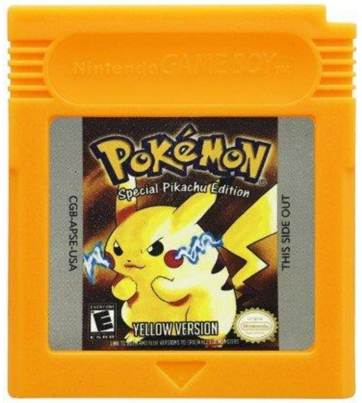
CSci 3081W: Program Design and Development

Lecture 15 – Deployment

History

Black box (ex. physical ATM)

Cartridges, cassettes, floppy disks, CD/DVD, flash drives



History

Internet – enabled users to be able to download software without a hard copy (as well as installation, updates, etc)

Deployment

General process that can be modified

Activities within the deployment process

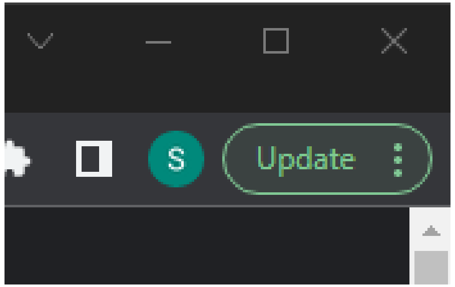
Can be handled by producer, consumer, or both

Deployment

Who deploys the software?

Depends!
Examples:

- The user clicks update
- DevOps: deployment/release coordinator(s)



Deployment Environment

Also known as “tiers”

A system or set of subsystems where a program or software is deployed and executed

In simple cases, you could have just a single environment

In industrial use/complex cases, the development environment and production environment are separated
- development environment - where the changes are made originally
- production environment - what the clients or users end up using
- usually there are stages in between
- allows phased deployment (aka rollout), testing, and rollback in case of bugs/issues

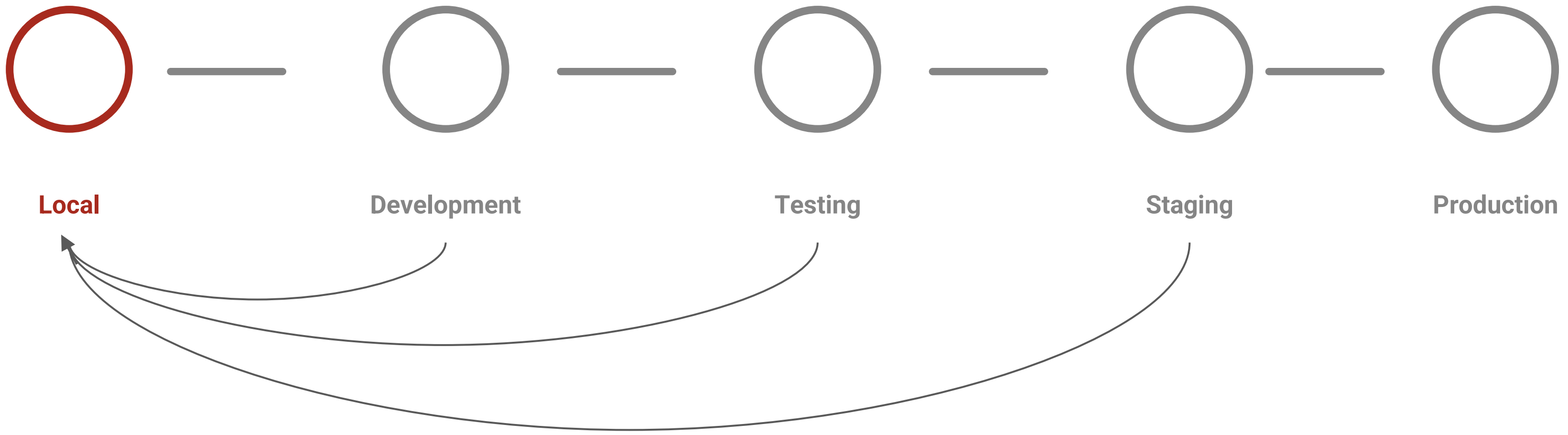
Deployment Environments

Environment Name	**Description**
Local	Developer's desktop/workstation
Development	Development server acting as a sandbox where unit testing may be performed by the developer
Testing	The environment where interface testing is performed
Staging/Pre-production/External-Client Acceptance	Mirror of production environment
Production	Serves end-users/Client

