An Intro to Tech Jobs

Just what is "software engineering"?

Not any more!

- Much more collaboration
- More accessible tools
- Wider range of technology
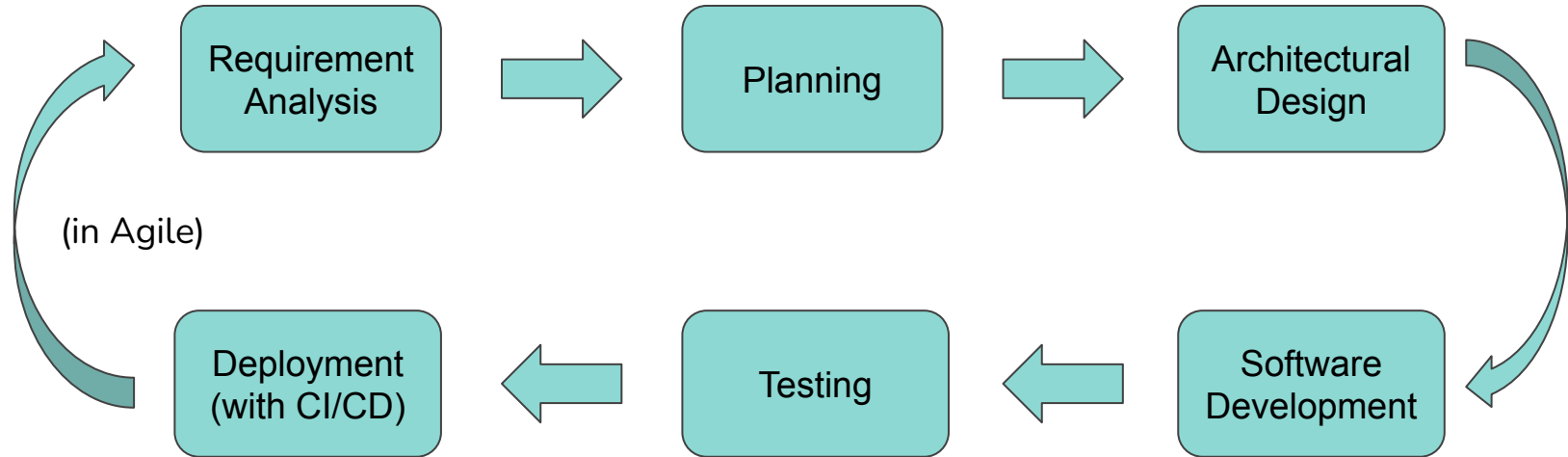- Much bigger ecosystem

But first – The Software Development Life Cycle! (SDLC)

- A process that produces the highest quality and lowest cost software in the shortest time.
- Needs a model for actual delivery - Agile, Waterfall, Iterative, Spiral
- All tech jobs are integrated into it

