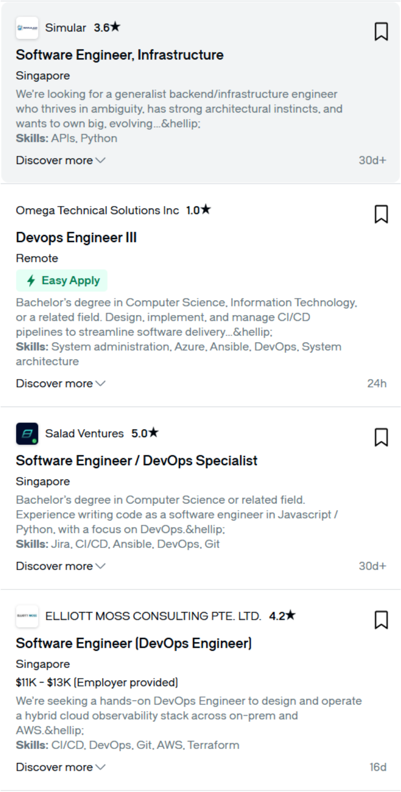
CS3213: Foundations of Software Engineering

DevOps

DevOp Skills in Need



Simular 3.6★

Software Engineer, Infrastructure

Singapore

We're looking for a generalist backend/infrastructure engineer who thrives in ambiguity, has strong architectural instincts, and wants to own big, evolving…&hellip;

Skills: APIs, Python

Discover more 30d+

Omega Technical Solutions Inc 1.0★

Devops Engineer III

Remote

Easy Apply

Bachelor's degree in Computer Science, Information Technology, or a related field. Design, implement, and manage CI/CD pipelines to streamline software delivery…&hellip;

Skills: System administration, Azure, Ansible, DevOps, System architecture

Discover more 24h

Salad Ventures 5.0★

Software Engineer / DevOps Specialist

Singapore

Bachelor's degree in Computer Science or related field. Experience writing code as a software engineer in Javascript / Python, with a focus on DevOps.&hellip;

Skills: Jira, CI/CD, Ansible, DevOps, Git

Discover more 30d+

ELLIOTT MOSS CONSULTING PTE. LTD. 4.2★

Software Engineer (DevOps Engineer)

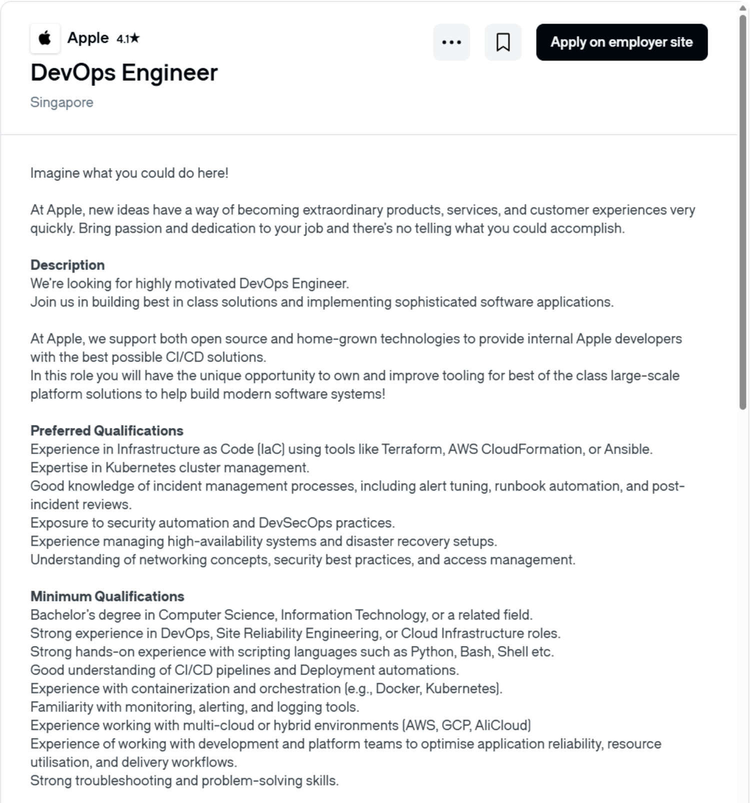
Singapore

$11K - $13K (Employer provided)

We're seeking a hands-on DevOps Engineer to design and operate a hybrid cloud observability stack across on-prem and AWS.&hellip;

Skills: CI/CD, DevOps, Git, AWS, Terraform

Discover more 16d



Apple 4.1★

DevOps Engineer

Singapore

Imagine what you could do here!

