

Course Introduction

Chris Timperley · Monday, August 24, 2026

Hi, I’m Chris

Associate Teaching Professor · S3D

I teach software engineering through real codebases, collaborative projects, and authentic engineering practice.

I build and lead complex software systems, especially in robotics and program repair.

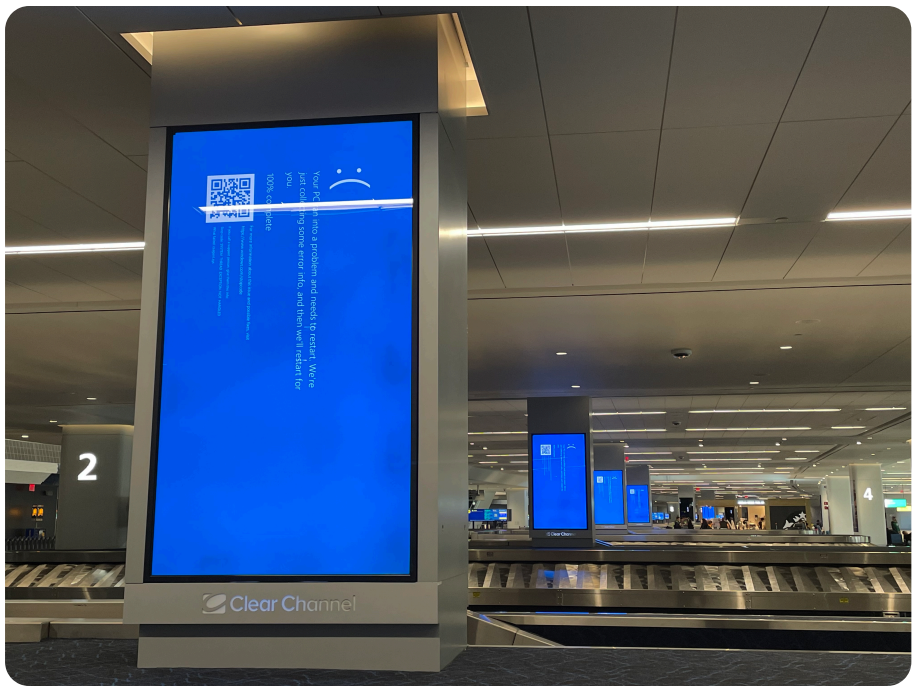
I study how developers change software and how we can make those changes safer and more effective.



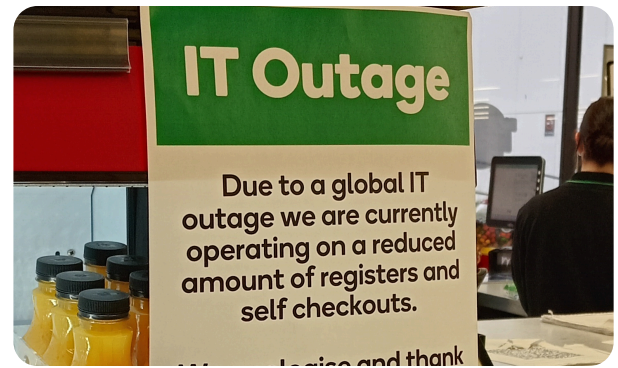
THE COMMON THREAD | **Making complex software safer and easier to change.**

Why DevOps?

CrowdStrike (2024)



LaGuardia Airport, July 19, 2024 · Smishra1 / Wikimedia Commons (CC BY-SA 4.0)



Palmerston North, New Zealand · Kiwi128 / Wikimedia Commons (CC0)

8.5 million

Windows devices affected—less than 1% of the installed base.

Microsoft incident response

What would you do?

With a neighbour, describe:

1. **one technical control** that might have limited the incident;
2. **one piece of feedback** that responders needed; and
3. **one organizational decision** that affected how far the incident spread

How one defect became a system-wide failure



01
Change
21 fields supplied

02
Validate
Mismatch **escaped**

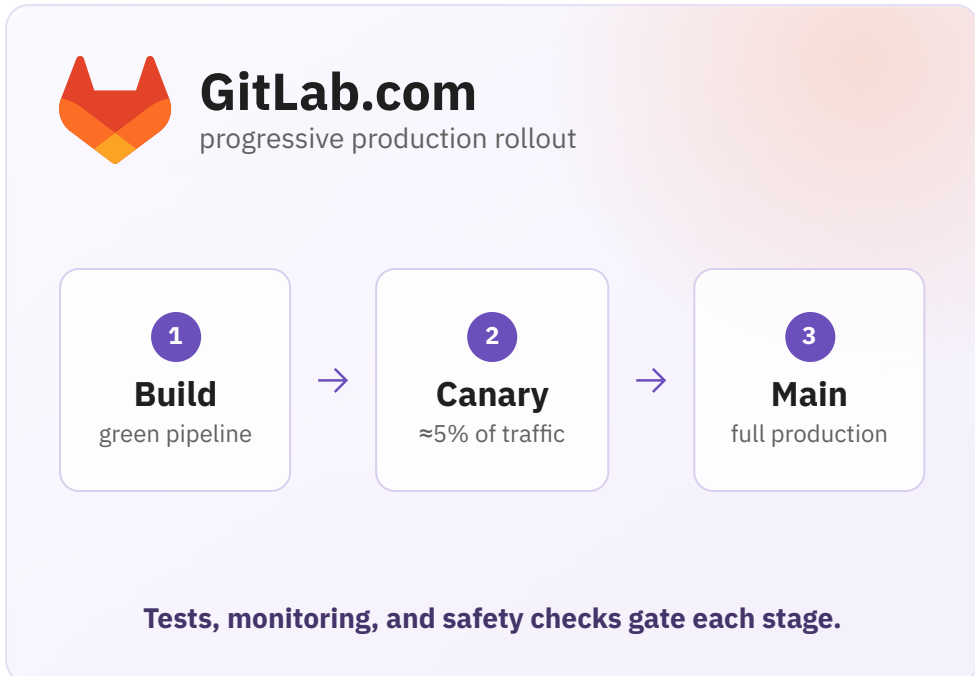
03
Roll out
Broad reach

04
Detect
Crashes became the **signal**

05
Recover
Manual repair

Read the Executive Summary of the Root Cause Analysis

Production change can become routine



GitLab.com deployment path · GitLab Engineering (2025)