Deployment Environments

Local Environment

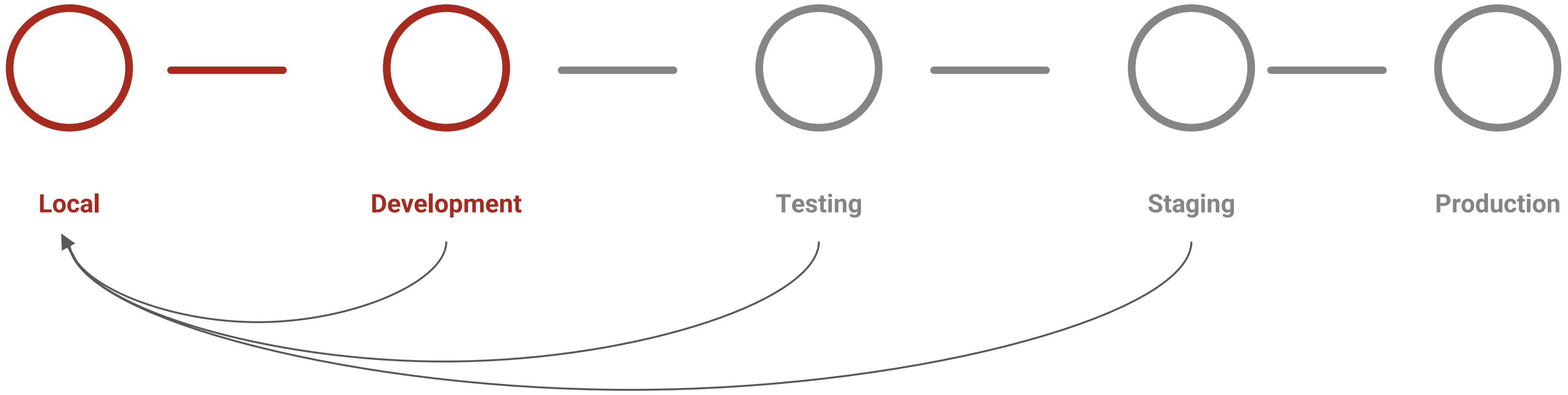
- **The place where developers code the features assigned to them for the sprint.**
- **Usually the place where your IDE exists.**



Deployment Environments

Development Environment

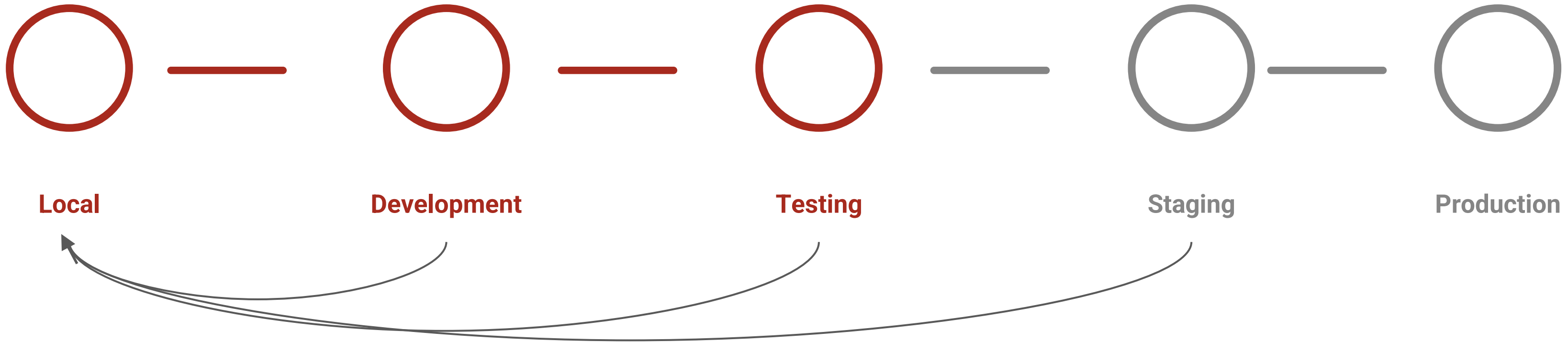
- **Colloquially known as ‘dev’ environment.**
- **A ‘sandbox’ environment where code is deployed to servers such that unit testing can be performed.**
- **For example: if there’s 10 features being developed simultaneously, then there should ideally be 10 different dev environments such that we can see how each feature behaves individually.**