Steps of the SDLC

```
Requirement          →    Planning    →    Architectural
Analysis                                   Design

(in Agile)

Deployment    ←    Testing    ←    Software
(with CI/CD)                       Development
```
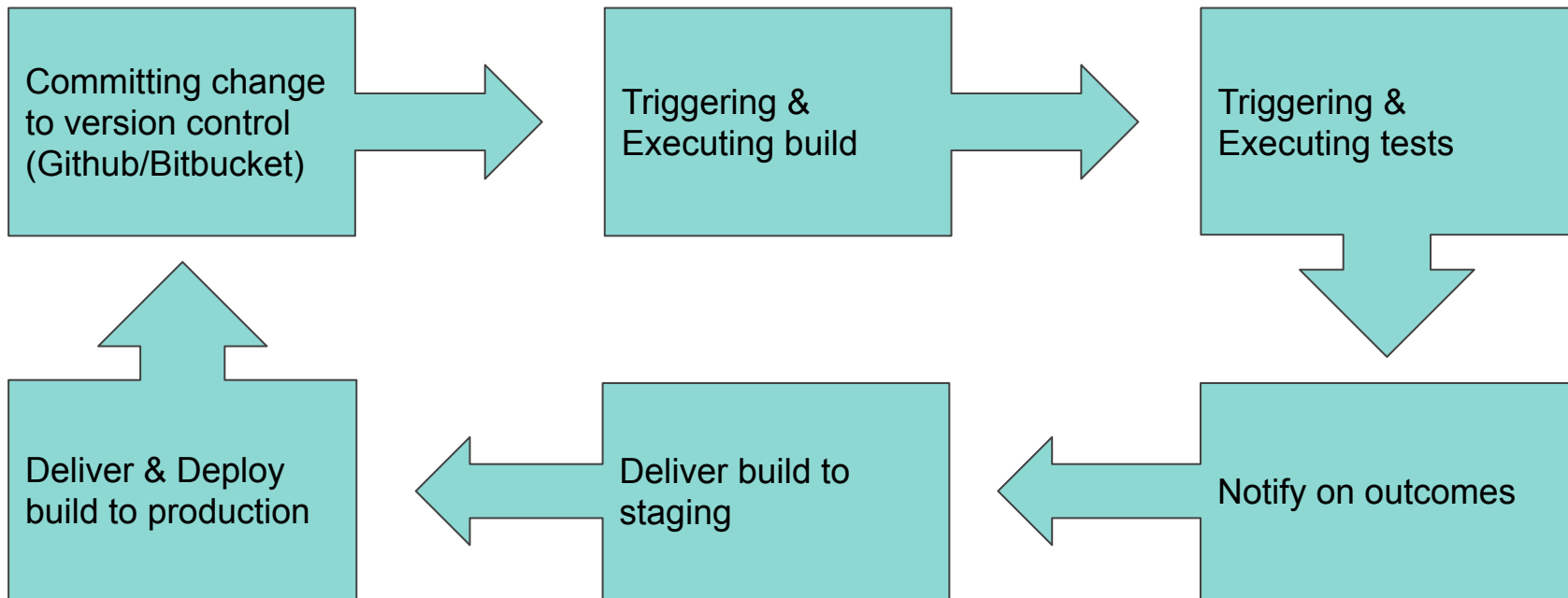
CI/CD

Modern and scalable software often utilises continuous integration, delivery and deployment (CI/CD).

- CI is the strategy of running tests continuously and automatically every time a developer makes changes to the source code. Catches bugs early, ensures new code doesn't break old code or other parts of the app. Tools: Jenkins, CircleCI, CodeShip, etc.
- Continuous Delivery is the methodology of frequent shipping of code to different environments manually, like testing/staging/production. It's more akin to an approach than anything that relies on supporting tools.
- Continuous Deployment is the automation of both delivery and integration, to achieve constant and fluid release of code into production environments. This requires a lot of preparation and great coding practices due to the risky nature of it. The tools are mostly the same as they are for integration. (Jenkins, CircleCI, GitLab CI, etc)

CI/CD pipeline example:

Committing change to version control (Github/Bitbucket) → Triggering & Executing build → Triggering & Executing tests

Deliver & Deploy build to production ← Deliver build to staging ← Notify on outcomes

Waterfall, Agile, Iterative, Spiral

- All are aiming to achieve the same goal
- All have their uses, pros and cons
- However, Agile tends to be much more popular, especially for software projects.
  - Having short, cyclical iterations makes the SDLC more efficient
  - We develop software and features continually and in small increments

Optional task:

- In 4 breakout rooms, explore the following models, their pros and cons:
  - Waterfall
  - Iterative
  - Spiral
  - Agile
- You might need to read up on the others to compare and contrast

It's not just "engineers" these days!

As the ecosystem evolves and the industry grows, more and more specialist supporting roles are emerging. The SDLC accommodates more than "just" devs!