At Apple, new ideas have a way of becoming extraordinary products, services, and customer experiences very quickly. Bring passion and dedication to your job and there's no telling what you could accomplish.

Description

We're looking for highly motivated DevOps Engineer.

Join us in building best in class solutions and implementing sophisticated software applications.

At Apple, we support both open source and home-grown technologies to provide internal Apple developers with the best possible CI/CD solutions.

In this role you will have the unique opportunity to own and improve tooling for best of the class large-scale platform solutions to help build modern software systems!

Preferred Qualifications

Experience in Infrastructure as Code (IaC) using tools like Terraform, AWS CloudFormation, or Ansible.

Expertise in Kubernetes cluster management.

Good knowledge of incident management processes, including alert tuning, runbook automation, and post-incident reviews.

Exposure to security automation and DevSecOps practices.

Experience managing high-availability systems and disaster recovery setups.

Understanding of networking concepts, security best practices, and access management.

Minimum Qualifications

Bachelor's degree in Computer Science, Information Technology, or a related field.

Strong experience in DevOps, Site Reliability Engineering, or Cloud Infrastructure roles.

Strong hands-on experience with scripting languages such as Python, Bash, Shell etc.

Good understanding of CI/CD pipelines and Deployment automations.

Experience with containerization and orchestration (e.g., Docker, Kubernetes).

Familiarity with monitoring, alerting, and logging tools.

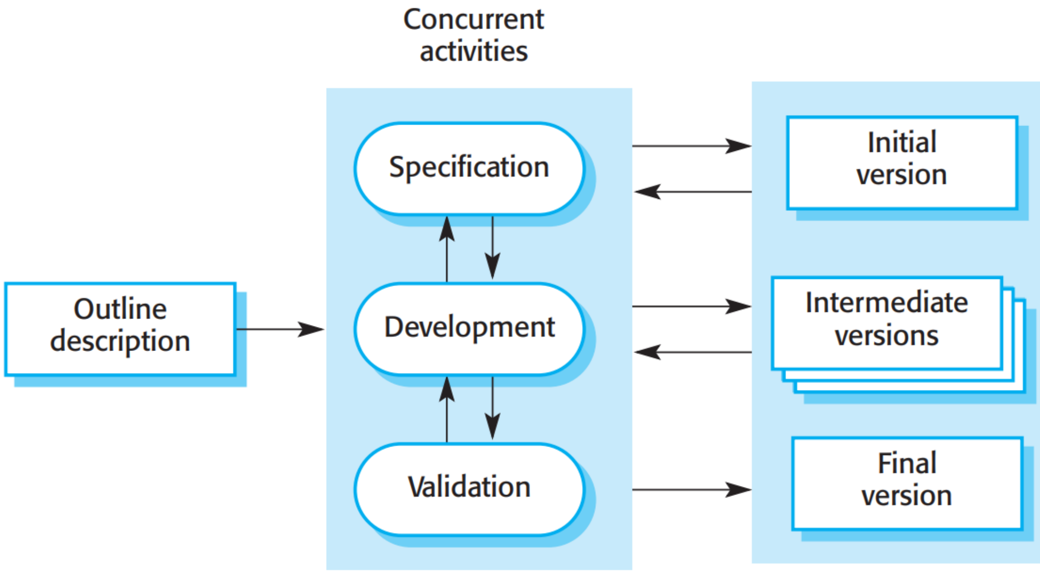
Experience working with multi-cloud or hybrid environments (AWS, GCP, AliCloud)

Experience of working with development and platform teams to optimise application reliability, resource utilisation, and delivery workflows.

Strong troubleshooting and problem-solving skills.