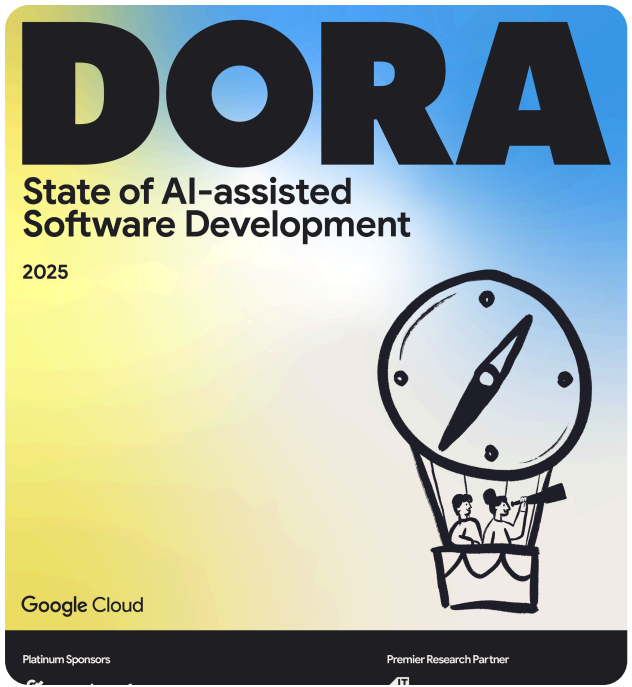
Up to 12 deploys/day

Canary stages, safety checks, and compatibility rules make frequent change zero-downtime and rollback-ready.

GitLab Engineering

Automation amplifies whatever the system already knows: good constraints or bad assumptions.

How does AI affect software delivery?



DORA · State of AI-assisted Software Development · 2025

> “AI’s primary role is as an **amplifier**, magnifying an organization’s existing strengths and weaknesses.
>
> — DORA · 2025

What becomes scarce as code output rises?

> "Developers generate significantly more code with AI assistants, but most delivery processes were built for lower, manual volumes."



Matthias Patzak
AWS Executive in Residence
AWS blog

aws

> "The right place for us humans is to build and manage the working loop..."



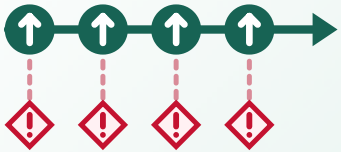
Kief Morris
Thoughtworks
MartinFowler.com

thoughtworks

THE DEVOPS QUESTION | **Can delivery and feedback keep pace?**

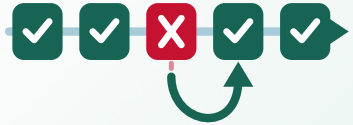
Vote: Must a team choose between speed and safety?

A



Shipping more often creates more opportunities to fail

B



Shipping more often can make each failure smaller and easier to recover

C



The question is missing something important

Software delivery is a feedback loop: change, run, observe, decide

INTENT — PRODUCTION — DECISION



01	02	03	04	05
Commit	**Build + verify**	**Release + run**	**Observe**	**Next change**
State the **change**	**Test** assumptions	Put it into **operation**	See what **happened**	Act on the **evidence**

This course studies how teams deliver changes, observe their effects, and decide what to change next.

What is DevOps?

r/devops keeps asking...

r/devops • 9y ago

Can someone explain what DevOps is?

u/Seref15

"Developers create new bugs and I make sure they are delivered to our infrastructure in an extremely timely and non-interruptive fashion."

r/devops • 2y ago

What is DevOps, Really?

u/fire-d-guy

"No one knows what it means, we all just parrot the phrase because everyone else does."

r/devops • 3y ago

How do you explain DevOps to non-technical people?

u/air-

"Computer janitor."

r/devops • 7mo ago

What is DevOps? (Discussion)

u/unitegondwanaland

"DevOps is what the hiring manager at the company you work at tells you it is."

One question. Many meanings.

How would you define DevOps?



Historically, Dev and Ops had different goals



DEVELOPMENT

Deliver change

Complete features, start new work, and hand changes off.

OPERATIONS

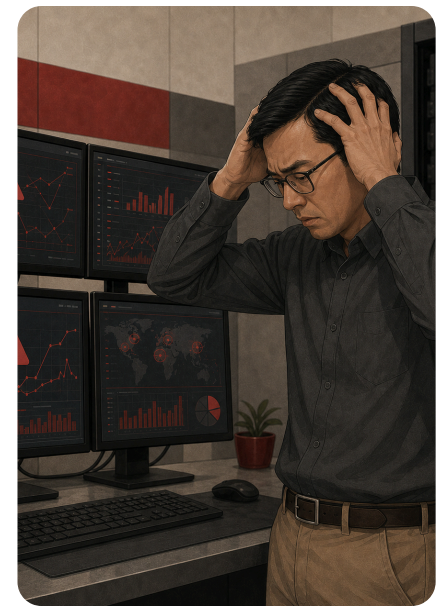
Protect stability

Prevent disruption, preserve service, and control change.

Each team made reasonable choices based on its own goals, but neither was responsible for the overall result.

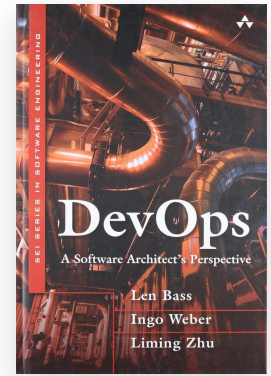
What gets lost at the handoff?

- **Hidden assumptions:** production differs from test
- **Long queues:** changes wait for release windows
- **Weak feedback:** failures reach developers late
- **Split ownership:** incidents bounce between teams



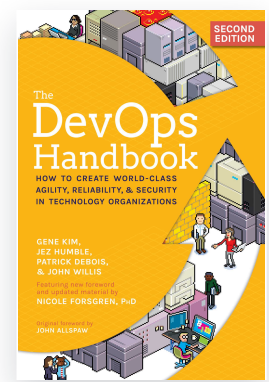
DevOps is a set of practices intended to **reduce the time between committing a change** to a system and the change being placed into normal production, while **ensuring high quality**.