Deployment Environments

Testing Environment

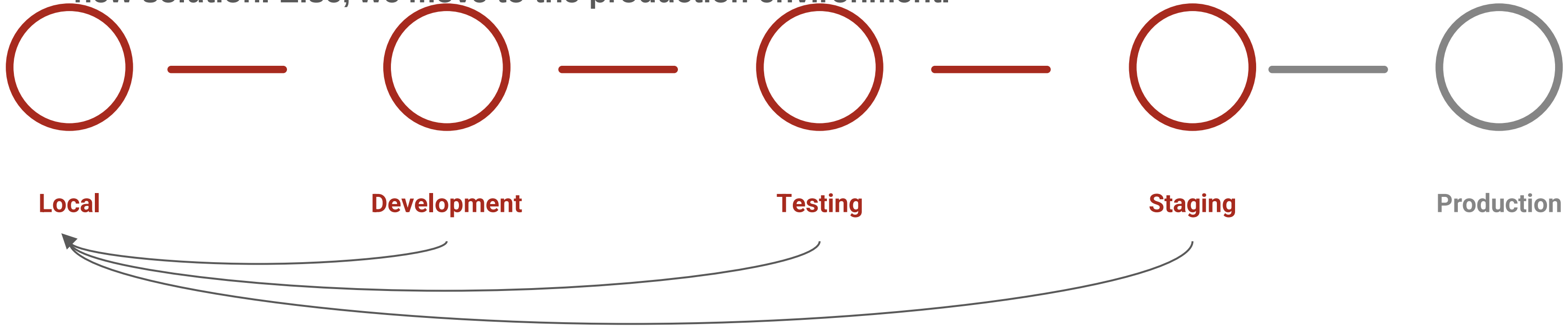
- **The environment where interface testing is performed**
- **All the features that were developed individually are combined into one ‘release’ so that their effects can be scrutinized together by the QA (Quality assurance) team by testing different workflows as well as checking how things work together at scale**
- **For eg: the 10 different features developed in the last sprint are all merged together in Github and their combination is now deployed to a testing environment where the QA team makes sure that none of the existing features are impacted by the changes made as well as none of the changes interfere with other changes made during the same sprint.**



Deployment Environments

Staging Environment

- **This environment imitates the production environment.**
- **The ‘release’ is deployed to the staging environment so that clients can check if the changes/features look okay.**
- **Integral part of the Agile process.**
- **If you aren’t developing for a specific client, it’s still good to see how your release behaves in an environment that best resembles production**
- **Eg: The features developed over the last sprint are now shown to the client where they can green flag/red flag changes based on their liking. If a change gets red flagged we go back to the local environment to develop a new solution. Else, we move to the production environment.**

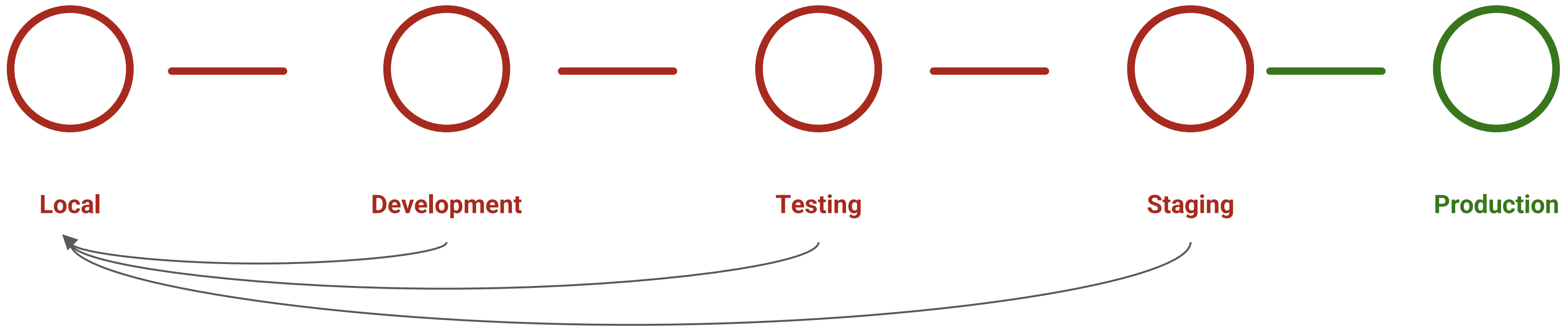


Deployment Environments

Production Environment

- **The actual environment where code is run such that it generates revenue for the company.**
- **Outages/mistakes often impact revenue, hence the previous environments in this process.**
- **Eg: In our day to day lives we only interact with production environments. www.youtube.com, www.gmail.com, etc. are all production environments.**

The entire process of running the changes on all the different environments usually happens over a week. This is after the developer has already coded the changes/features.