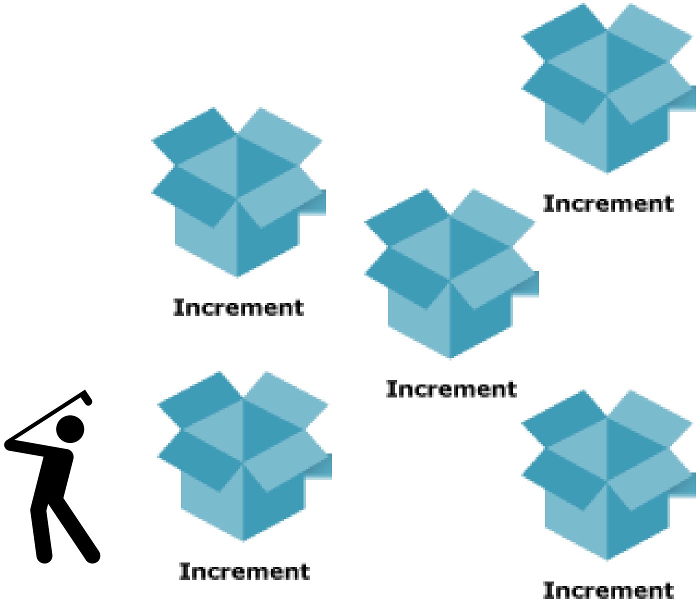
https://www.glassdoor.sg/

Incremental Development



Modern software process models often develop software in increments

Increments and Ops



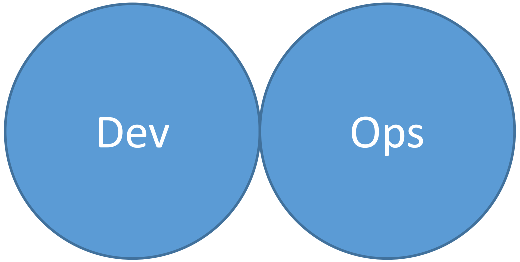
Development/Scrum team

Operations team

https://www.scrum.org/resources/what-scrum-module

Silos

- Developers and Operations were traditionally separate (“siloed”)
- Developers: responsible for creating increments/value
- Operations: managing and maintaining IT infrastructure
- Interests can be conflicting!



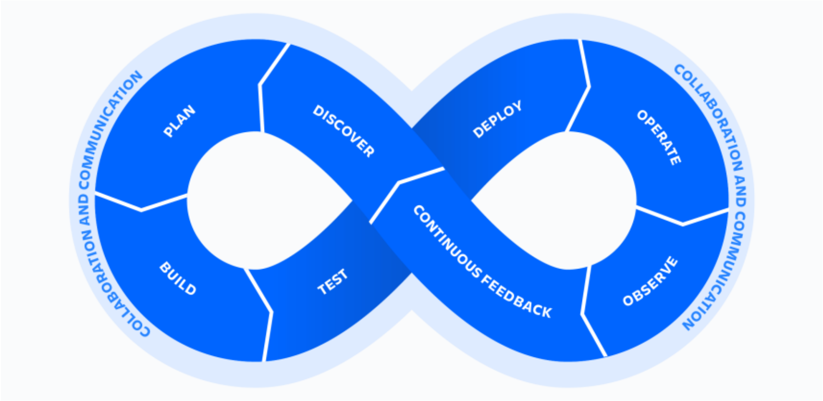
DevOps

- **DevOps**: combines the terms *Development* and *Operations*
- Typically associated with a mindset, practices, and tools
- Idea: the **whole organization** (including development and operation teams) **work together** to deliver value
  - In particular, Development and Operations
- Make production **deployments routine and predictable**

DevOps vs. Agile

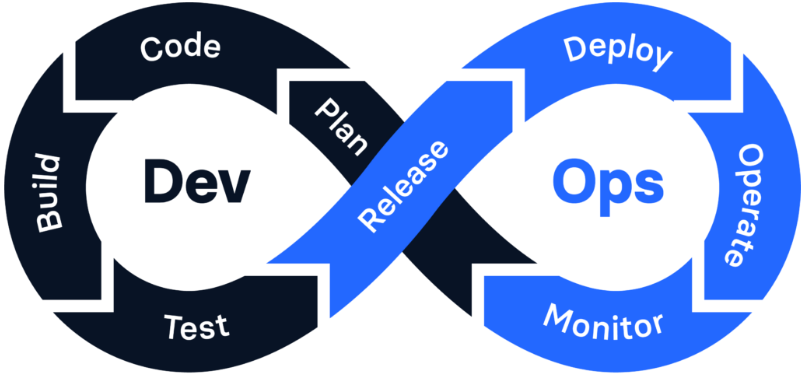
- Many parallels with agile and lean software engineering
- Often seen as a logical continuation of agile software development
- Also parallels to Lean and TPS

DevOps



PLAN
DISCOVER
DEPLOY
OPERATE
BUILD
TEST
CONTINUOUS FEEDBACK
OBSERVE
COLLABORATION AND COMMUNICATION
COLLABORATION AND COMMUNICATION

https://www.atlassian.com/devops



https://ubiqware.net/en/our-approach-to-devops/

Principle: Flow

- DevOps emphasizes the concept of *flow*, similar to its notion in the TPS
- Reduce batch sizes: small batch sizes can be delivered more quickly
- Reduce the number of handoffs: passing work between departments and teams increases time to deployment
- Continually identify and eliminate constraints and waste