Senior Technical Architect (Content Discovery)
BBC
Salford, England, United Kingdom (Hybrid)

Promoted

Software Engineer - UNO Portal
RES
Truro, England, United Kingdom (Hybrid)

Promoted

DevOps Engineer (Terraform)
Synechron
London, England, United Kingdom (Hybrid)

Promoted · Easy Apply

SC Cleared Software Engineer
iO Associates - UK/EU
Gloucester, England, United Kingdom

Promoted

Tech Lead - £750 outside IR35
Source Technology
London, England, United Kingdom (Hybrid)

6 days ago

Senior Consultant, Full Stack/Java Developer, Public Sector
Deloitte
Manchester, England, United Kingdom (Hybrid)

Promoted

Software Engineer
CGI
Tweedbank, Scotland, United Kingdom (On-site)

Promoted

Full Stack Engineer
Vekta Group
Edinburgh City, Scotland, United Kingdom (On-site)

Promoted · Easy Apply

Senior UI Developer
Naq
London Area, United Kingdom (Hybrid)

Promoted · Easy Apply

C# Software Engineer
Spencer Scott - Technology Recruitment
Belfast, Northern Ireland, United Kingdom (Hybrid)

Promoted · Easy Apply
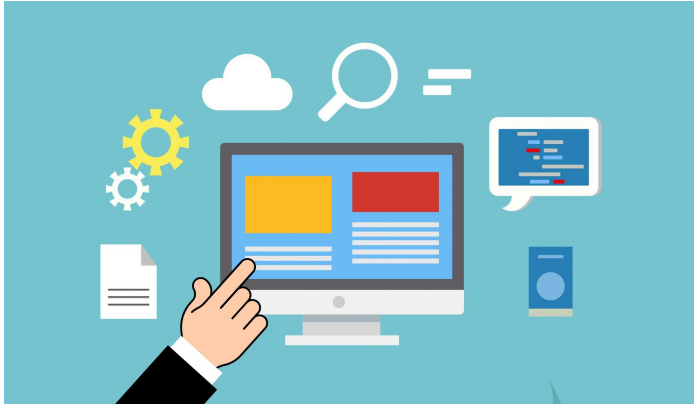
Architect

DevOps

Tech Lead

Software Engineering

Engineer vs. Developer

- Fundamentally the same thing!
- …but there are some differences within the role
  - Junior
  - Mid-level / "medior"
  - Senior
- Difference between each level depends on the company
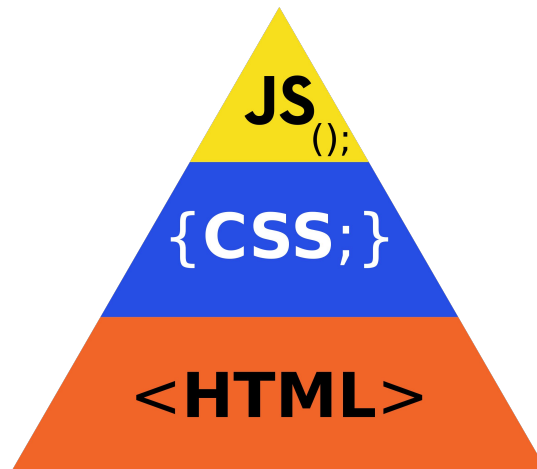
Software Engineers

Front-End

Back-End

Front-End Engineers

- Work on the user-facing parts of an application
- Implement new features to improve user experience
- Ensure accessibility standards are met
- Working with:
  - HTML
  - CSS
  - JavaScript
  - Frameworks - React, Vue, Flask

Back-End Engineers

- Work on the server-side parts of an application
- Implement new features to support front-end functionality
- Ensure the app runs efficiently
- Working with:
    - Java, Python, JavaScript, ...
    - SQL
    - Frameworks - Spring, Django
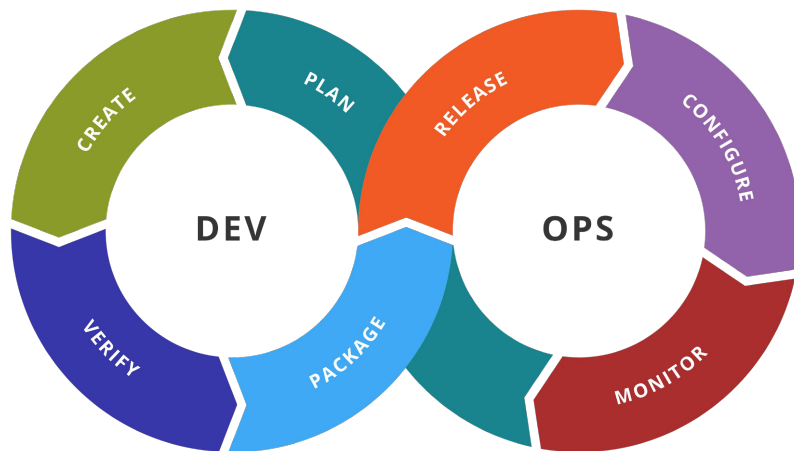
Things they don't do (much)