Bass, Weber & Zhu · 2015



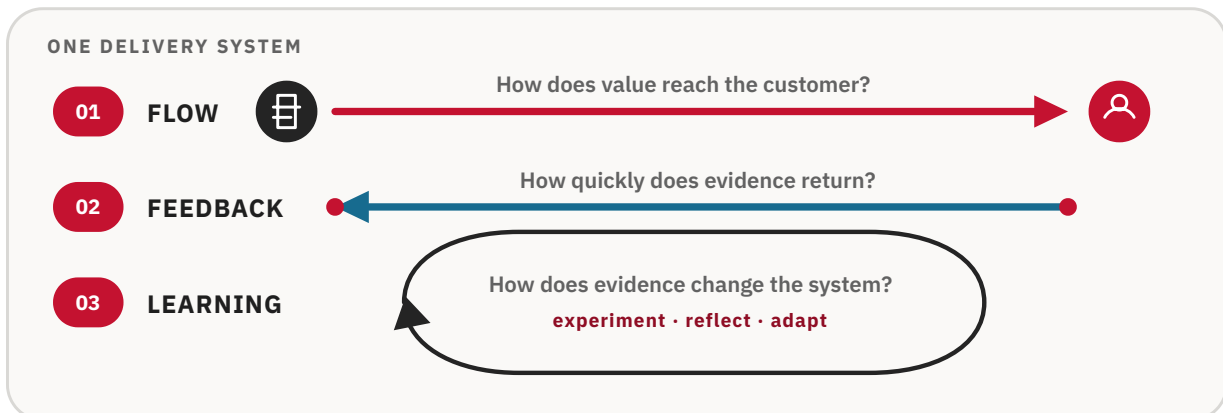
DevOps is the outcome of applying the most trusted **principles** from the domain of physical manufacturing and leadership **to the IT value stream**.

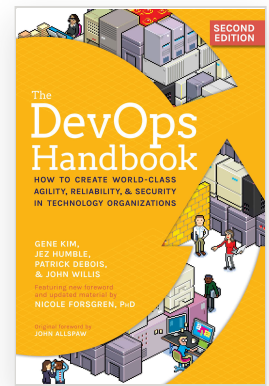
Kim, Humble, Debois, Willis & Forsgren · 2021



DevOps is the engineering of a fast, safe learning system for software delivery.