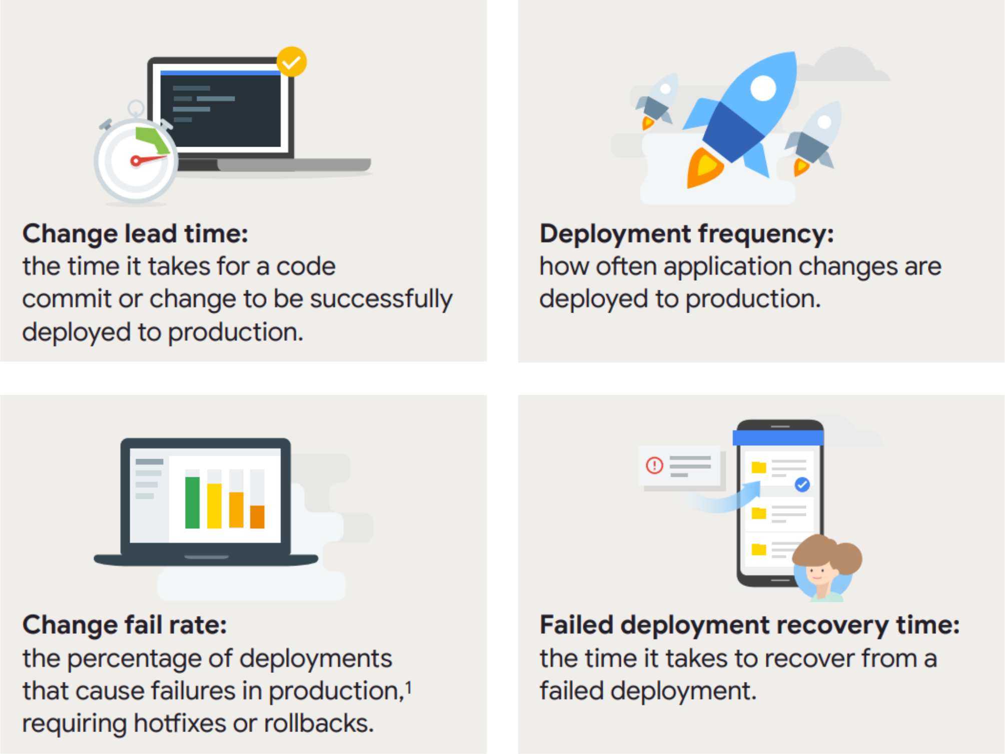
Principle: Feedback Loops

- Insight: failure and issues cannot prevented for complex systems
- Feedback and feedforward loops to detect problems and address them while they are smaller, cheaper, and easier to fix
- Fast and constant feedback from all stages of the value stream
- Automatic build, integration, and test processes to obtain immediate feedback
- Propagate new knowledge across the team/company
- Find and fix problems where the work is performed

Principle: Continuous Learning and Experimentation

- Require and actively promote learning
- Instead of work being rigidly defined, it is dynamic
- Performing experiments to generate new improvements enabled by standardization of work procedures and documentation of the results
- Mechanism so that new local learnings are propagated to the rest of the organization

DevOps Research and Assessment (DORA) Key Metrics



Change lead time:
the time it takes for a code commit or change to be successfully deployed to production.

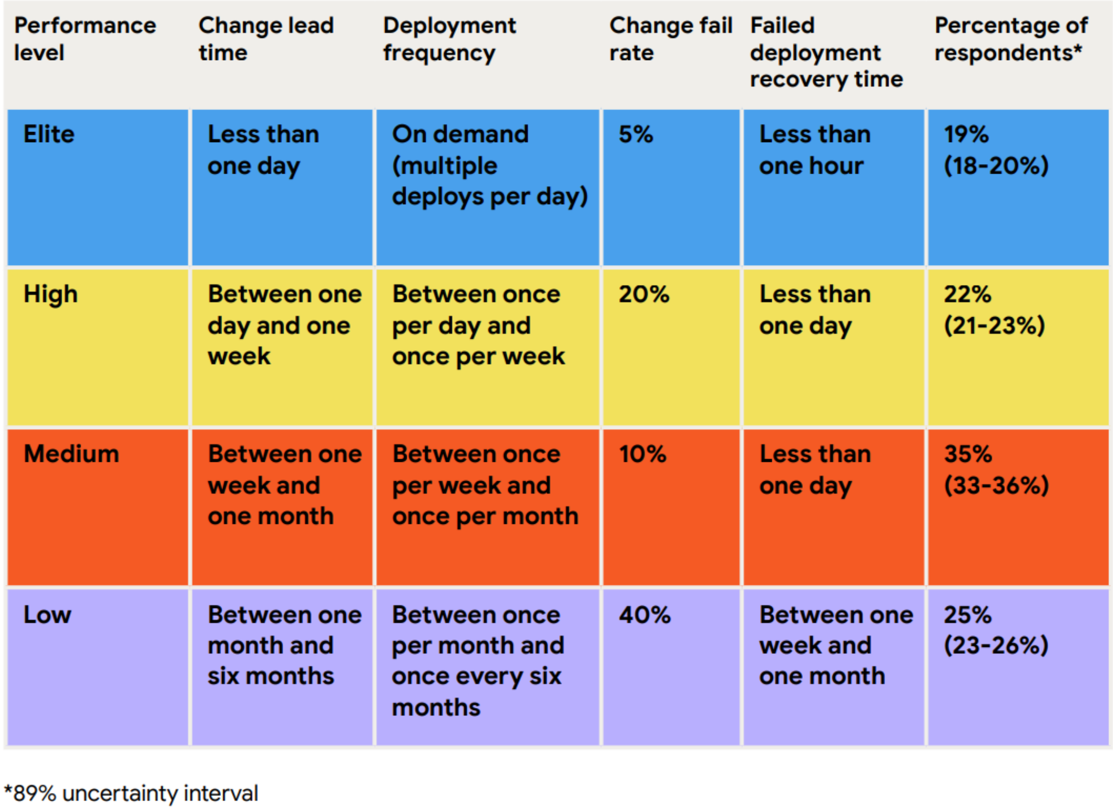
Deployment frequency:
how often application changes are deployed to production.

