COM619 DevOps Development Operations Wrap Up

Dr Craig Gallen Eng.D MIEE, C.Eng

Craig Gallen

Email : craig.gallen@solent.ac.uk

Desk : JM506 (or at home)

Mobile: +44 (0) 7789 938012

Teaching approach

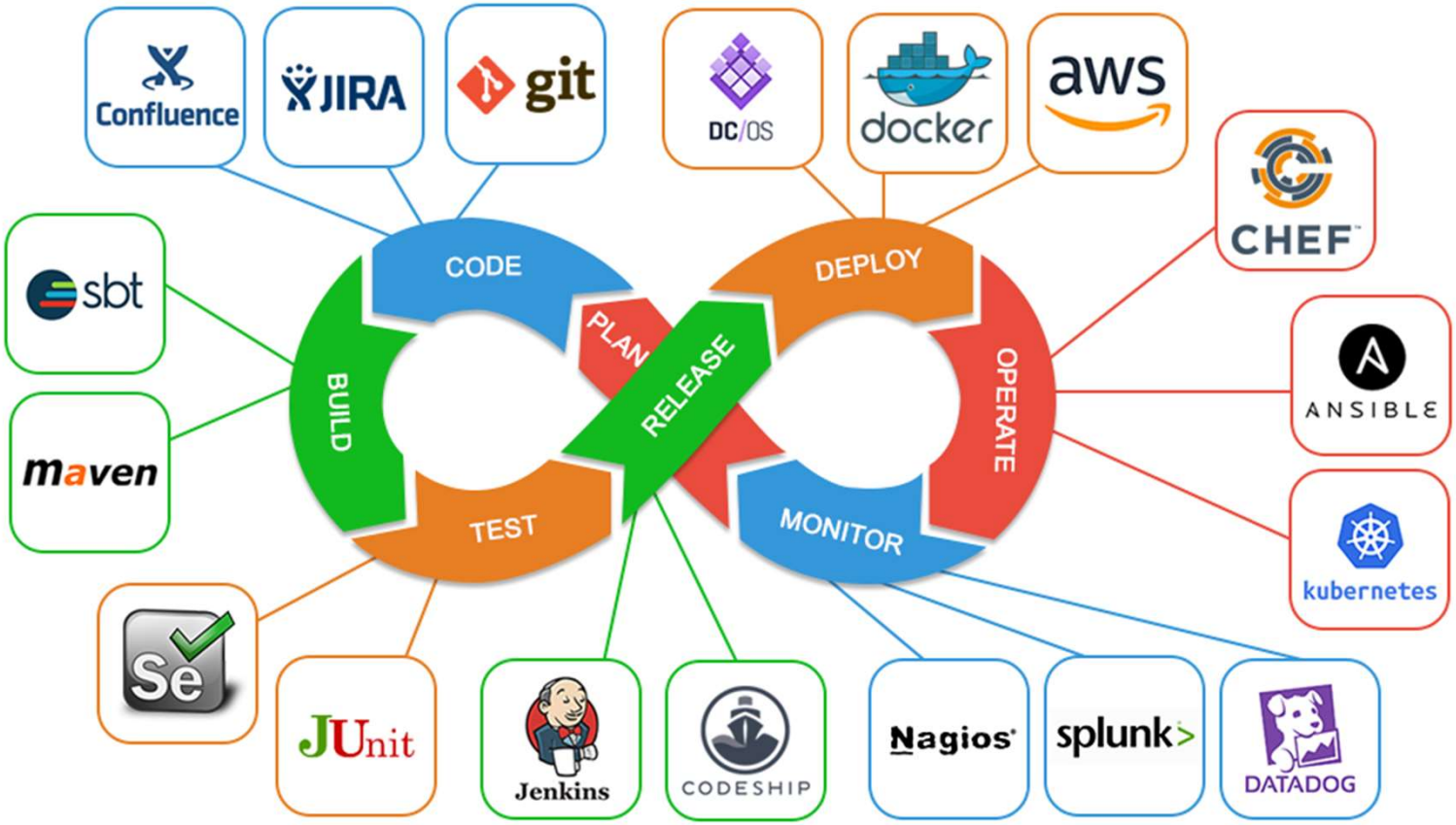
Designing and organising a **realistic and achievable project**
which allows students to **experience a DevOps environment**
and
contrast their experiences with industry best practices

What is DevOps?

