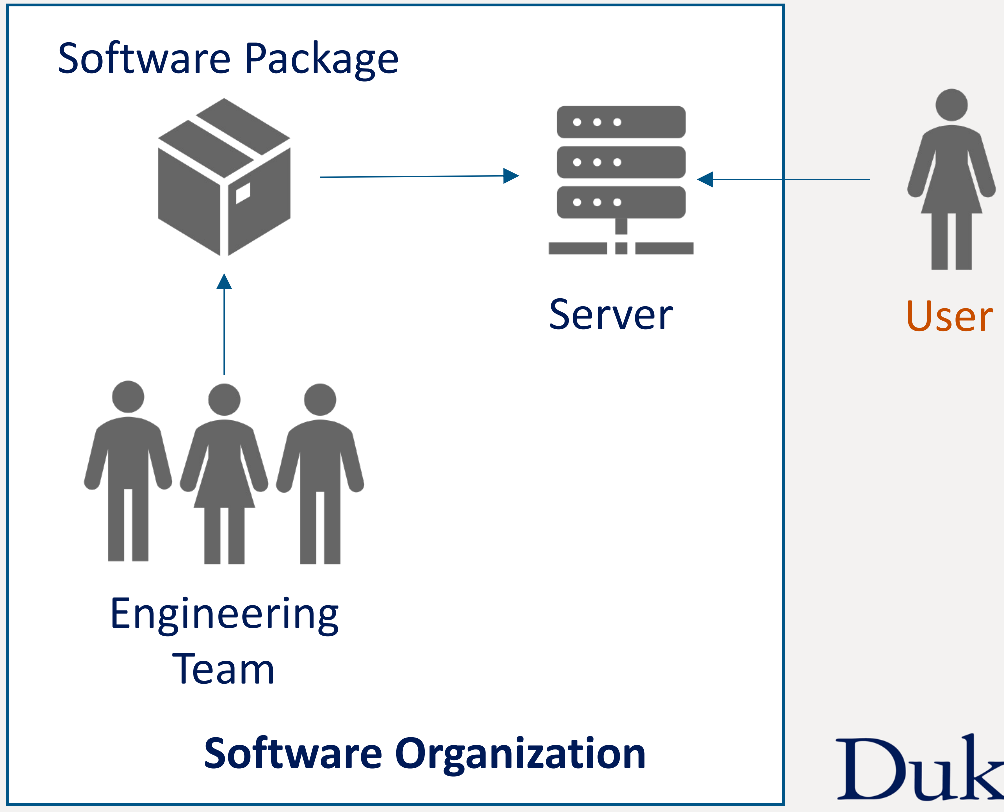
Duke

Software Delivery Models

Software as a Product



Software as a Service



Duke

Traditional Software Operations

Development

- Prioritizes Features with Stakeholders
- Implements Features (Coding)
- Tests Features
- Initiates Release to Users

Operations

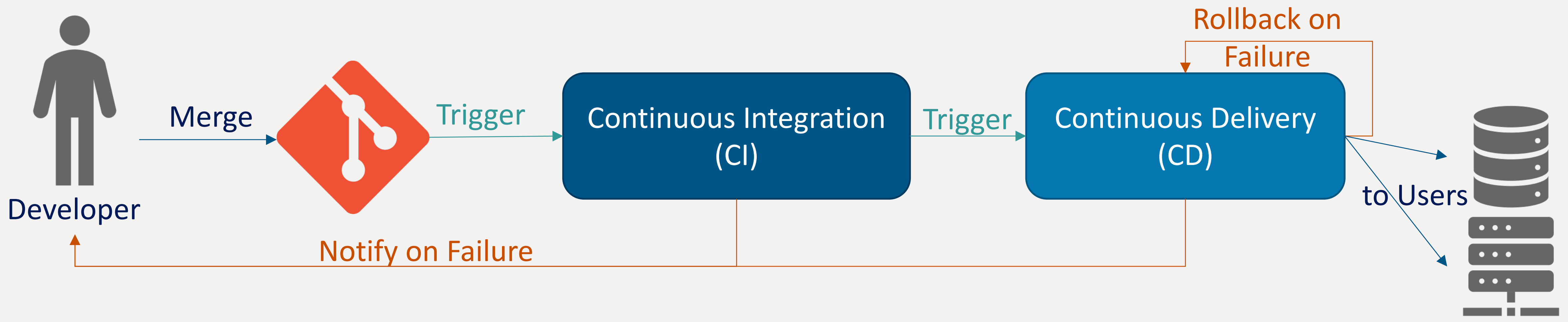
- Executes Release to Users (Deployment)
- Purchases and Maintains Servers
- Handles technical issues for Users (Service Desk)
- Handles outages (Incident Management)