Change fail rate:
the percentage of deployments that cause failures in production,[1] requiring hotfixes or rollbacks.

Failed deployment recovery time:
the time it takes to recover from a failed deployment.

https://services.google.com/fh/files/misc/2024_final_dora_report.pdf

DORA: Performance Levels



Performance level	Change lead time	Deployment frequency	Change fail rate	Failed deployment recovery time	Percentage of respondents*
Elite	Less than one day	On demand (multiple deploys per day)	5%	Less than one hour	19% (18-20%)
High	Between one day and one week	Between once per day and once per week	20%	Less than one day	22% (21-23%)
Medium	Between one week and one month	Between once per week and once per month	10%	Less than one day	35% (33-36%)
Low	Between one month and six months	Between once per month and once every six months	40%	Between one week and one month	25% (23-26%)

*89% uncertainty interval

Observability and Monitoring

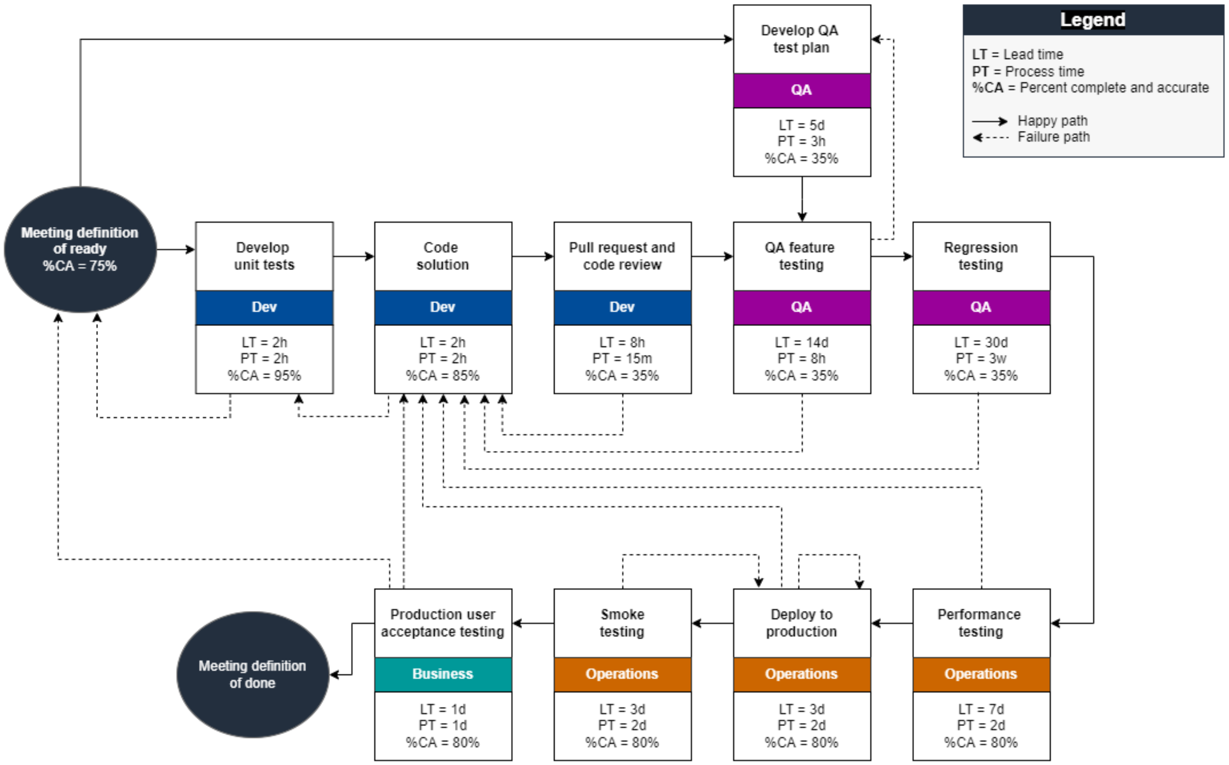
- To measure the DORA metrics or other metrics, monitoring of run code and infrastructure
- Use metrics and insights to improve the value stream

