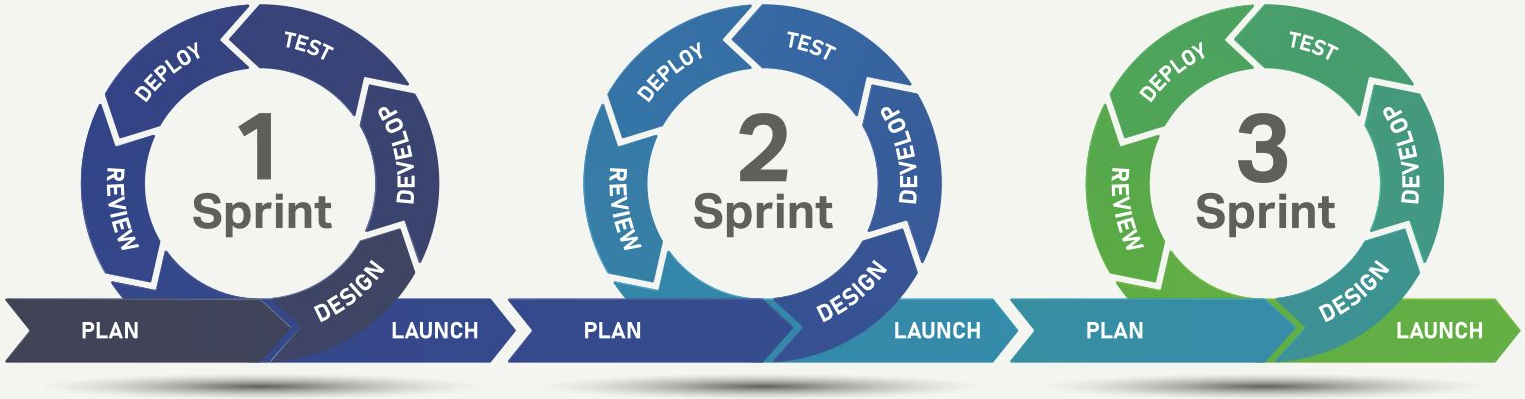
Ops and DevOps

Logistics

- Weekly survey
- How was the cloud deployment task?

AGILE
methodology



What does “Ops” do?

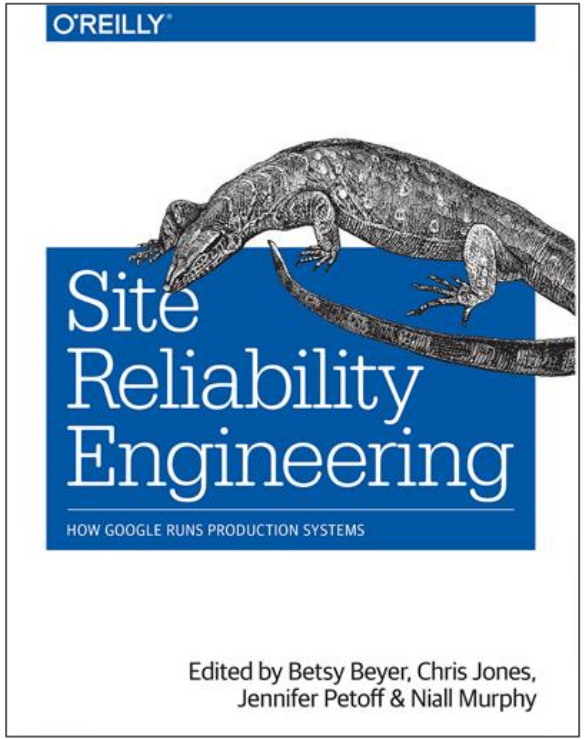
Operations

Provision and monitor the system in production, respond to problems

Avoid downtime, scale with users, manage operating costs

Heavy focus on infrastructure

Traditionally sysadmin and hardware skills



Core responsibilities

- Set up and maintain servers
- Provision networking, storage, and compute resources
- Automate deployment pipelines
- Manage staging and production environments
- Observability with logs, metrics, and alerts
- Plan incident responses
- Respond to failures
- Ensure security and compliance requirements
- Scale infrastructure

Operators on a Team

Operators cannot work in isolation

Rely on developers for software quality and performance

Negotiate service level agreements and budget (e.g., 99.9% vs 99.99% availability)

Risk management role (not risk avoidance)


SOFTWARE DEVELOPERS
QA

Dev vs. Ops

Breakout

Imagine we are building our own Netflix product…

- What are 3 problems that may arise in a deployment?
- What are 3 ways we could measure the operations side?
- How might we get “Dev” and “Ops” to work more collaboratively?