Moving Faster with DevOps Principles

- Automation
- Tight Integration between Development and Operations Teams
- Small Changes with Fast Feedback Loops
- “Shift Left”

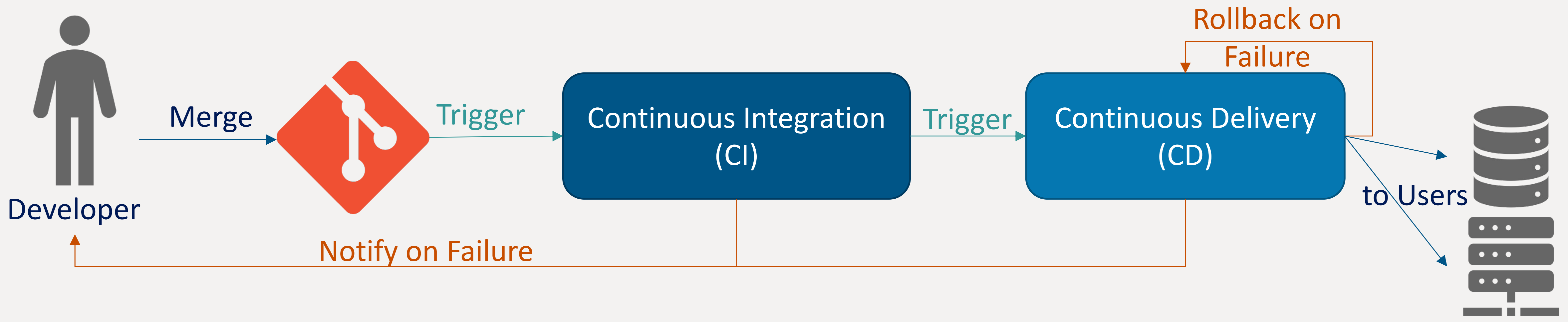
Duke

Key Idea: The Software Delivery Pipeline



Duke

Key Idea: The Software Delivery Pipeline



Continuous Integration (CI)

Pre-Commit Checks	Pre-Merge Checks	Post-Merge Checks	Integration Testing
Code Formatting	Code Compilation	Unit Testing & Coverage	Tests on Fresh Install
Commit Message	Can Merge?	Quality & Security Analysis	Dynamic Security Tests

Continuous Delivery (CD)

Release Automation	
Install Package	Analyze Telemetry
Publish Docs	Notify Engineers

Examples

DORA Metrics

Lead Time for Changes

- Length of time between when a feature is agreed to to when it is available to users
- Low is good
- Mean days – eg 13 days

Deployment Frequency

- Rate at which ANY new code is made available to users
- High is Good
- Mean releases/day – eg 5/day