Value Stream Mapping

- Value Stream Management: sequence of activities to deliver value; makes the flow of actions to deliver value visible
- Value Stream Mapping: identify and analyze value streams aiming to reduce or eliminate inefficiencies

https://www.atlassian.com/agile/value-stream-management

Development Value Stream Map (DVSM)



Legend

LT = Lead time
PT = Process time
%CA = Percent complete and accurate

→ Happy path
⇠ Failure path

Meeting definition of ready
%CA = 75%

Develop unit tests
Dev
LT = 2h
PT = 2h
%CA = 95%

Code solution
Dev
LT = 2h
PT = 2h
%CA = 85%

Pull request and code review
Dev
LT = 8h
PT = 15m
%CA = 35%

Develop QA test plan
QA
LT = 5d
PT = 3h
%CA = 35%

QA feature testing
QA
LT = 14d
PT = 8h
%CA = 35%

Regression testing
QA
LT = 30d
PT = 3w
%CA = 35%

Performance testing
Operations
LT = 7d
PT = 2d
%CA = 80%

Deploy to production
Operations
LT = 3d
PT = 2d
%CA = 80%

Smoke testing
Operations
LT = 3d
PT = 2d
%CA = 80%

Production user acceptance testing
Business
LT = 1d
PT = 1d
%CA = 80%

Meeting definition of done

https://docs.aws.amazon.com/prescriptive-guidance/latest/strategy-devops-value-stream-mapping/introduction.html

Continuous Integration and Deployment

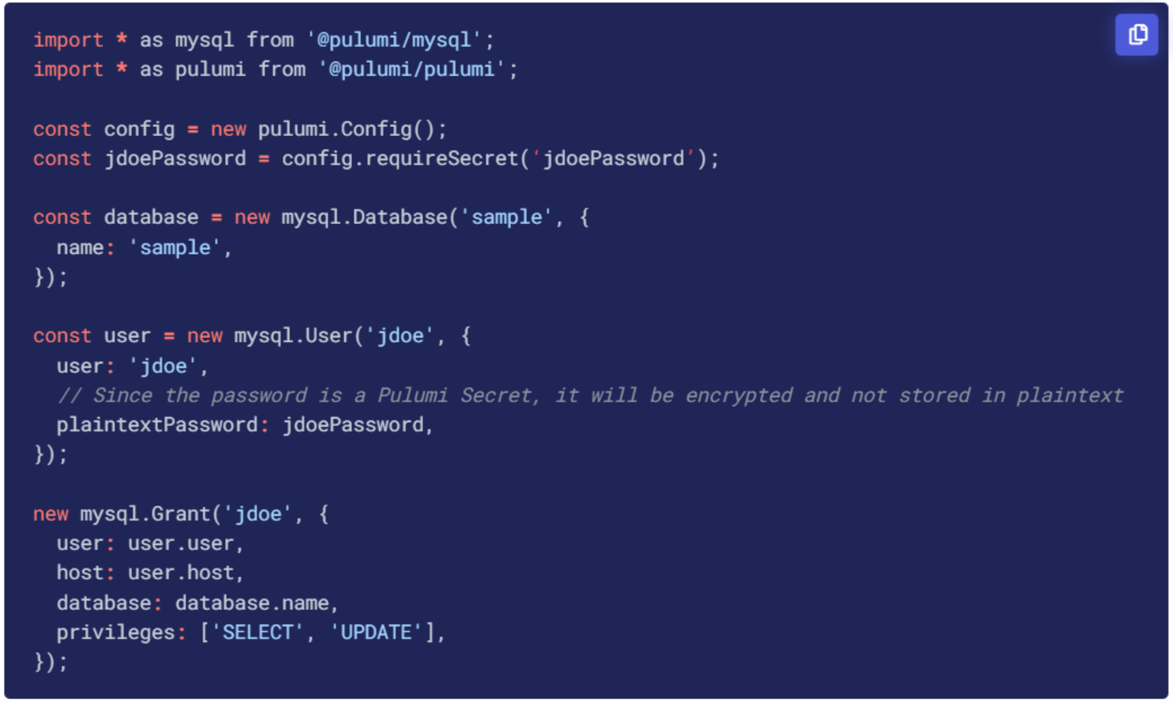
- Continuous Integration and Continuous Deployment (CI/CD): frequently integrate changes, test them, and automatically deploy them
  - Requires automated tests
- Automated way to push changes to production!

