Software Engineering

Session 10

Today

Lab

- DevOps/Continuous integration/github actions intro
- STAR stories intro
- Marta from careers (11am)

Seminar

- Sprint 4 planning + tasks
- Complete lab work

DevOps and continuous integration

- What is devOps?

Dev + Ops = DevOps

Automating the testing and deployment of code to increase speed and reliability

DevOps

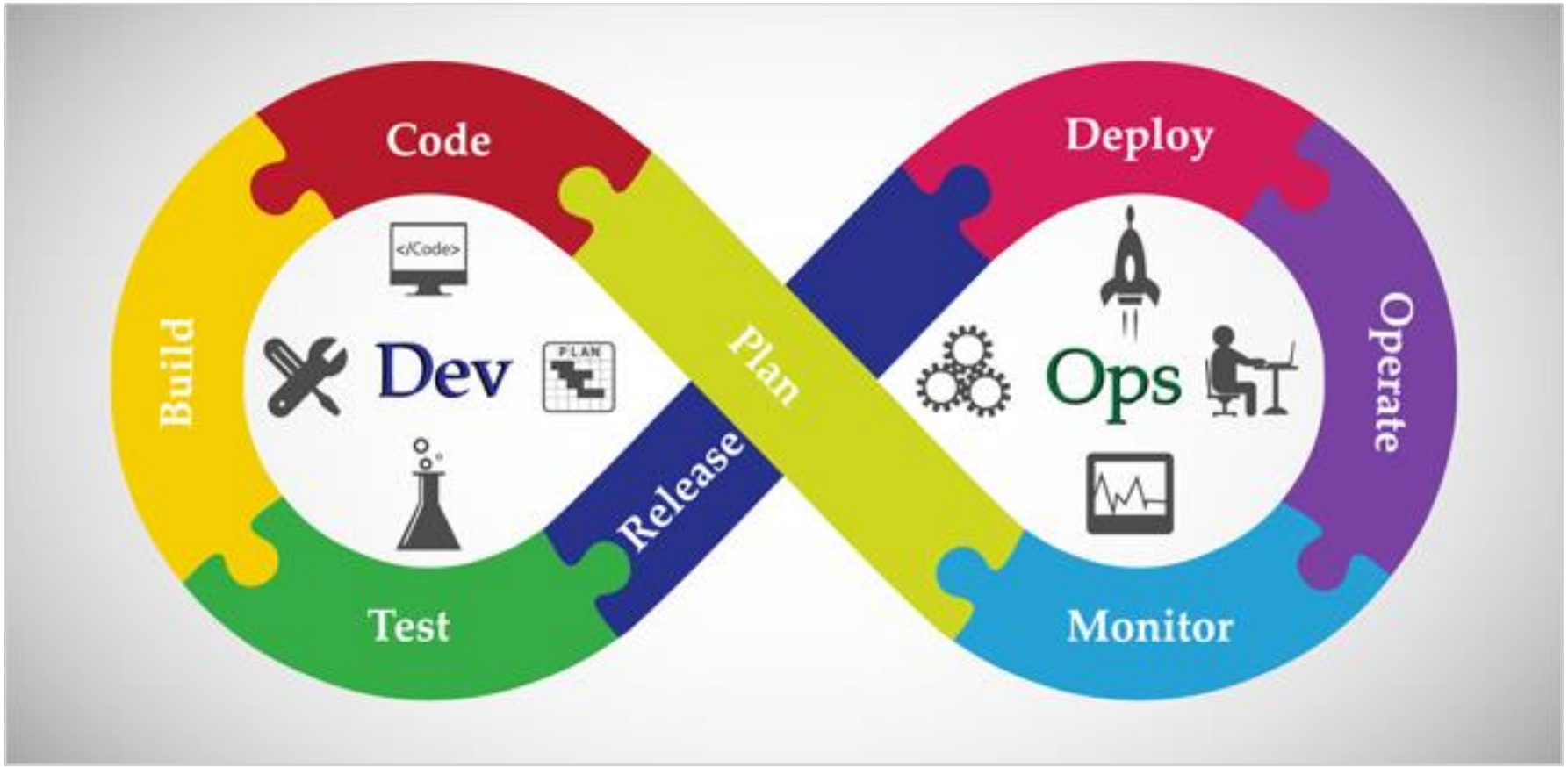
DevOps is a combination of software development (dev) and operations (ops). It is defined as a software engineering methodology which aims to integrate the work of development teams and operations teams by facilitating a culture of collaboration and shared responsibility.

It emphasizes collaboration, **automation**, **integration** and rapid feedback cycles. These characteristics help ensure a culture of building, testing, and releasing software that is **more reliable and at a high velocity**.

Why DevOps?