Wall of Confusion

David is a DEVeloper !

David wants to maximize **change**

Peter is an OPerator !

Peter wants to optimize **stability**

https://www.devops.ch/2017/05/10/devops-explained/

Developers

- Coding
- Testing, static analysis, reviews
- Continuous integration
- Bug tracking
- Running local tests and scalability experiments
- ...

Operations

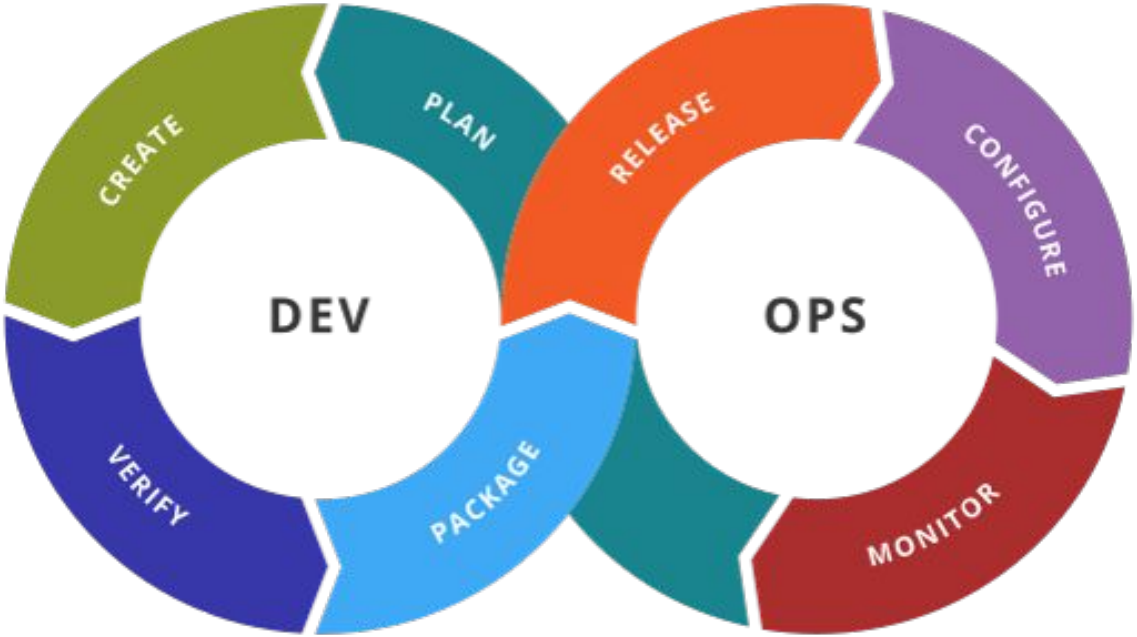
- Allocating hardware resources
- Managing OS updates
- Monitoring performance
- Monitoring crashes
- Managing load spikes, …
- Tuning database performance
- Running distributed at scale
- Rolling back releases
- ...

QA responsibilities in both roles

Quality Assurance does not stop in Dev

- Ensuring product builds correctly (e.g., reproducible builds)
- Ensuring scalability under real-world loads
- Supporting environment constraints from real systems (hardware, software, OS)
- Efficiency with given infrastructure
- Monitoring (server, database, etc)
- Bottlenecks, crash-prone components, … (possibly thousands of crash reports per day/minute)