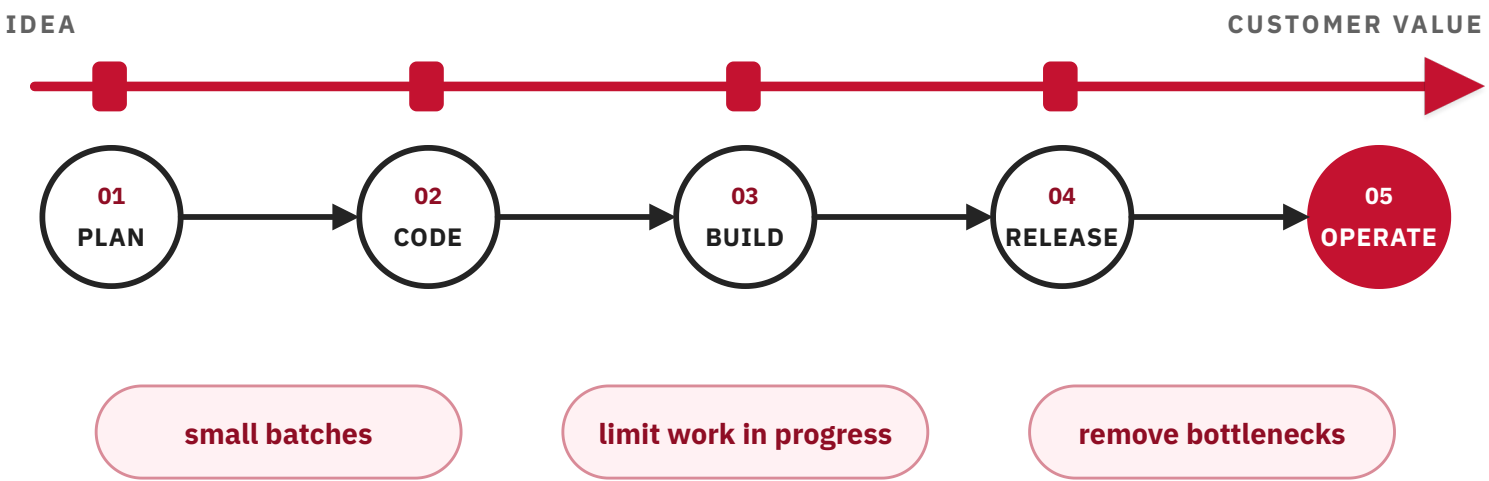
DevOps is a Philosophy



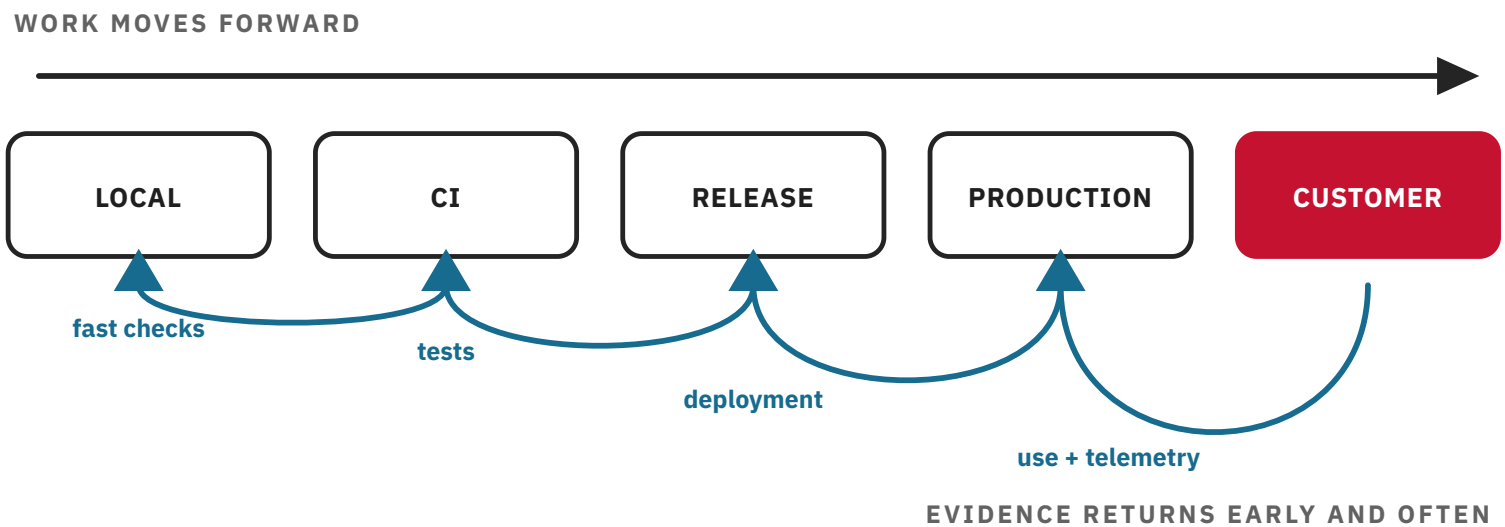


Kim, Humble, Debois, Willis & Forsgren · 2021

The First Way: accelerate flow from idea to customer

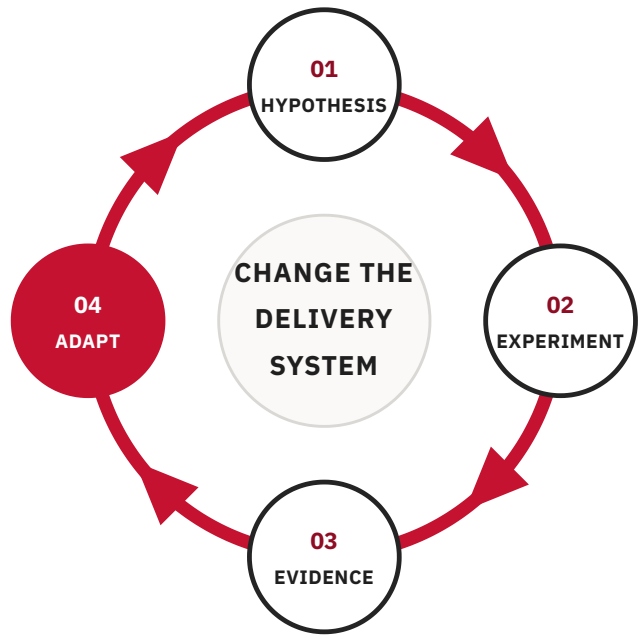


The Second Way: amplify fast feedback at every stage

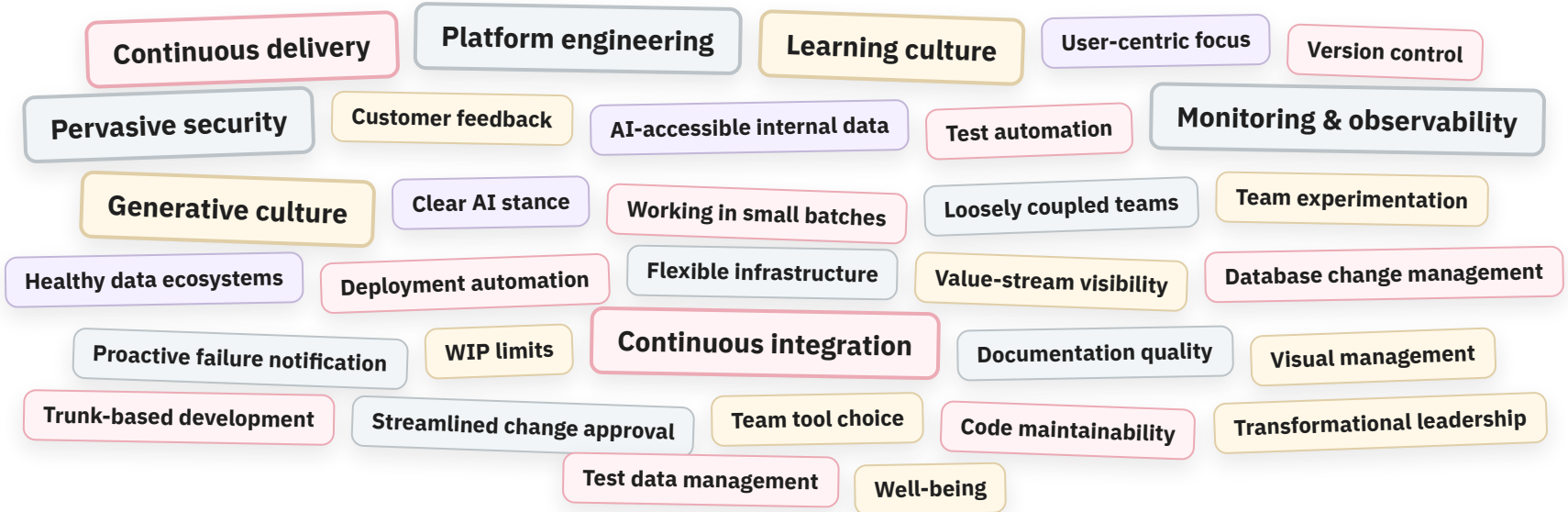


The Third Way: turn evidence into continual learning

Feedback becomes learning **when behavior changes.**

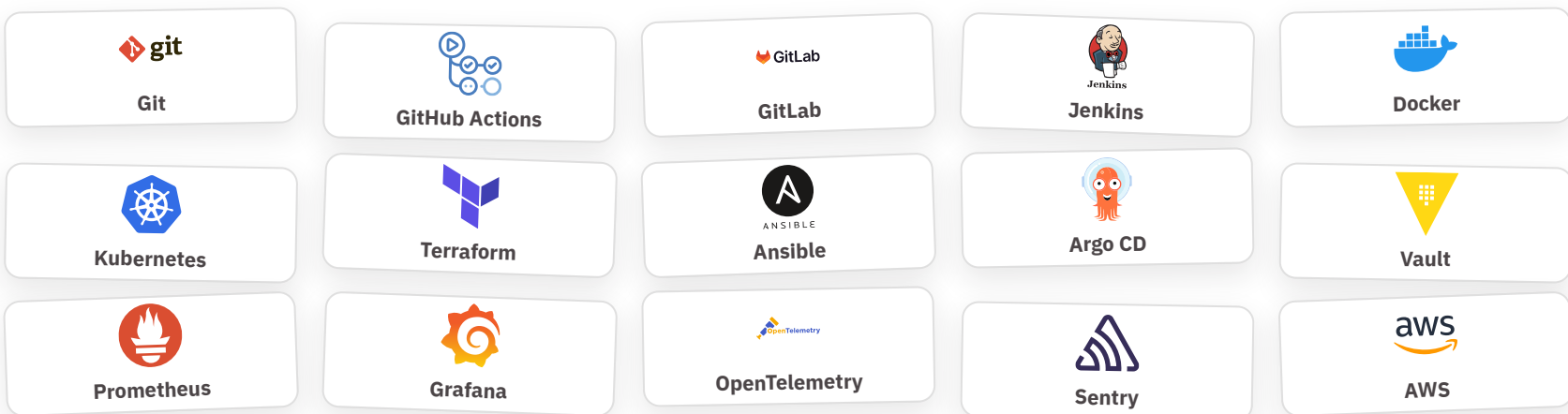


DevOps is a Set of Practices



Selected from the DORA capability catalog.

DevOps is Implemented via Tools

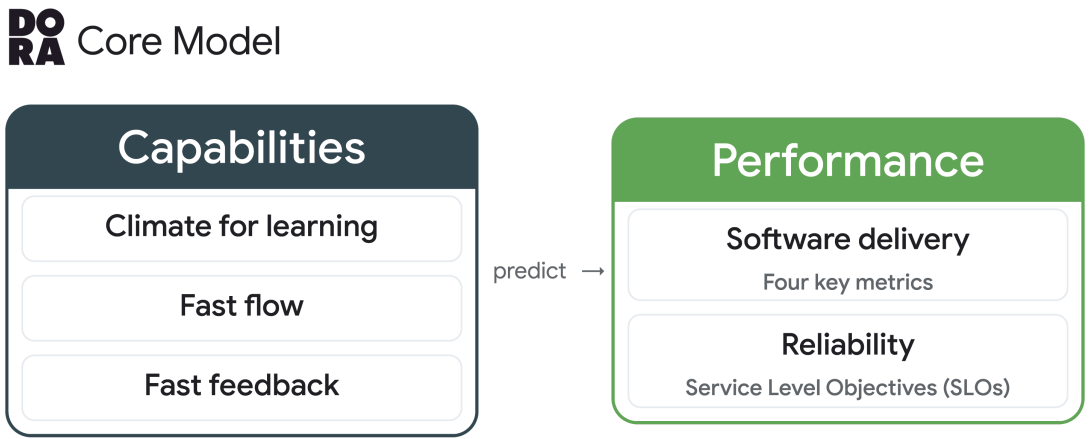


DevOps is not about the tools.

How do you know the Three Ways are working?



Capabilities predict performance



DORA Core Model v2.1.0 · Google LLC · CC BY 4.0

MAXIMIZE

Throughput

How quickly does change move through the system?

Change lead time
Time from a change committed to version control to its deployment in production.

Deployment frequency