Hotfixes

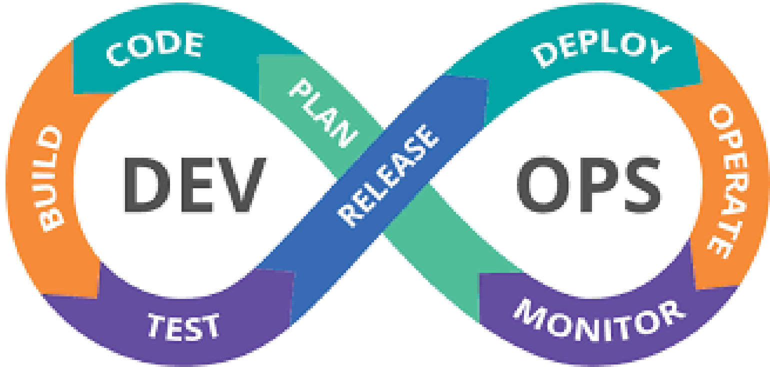
Problem:
Imagine that you’re in production and there’s an important bug that needs to be fixed immediately

Solution:

- local
- testing
- production

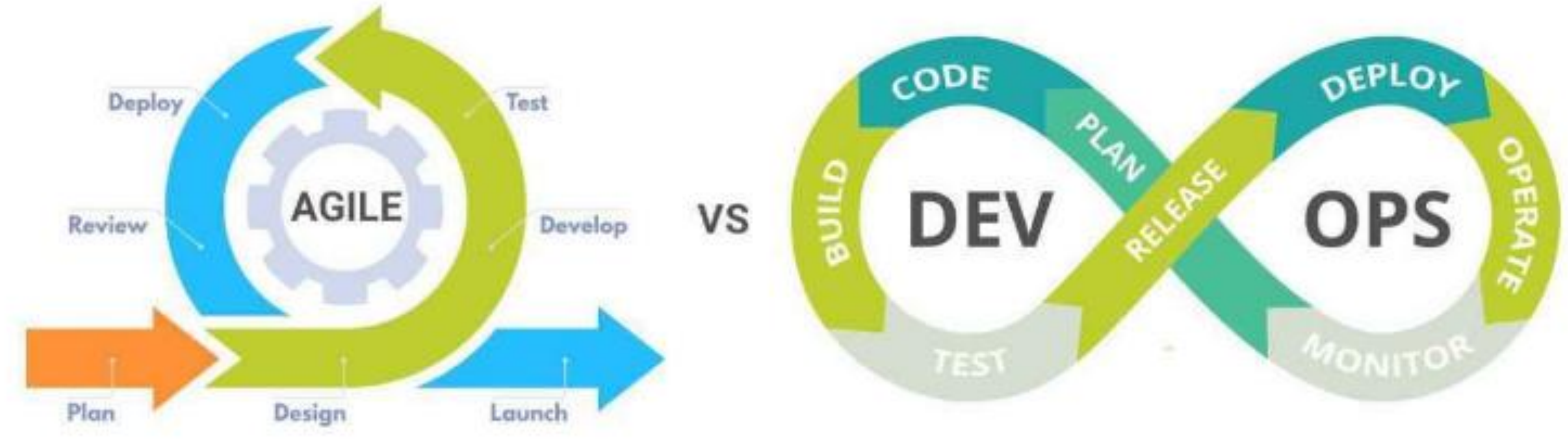
Intro to DevOps

- Partnership between Development and Operations teams (dev and ops)
  - Enables code deployment and production to be more rapid and automated
  - Before DevOps, these teams worked independently from one another
- Principles: flow, feedback, and continuous learning