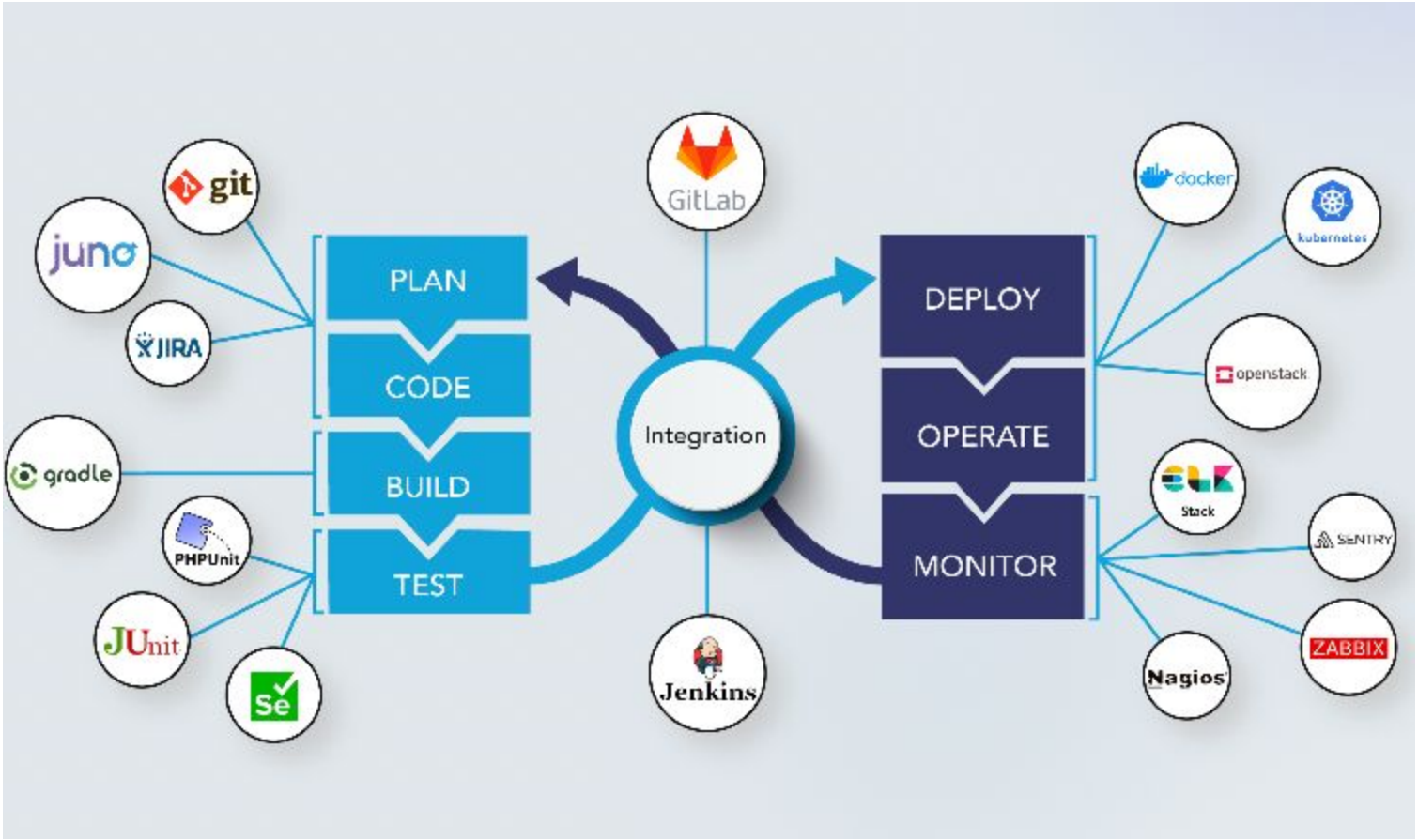
Evolving Agile to DevOps



Roles and job titles

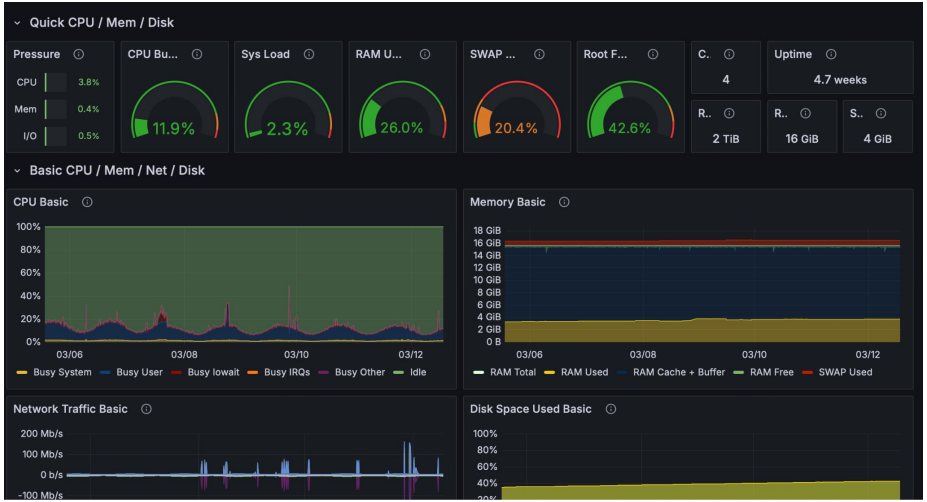
- Systems Administrators
- Site Reliability Engineer
- Platform/Infrastructure Engineer
- Operations Engineer
- Security Operations
- DevOps Engineer