Deployments per period, or the time between deployments.

Failed deployment recovery time
Time to recover from a deployment that requires immediate intervention.

MINIMIZE

Instability

How often does change create more work?

Change fail rate
Share of deployments that require immediate intervention.

Deployment rework rate
Share of unplanned deployments caused by a production incident.

USER-FACING RELIABILITY

Service Level Objective

A target level of reliability for a service's users.

Indicator
Which user-visible events count as good?

Target
What proportion of events must be good?

Window
Over what period is the target evaluated?

SHOPIFY · CHECKOUT LATENCY

99.9% of checkout attempts complete within 2 seconds over a rolling 28 days.

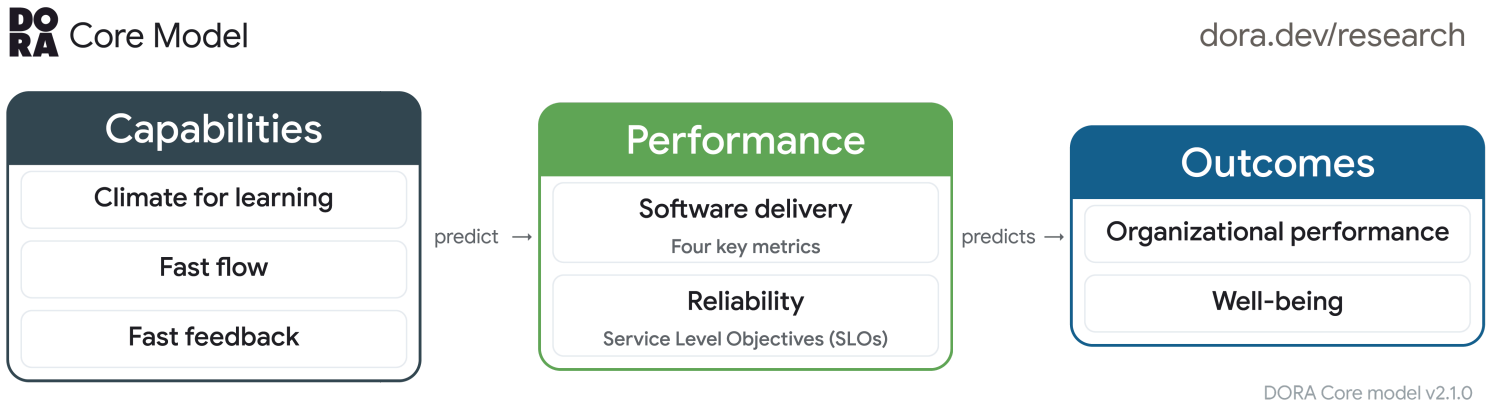
NETFLIX · PLAYBACK QUALITY

99% of viewing minutes play in HD without rebuffering over a rolling 28 days.

SPOTIFY · LIBRARY FRESHNESS

99% of saved tracks appear in a user's library within 10 seconds over a rolling 28 days.