- Continuous delivery of software (needed for Agile)
- Better collaboration between teams
- Easy deployment (so users get benefits)
- Better efficiency and scalability
- Errors are fixed at the initial stage
- Less manual intervention (which means fewer chances of error and better security)

DevOps tools – there are many!



DevOps Tools

Summary: core DevOps principles

1. **Automation of the software development lifecycle.** This includes automating testing, builds, releases, the provisioning of development environments, and other manual tasks that can slow down or introduce human error into the software delivery process.
2. **Collaboration and communication.** A good DevOps team has automation, but a great DevOps team also has effective collaboration and communication.
3. **Continuous improvement and minimization of waste.** From automating repetitive tasks to watching performance metrics for ways to reduce release times or mean-time-to-recovery, high performing DevOps teams are regularly looking for areas that could be improved.
4. **Hyperfocus on user needs with short feedback loops.** Through automation, improved communication and collaboration, and continuous improvement, DevOps teams can take a moment and focus on what real users really want, and how to give it to them.

https://about.gitlab.com/topics/devops/

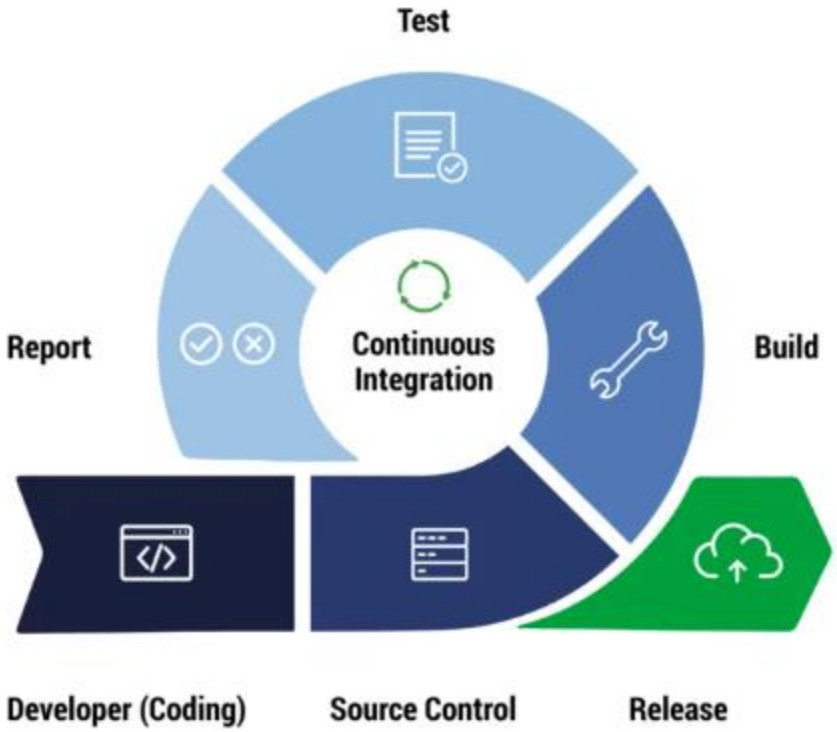
Our focus: Continuous Integration

- Continuous integration is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.
- This is to ensure that the software is always in a workable state
- Continuous integration is a key part of continuous delivery in which software updates are rolled out frequently
- Required for 'agile' software development to be effective

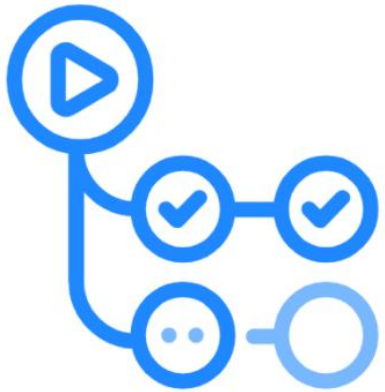
Continuous Integration

Requires:

- Code changes detected
- Integrated build triggered
- Tests run
- Report

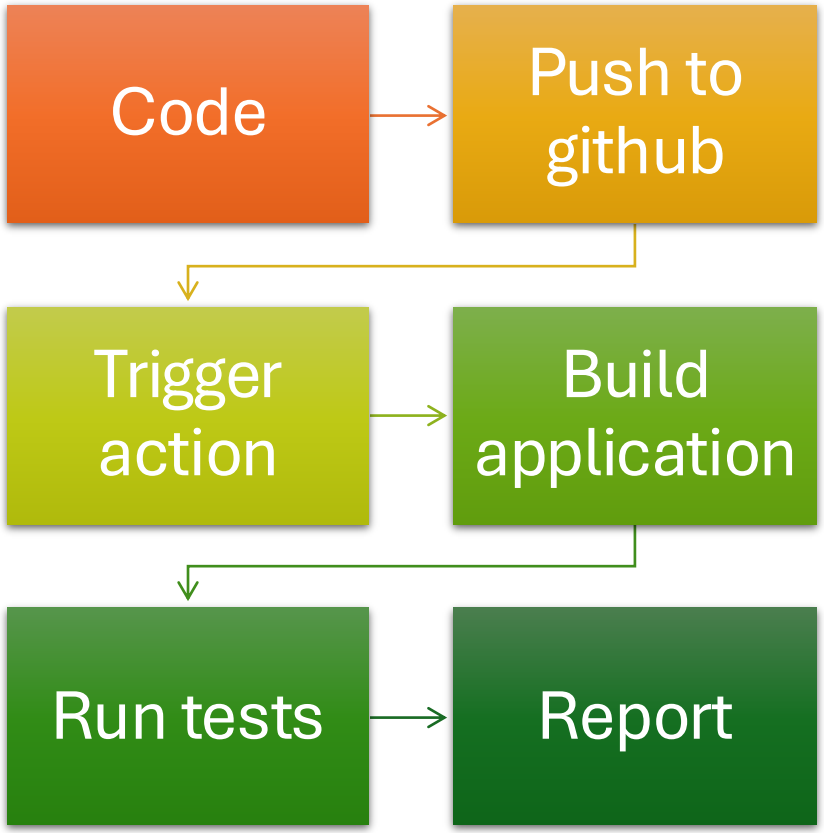


Github actions



A CI/CD platform that allows you to automate software development workflows, enabling tasks like building, testing, and deploying applications directly from your GitHub repository.

Our solution



Github actions: key terms

- **How it works:**
  - **Workflows:** You define workflows using YAML files, specifying the events that trigger them and the steps to be executed.
  - **Jobs:** Workflows consist of one or more jobs, which can run sequentially or in parallel.
  - **Steps:** Each job contains one or more steps that run a script or an action.
  - **Runners:** are virtual machines that run your instructions
  - **Secrets** are credentials required that are shared securely in github and used in your actions

https://docs.github.com/en/actions/about-github-actions/understanding-github-actions

Simple github actions yml

```yaml
# This is a basic workflow that is triggered by a push to the deploy branch

name: On push to deploy branch

# Controls when the action will run. Triggered by push to deploy branch
on:
  push:
    branches:
      - deploy

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
  # This workflow contains a single job called "greet"
  greet:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest

    # Steps represent a sequence of tasks that will be executed as part of the job
    steps:
    # Runs a single command using the runners shell
    - name: Send message
      run: echo "The deploy branch has been updated"
```

LAB…

- (look at the lab sheet)

Individual assignment: STAR stories

The STAR method (Situation, Task, Action, Result) is a commonly used way to evaluate candidates during job interviews. It is comprised of 4 steps:

- Situation: Describe the situation and when it took place.
- Task: Explain the task and what was the goal.
- Action: Provide details about the action you took to attain this.
- Result: Conclude with the result of your action.

STAR INTERVIEW TECHNIQUE

Example: Can you share a time when you've had to juggle multiple priorities at work?

Situation: While working as a client success manager at a tech company, one of my colleagues left the company for a new opportunity. My manager asked me to take on some of her responsibilities.

Task: I had to reprioritize my own clients and projects to make room for her most important ones. It was overwhelming at first, with so many tasks to juggle and my unfamiliarity with my colleague's book of business.

Action: I worked through my responsibilities and reprioritized them based on the company's goals, my availability, and other factors (with a bit of input from my manager). I also came up with ways to automate certain tasks to free up more of my time.

Result: Thanks to new automation efforts and successful prioritizing, none of our clients realized that there had been an internal shift at the company. Our team's high quality of service was maintained — and I became more efficient in the process.

Your questions (choose 1)

- Tell me about a time when there was a disagreement in your team.
- Tell me about a time when you were stuck on a task.
- Tell me about a time when your project seemed to be stuck in a rut and not moving
- ahead at the required pace.
- Tell me about a time you were concerned that a deadline might be missed.
- Tell me about a time when you had to correct a colleague's work.
- Tell me about a time when you had to challenge the actions of someone in your
- team.
- Tell me about a time when you made a wrong decision
- Tell me about a time when you had to shift priorities

- Things to remember:

- You must write about a real situation that you faced during your coursework.
- Do not use AI for this assignment. If you hand in an answer that is generic and which
- I cannot recognize from the work that I know you have done, your work will not be
- marked.
- You should not name names in your group or apportion blame. If you need to refer
- to something somebody else has done, please just refer to ‘a group member’.
- You should be able to tell your STAR story from a positive perspective. Even if things
- went wrong, a good STAR story can show that you took effective action to resolve a
- difficult situation.