DevOps and Agile

- Not mutually exclusive

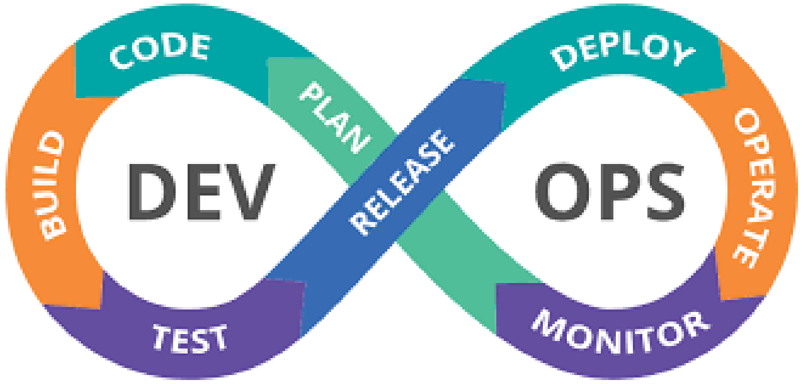


DevOps and Agile

	DevOps	Agile
Philosophy and Focus	- Rooted in stability, consistency, and planning - Seeks to identify new ways to improve and streamline processes - Focuses on maximizing efficiency, identifying programmable processes, and increasing automation	- Centers around adaptability and keeping pace with customer needs and expectations - Features are described as user stories, placing the focus on the individual user, what they need, and why
Scope	- Represents the intersections of development, operations, and quality assurance - Cross-disciplinary teams unite and collaborate in the development and delivery of software	-Specific to the development team, its productivity, and its progress toward completing the project at hand -Development is completed in incremental sprints, and software delivery, deployment, or ongoing maintenance of each release is managed by different teams
Manifestations	- Continuous integration - Continuous delivery - Continuous deployment	- Scrum - Kanban - Extreme programming - and others (crystal, lean, DSDM, etc.)

Additional Reading

- DevOps:
  - https://itrevolution.com/product/the-devops-handbook-second-edition/
  - https://learn.microsoft.com/en-us/azure/devops/?view=azure-devops