Performance predicts outcomes



DORA Core Model v2.1.0 · Google LLC · CC BY 4.0

Organizational performance

Mission and business goals · product quality · customer satisfaction · operating efficiency

Well-being

Job satisfaction · productivity · reduced burnout · reduced rework

Course Overview

The course spans software delivery, operations, and continuous improvement



Build

Create repeatable delivery environments

- containers and orchestration
- source control
- platform engineering
- harness engineering



Verify

Check each change before it ships

- CI and automated testing
- secrets, keys, and auth
- dependencies, vulnerabilities, and versions
- DevSecOps in operations



Release

Move changes into production safely

- release engineering
- CD and environments
- GitOps
- flags, canaries, and shadow traffic



Operate

Run production safely and reliably

- logs, metrics, and traces
- dashboards and action
- reliability and resilience
- recovery and learning



Improve

Use evidence to improve the system

- experimentation
- evolutionary code and APIs
- evolutionary data
- people, teams, and outcomes

Lectures introduce concepts, labs build skills, and assignments integrate them



Lecture

Reason about delivery systems from **principles and examples**

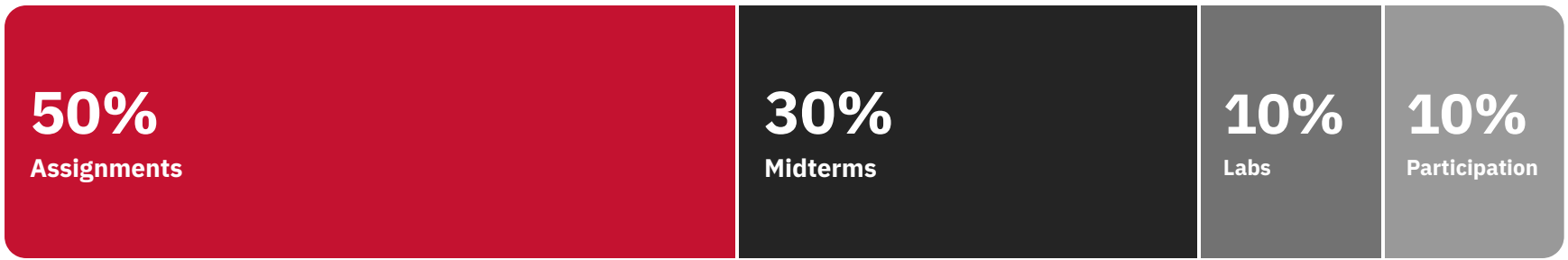
Lab

Practice **one capability** in a focused exercise

Assignment

Combine capabilities and **make trade-offs** under constraints

Assignments account for half of the course grade



Assignments
Seven cumulative deliverables

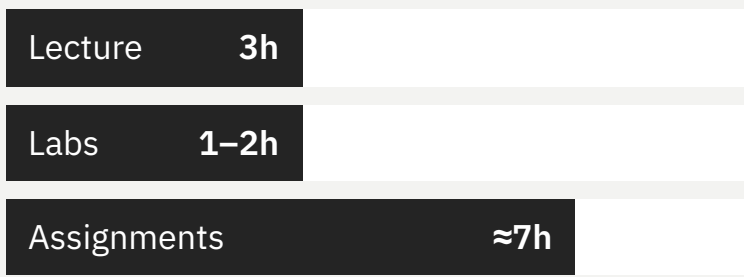
Midterms
Two paper exams · 15% each

Labs
Focused practice · pass/fail

Participation
In-class activities

A typical week takes about 12 hours

≈12 hours / week



4 penalty-free late days

- shared across graded checkpoints
- each started 24 hours uses one
- after four, each extra day costs 10%
- submissions close after 96 hours
- fixed operational cutoffs are marked

You are responsible for everything you submit

Understand
every line and every decision

Write
the substance of reports yourself

Disclose
where and how AI tools helped

Own
correctness, security, and quality

ChatGPT

Claude

GitHub Copilot

Gemini

Perplexity

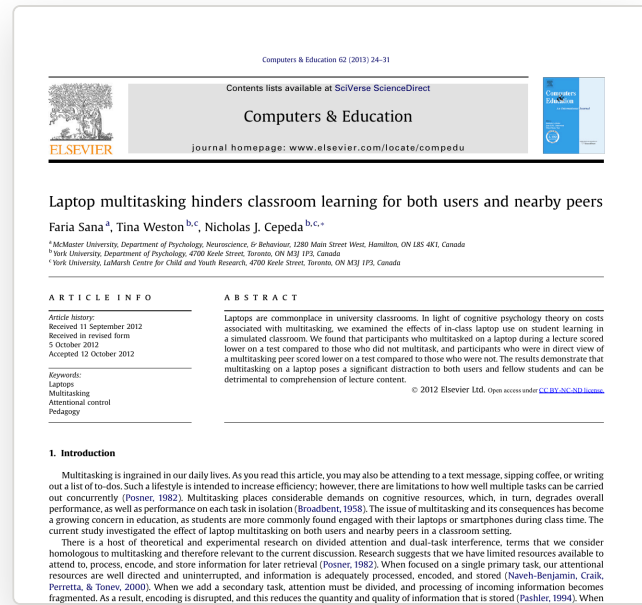
Mistral

Laptop multitasking can hinder learning for users and nearby peers

Laptops away
unless you are seated in the back row

Tablets are OK
for taking notes

Planned laptop use
we will tell you when to bring them out



Computers & Education 62 (2013) 24–31

Contents lists available at SciVerse ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

Laptop multitasking hinders classroom learning for both users and nearby peers

Faria Sana [a], Tina Weston [b,c], Nicholas J. Cepeda [b,c,*]

[a] McMaster University, Department of Psychology, Neuroscience, & Behaviour, 1280 Main Street West, Hamilton, ON L8S 4K1, Canada
[b] York University, Department of Psychology, 4700 Keele Street, Toronto, ON M3J 1P3, Canada
[c] York University, LaMarsh Centre for Child and Youth Research, 4700 Keele Street, Toronto, ON M3J 1P3, Canada

ARTICLE INFO

Article history:
Received 11 September 2012
Received in revised form
5 October 2012
Accepted 12 October 2012

Keywords:
Laptops
Multitasking
Attentional control
Pedagogy

ABSTRACT

Laptops are commonplace in university classrooms. In light of cognitive psychology theory on costs associated with multitasking, we examined the effects of in-class laptop use on student learning in a simulated classroom. We found that participants who multitasked on a laptop during a lecture scored lower on a test compared to those who did not multitask, and participants who were in direct view of a multitasking peer scored lower on a test compared to those who were not. The results demonstrate that multitasking on a laptop poses a significant distraction to both users and fellow students and can be detrimental to comprehension of lecture content.