Infrastructure as Code

- Infrastructure as Code: configuration of the server infrastructure should be treated similar to code
- Configuration files or scripts
- Applying best practices such as version control and CI/CD

Infrastructure as Code

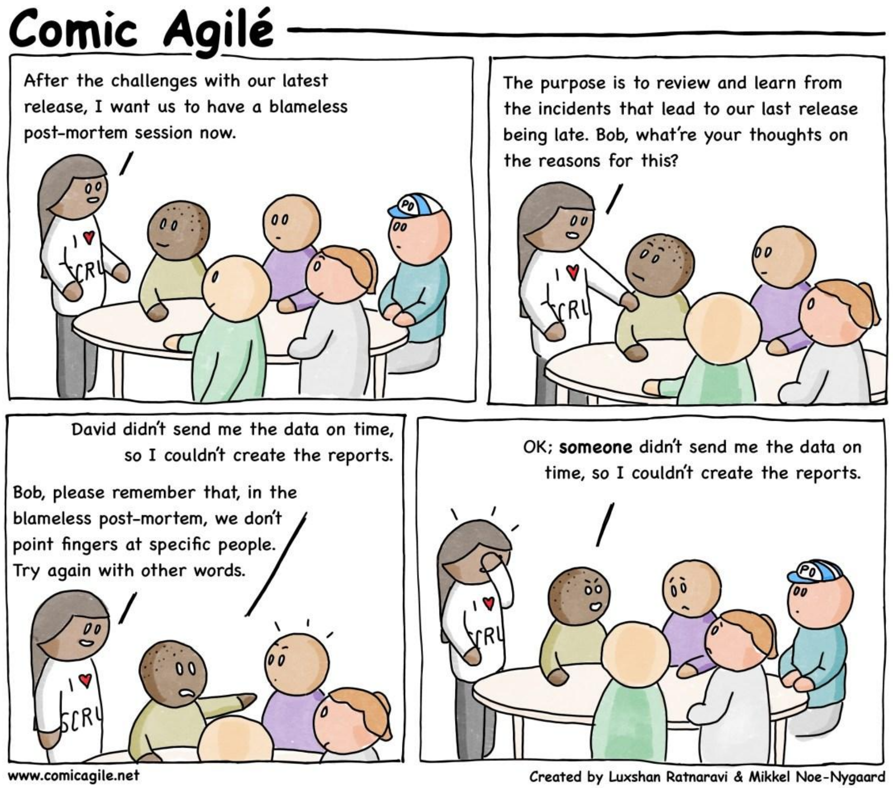
Pulumi



```
import * as mysql from '@pulumi/mysql';
import * as pulumi from '@pulumi/pulumi';

const config = new pulumi.Config();
const jdoePassword = config.requireSecret(‘jdoePassword’);

const database = new mysql.Database('sample', {
  name: 'sample',
});

const user = new mysql.User('jdoe', {
  user: 'jdoe',
  // Since the password is a Pulumi Secret, it will be encrypted and not stored in plaintext
  plaintextPassword: jdoePassword,
});

new mysql.Grant('jdoe', {
  user: user.user,
  host: user.host,
  database: database.name,
  privileges: ['SELECT', 'UPDATE'],
});
```