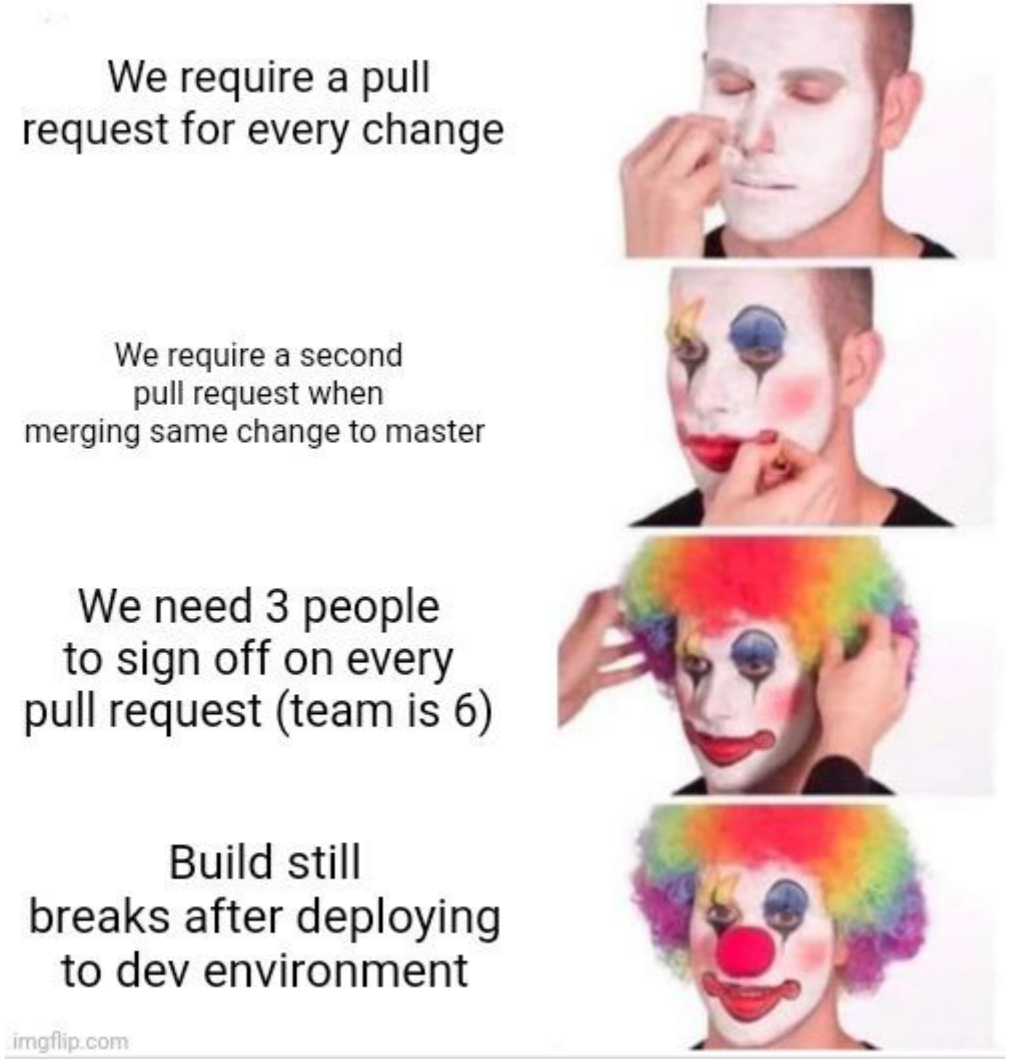
Deployment

MY COWORKERS WATCHING ME DEPLOY A "SMALL FIX" ON A FRIDAY
Get ready everyone, he's about to do something stupid.

Deployment

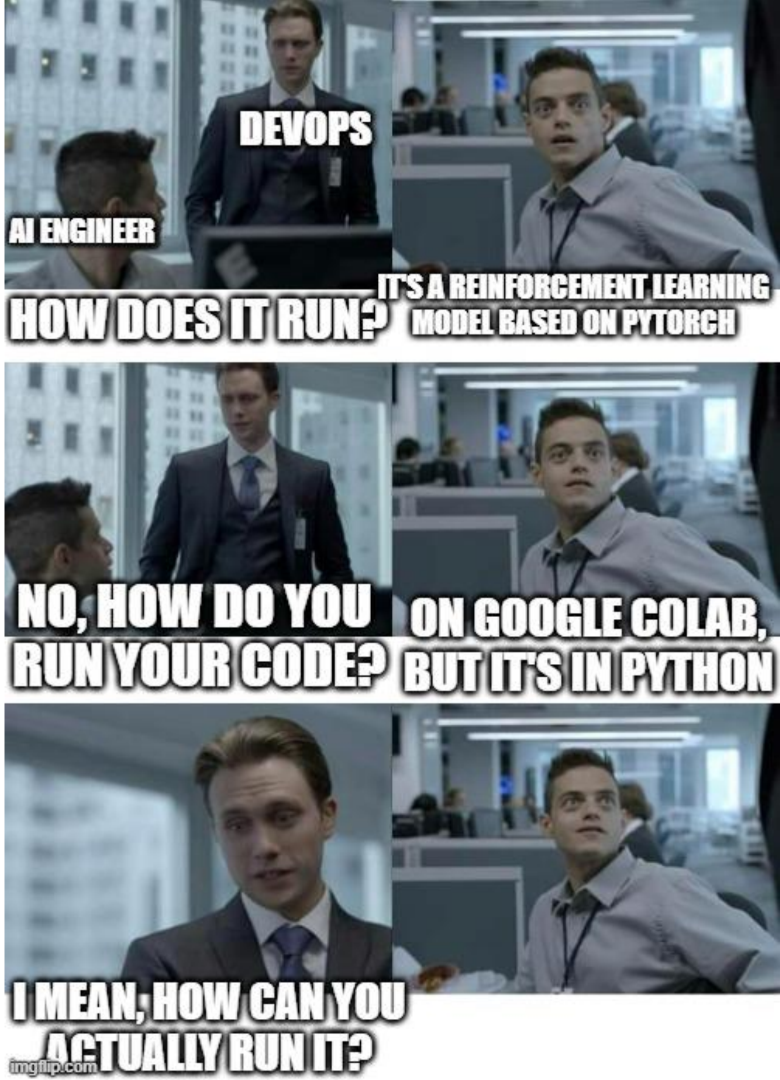
DEPLOYING ON A FRIDAY
ON CHRISTMAS EVE
IT'S JUST A SMALL CHANGE
NOTHING COULD GO WRONG
imgflip.com

Deployment


We require a pull request for every change
We require a second pull request when merging same change to master
We need 3 people to sign off on every pull request (team is 6)
Build still breaks after deploying to dev environment
imgflip.com

DevOps

Software Team
gifmemes.io


DEVOPS
AI ENGINEER
HOW DOES IT RUN?
IT'S A REINFORCEMENT LEARNING MODEL BASED ON PYTORCH
NO, HOW DO YOU RUN YOUR CODE?
ON GOOGLE COLAB, BUT IT'S IN PYTHON
I MEAN, HOW CAN YOU ACTUALLY RUN IT?
imgflip.com

Quiz 4 Practice Problems