© 2012 Elsevier Ltd. Open access under CC BY-NC-ND license.

1. Introduction

Multitasking is ingrained in our daily lives. As you read this article, you may also be attending to a text message, sipping coffee, or writing out a list of to-dos. Such a lifestyle is intended to increase efficiency; however, there are limitations to how well multiple tasks can be carried out concurrently (Posner, 1982). Multitasking places considerable demands on cognitive resources, which, in turn, degrades overall performance, as well as performance on each task in isolation (Broadbent, 1958). The issue of multitasking and its consequences has become a growing concern in education, as students are more commonly found engaged with their laptops or smartphones during class time. The current study investigated the effect of laptop multitasking on both users and nearby peers in a classroom setting.

There is a host of theoretical and experimental research on divided attention and dual-task interference, terms that we consider homologous to multitasking and therefore relevant to the current discussion. Research suggests that we have limited resources available to attend to, process, encode, and store information for later retrieval (Posner, 1982). When focused on a single primary task, our attentional resources are well directed and uninterrupted, and information is adequately processed, encoded, and stored (Naveh-Benjamin, Craik, Perretta, & Tonev, 2000). When we add a secondary task, attention must be divided, and processing of incoming information becomes fragmented. As a result, encoding is disrupted, and this reduces the quantity and quality of information that is stored (Pashler, 1994). When

Sana, Weston, and Cepeda · Computers & Education · 2013

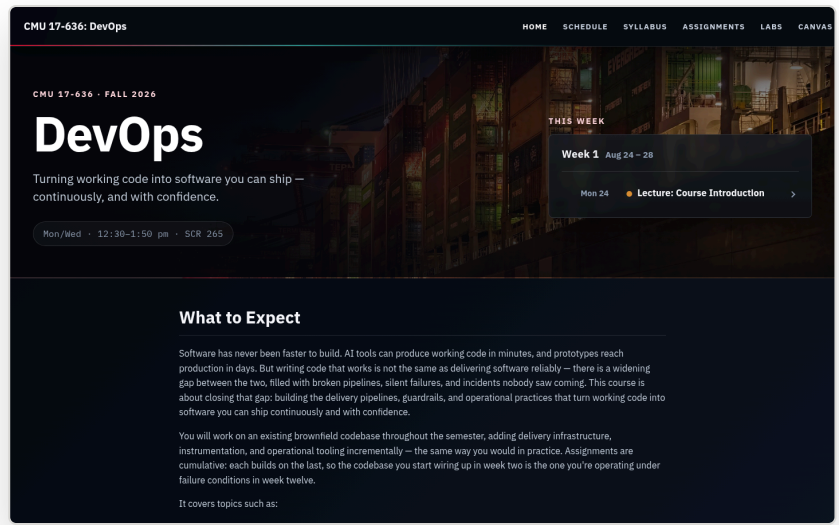
Lectures are highly interactive and participation is a key part of your experience

10%

in-class activities

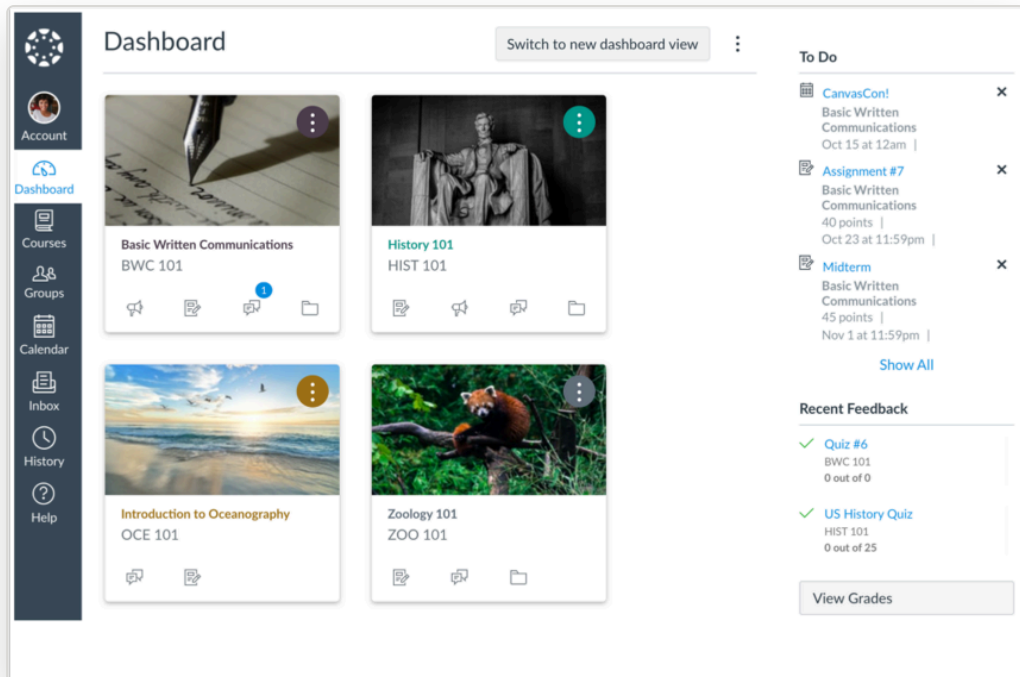
In class	Expect frequent whole-class discussions and small-group breakout activities.
Miss a class	Four absences are ignored automatically; missed activities are not made up.
Need support	For extended absences, contact course staff and involve your academic advisor early.