- Requirement gathering
- Building applications from scratch
- Publishing/deploying code

DevOps Engineer

- Manages the deployment process
- Implements process to support the dev team
- Manages source code (eg. GitHub)
- Containerisation
- Automation of processes
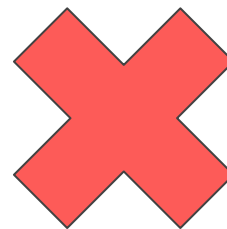
Site Reliability Engineer (SRE)

- Ensures the app stays up!
- Works with devops to manage demand
- Monitors and manages physical infrastructure
- The one who gets the midnight phone call...

Testers

- Not always a separate role
- Write tests to ensure the product meets the specification
- Can cover many forms of testing
  - Unit
  - Integration
  - End-to-end
  - Acceptance

"Tech Adjacent" Jobs

- Product Owner - managing a project backlog
- Scrum Master - managing Scrum processes
- UI/UX designer - designing and testing an app's front end
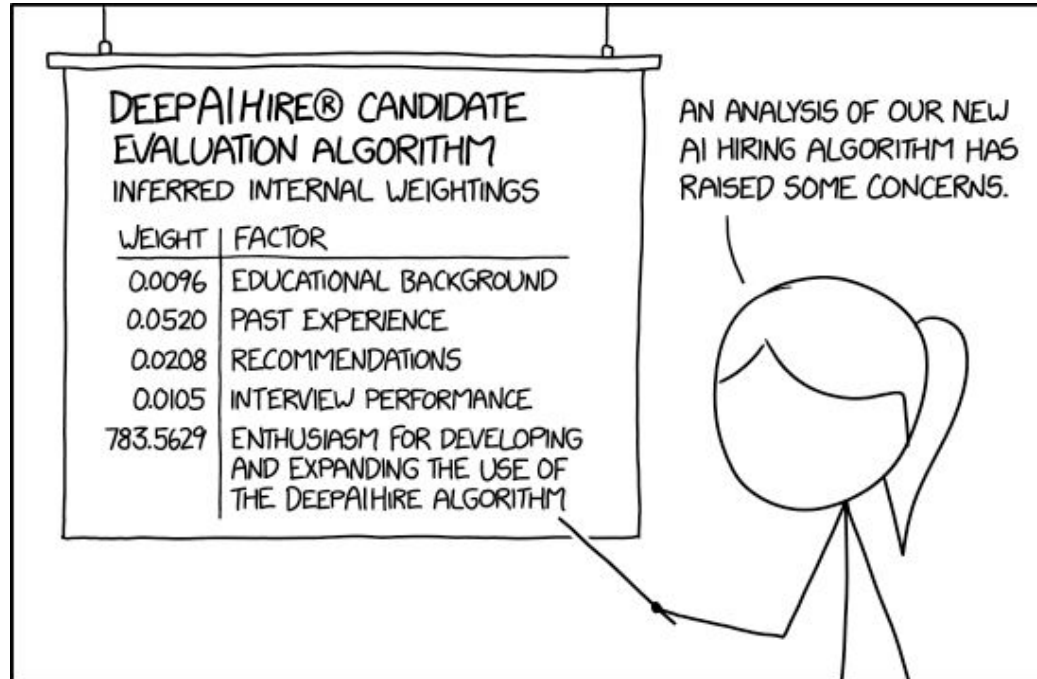- Support Engineer
- Technical Writer

What they all need...

- Communication
- Knowledge of the other roles and how they relate to each other
- Interest and enthusiasm for technology
- Understanding of the development process

And Finally…

Q&A