Change Failure Rate

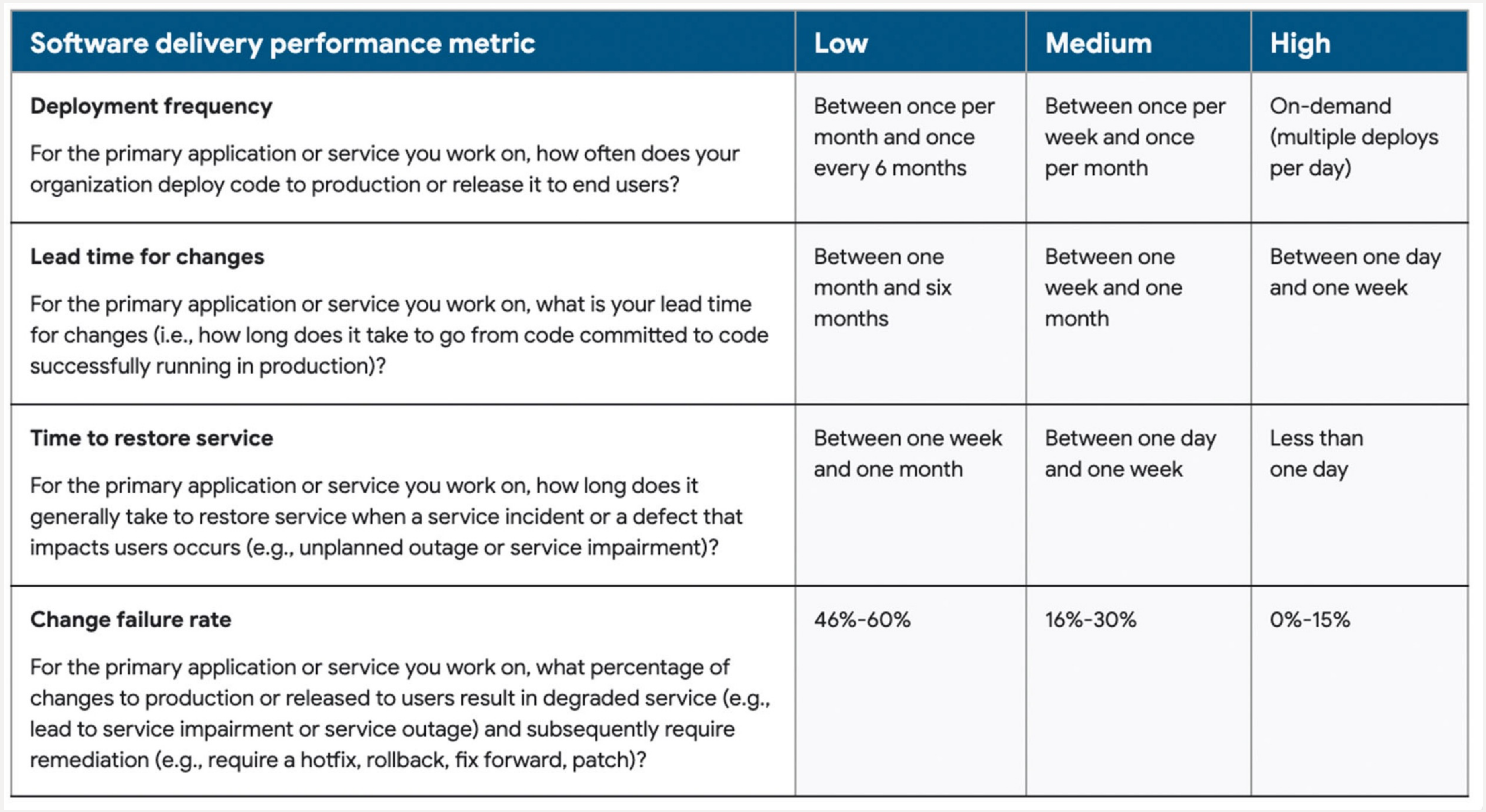
- Percentage of initiated changes that do not complete successfully
- Low is Good
- As % - eg, 12.3%

Time to Restore Service

- After an outage starts, the length of time until users are no longer experiencing the outage
- Low is Good
- Mean Minutes – eg, 104 minutes

Duke

DORA Metrics



Software delivery performance metric	Low	Medium	High
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	Between once per month and once every 6 months	Between once per week and once per month	On-demand (multiple deploys per day)
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Between one month and six months	Between one week and one month	Between one day and one week
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Between one week and one month	Between one day and one week	Less than one day
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	46%-60%	16%-30%	0%-15%

Low/Medium/High represent groupings of organizations based on overall DORA metric performance

Source: https://cloud.google.com/blog/products/devops-sre/dora-2022-accelerate-state-of-devops-report-now-out

Duke

Key Engineering Practices for DevOps

- Design
  - Loose Coupling Between Systems
  - Single Responsibility Principle Everywhere
- Automation
  - CI – Code Analysis and Testing
  - CD – Safe Deployment, Incremental Rollouts, Deploy when Ready
  - IaaC
- Operations
  - Observability, Monitoring, and Alerting
  - On-Call Rotations and Escalation Paths
  - Blameless Postmortems
- Reliability
  - Fault Tolerant Design, Redundancy, and Self Healing
  - Safe Failure
  - Backups and Data Recovery

Duke

Products, Tools, and Technologies

- Too Many to Count
- Common Tools
  - SCM: GitHub/GitLab/BitBucket
  - Containerization: Kubernetes/Docker
  - Hosting: AWS/Microsoft Azure/Google Cloud
  - IaaC: Terraform/Chef/Puppet/Ansible
  - Observability: Splunk/DataDog/New Relic/AppDynamics/Dynatrace/Lightstep
- The Cloud Native Computing Foundation Landscape shows the breadth of the environment:
  - https://landscape.cncf.io/

Duke

Summary

- DevOps is more a set of ideas than specific practices
- DevOps is best understood in the industrial context that created it
- Much of the major growth in tech in the 2010’s was companies enabling these ideas, Cloud providers being the biggest examples (AWS, Microsoft Azure, Google Cloud, etc)
- The industry has coalesced around a set of high-level practices and metrics, but the implementation details are still a matter of considerable debate and experimentation
- By extension, the tooling landscape continues to evolve. There are many startups providing new tooling.