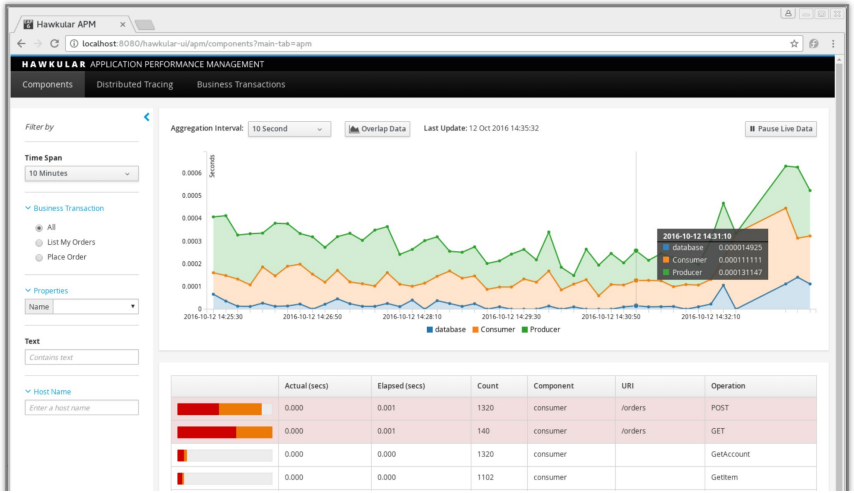
Note: Many varying titles with overlapping responsibilities


git
juno
JIRA
gradle
PHPUnit
JUnit
Se
PLAN
CODE
BUILD
TEST
GitLab
Integration
Jenkins
DEPLOY
OPERATE
MONITOR
docker
kubernetes
openstack
Stack
SENTRY
ZABBIX
Nagios

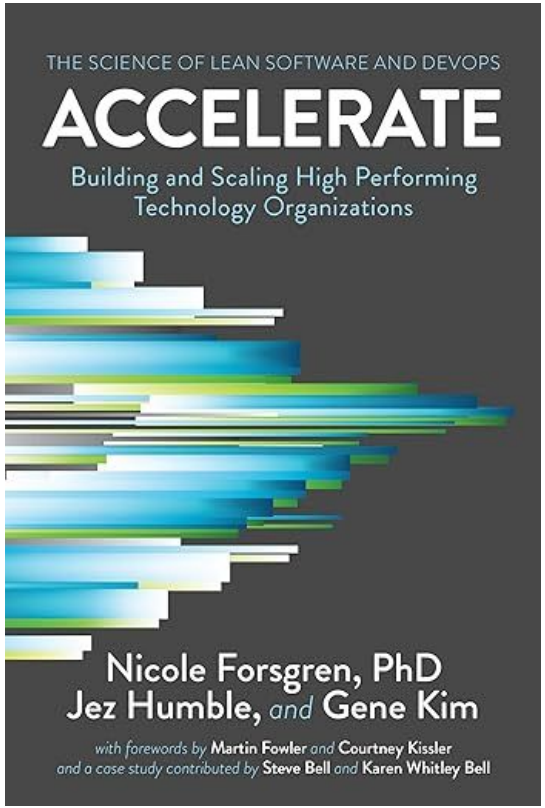
Observability

Can you see or measure how your system is doing?





Accelerate – Forsgren et al



Key metrics:

1. Delivery Lead Time: shorter is better
2. Deployment Frequency: higher is better
3. Mean Time To Restore (MTTR): shorter is better
4. Change Fail Percentage: Lower Is Better

Studies Demonstrate Key Capabilities and Categories

The book reveals 24 key capabilities that have a proven impact on the performance of software delivery. These are divided into 5 categories:

- Continuous delivery
- Architecture
- Product and process
- Lean management and monitoring
- Culture