https://www.pulumi.com/blog/managing-your-mysql-databases-with-pulumi/

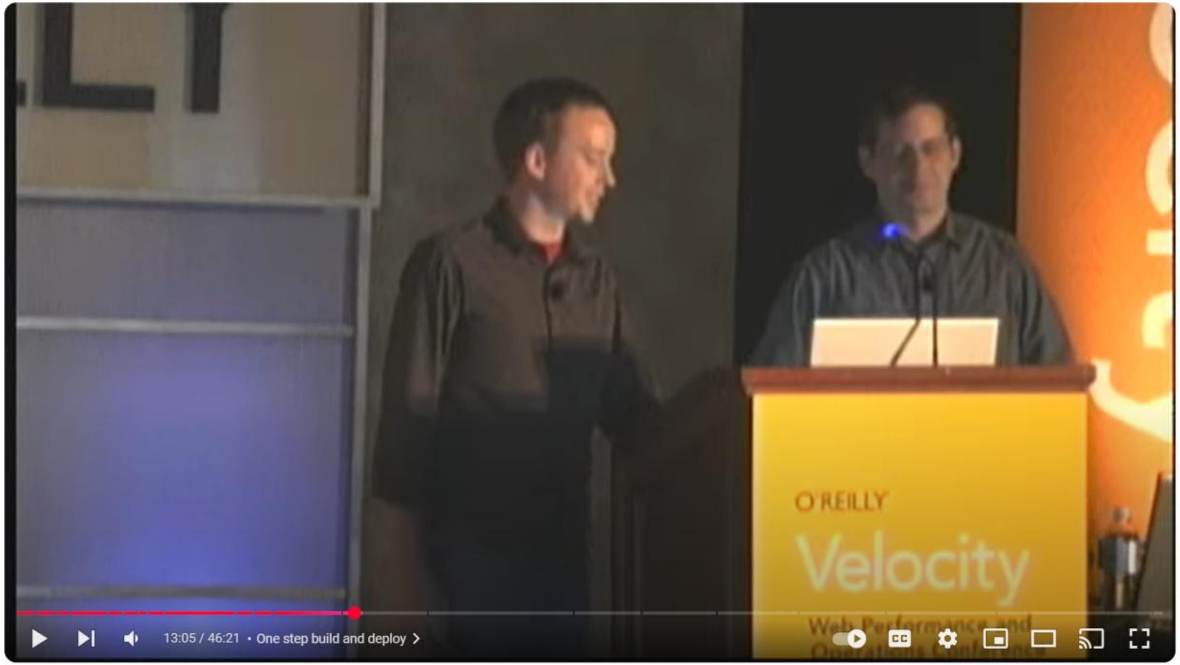
Learning from Incidents: Blameless Postmortems


Comic Agilé
After the challenges with our latest release, I want us to have a blameless post-mortem session now.
The purpose is to review and learn from the incidents that lead to our last release being late. Bob, what're your thoughts on the reasons for this?
David didn't send me the data on time, so I couldn't create the reports.
Bob, please remember that, in the blameless post-mortem, we don't point fingers at specific people. Try again with other words.
OK; **someone** didn't send me the data on time, so I couldn't create the reports.
I ♥ SCRUM
PO
www.comicagile.net
Created by Luxshan Ratnaravi & Mikkel Noe-Nygaard

Summary and Important Points

- Challenge: how to put increments into production? The challenge of silos.
- DevOps: The whole organization works together to deliver value, in particular, development and operation teams.
- Key principles such as flow, feedback, and continuous learning and improvement.
- Making DevOps measurable through DORA (change lead time, deployment frequency, change fail rate, failed deployment recovery time)
- Key practices, such as monitoring, infrastructure as code, and learning from incidents

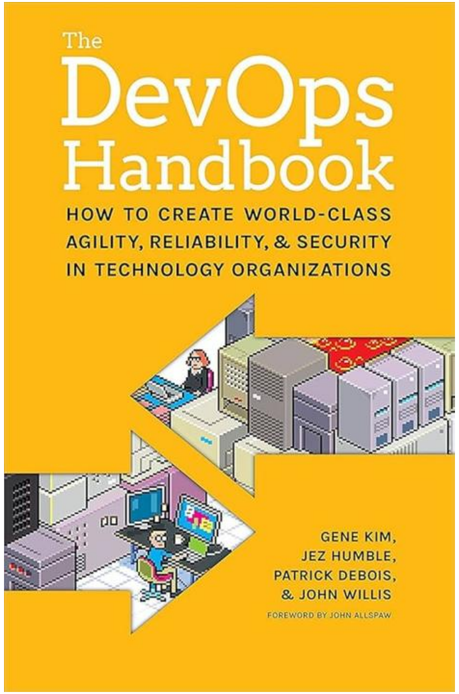
“10+ Deploys per Day: Dev and Ops Cooperation at Flickr” (2009)



Velocity 09: John Allspaw and Paul Hammond, "10+ Deploys Pe

https://www.youtube.com/watch?v=LdOe18KhtT4

The DevOps Handbook



The DevOps Handbook

HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS

GENE KIM,
JEZ HUMBLE,
PATRICK DEBOIS,
& JOHN WILLIS

FOREWORD BY JOHN ALLSPAW