Everything you need to know about the course is on the website

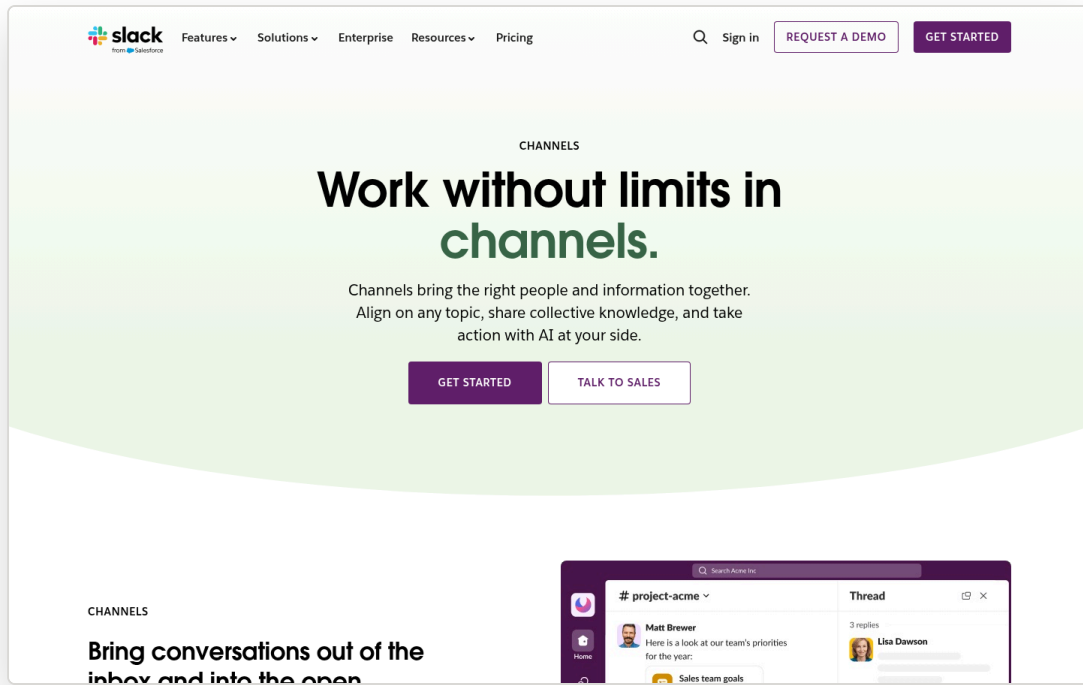


cmu-devops.github.io

Submit work, see grades, and request regrades in Canvas



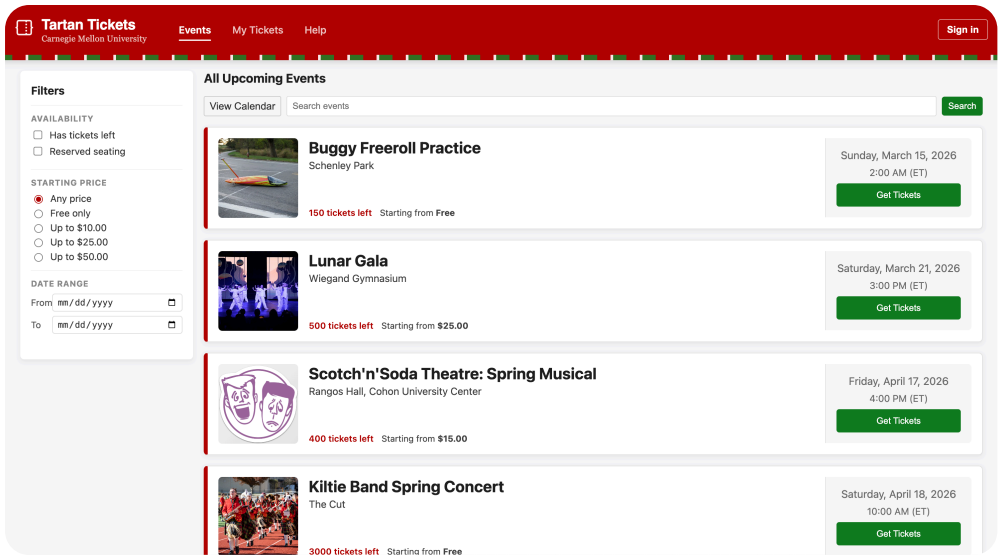
Use Slack for announcements, questions, and coordination



Contact course staff for appointments, support, and exceptional circumstances



Tartan Tickets



Tartan Tickets

- a university events ticketing platform
- a browser application and API gateway
- multiple services, databases, and a payment provider
- existing behavior, data, dependencies, and defects

Your job is to make the system safer to change, operate, and recover.

Assignments

ESTABLISH CONFIDENCE

1. Build a safety net
2. Add CI guardrails
3. Respond to a secrets incident

→

DELIVER CHANGE

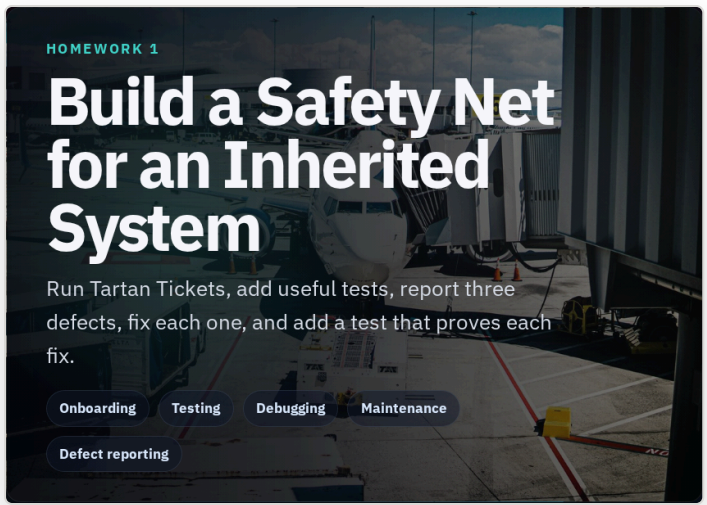
4. Build a CD pipeline
5. Release with evidence
6. Evolve without downtime

→

OPERATE UNDER PRESSURE

7. Run the game day

HW1: Build a Safety Net



Two graded checkpoints

A — Starting Point and Defect Reports
Sunday, August 30 · 11:59 p.m. ET

B — Fixes and Safety Net
Sunday, September 6 · 11:59 p.m. ET

https://cmu-devops.github.io/assignments/hw01

Getting Started

Before next time:

1. Read the syllabus and current schedule on the course website.
2. Join the Slack workspace (see Canvas announcement).
3. Complete the Google Form with your GitHub ID.
4. Get access to your repository and try to launch the DevContainer.
5. Post any issues to **#technical-support** on Slack.

🚀 Today, we learned how to ...

- Explain DevOps as the engineering of a fast, safe learning system for software delivery
- Use flow, feedback, and continual learning to diagnose a delivery problem
- Interpret the five DORA metrics as performance signals rather than targets
- Describe the course rhythm, assessments, and cumulative project