- **Microsoft**
  - A compound of development (Dev) and operations (Ops), DevOps is the **union of people, processes and technology** to continually provide value to customers.
  - https://azure.microsoft.com/en-gb/overview/what-is-devops/

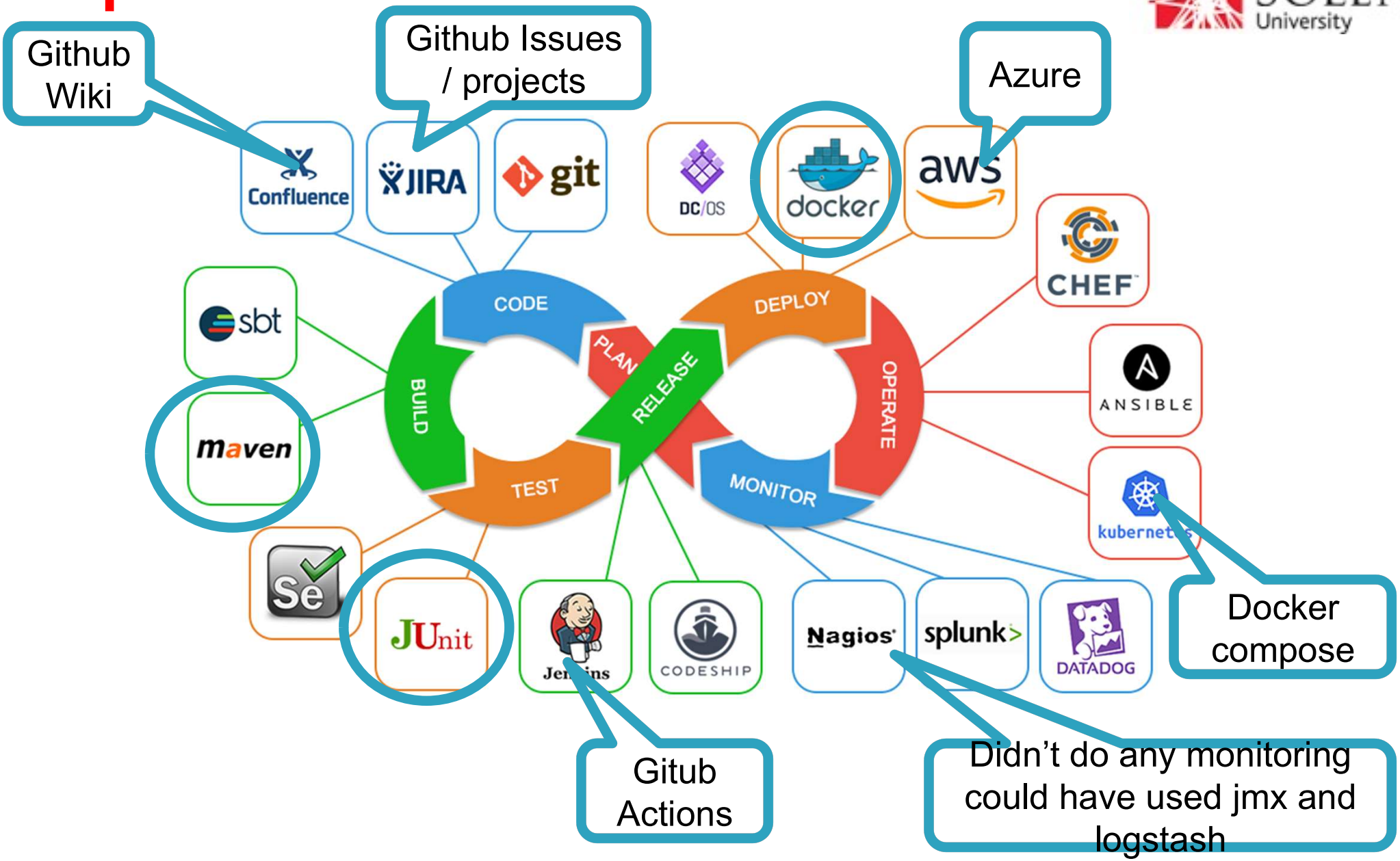
- **Amazon**
  - DevOps is the **combination of cultural philosophies, practices, and tools** that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.
  - https://aws.amazon.com/devops/what-is-devops/

Devops Tools and Process



- **https://www.pentalog.com/blog/the-devops-engineer-job-explained**

Devops Tools and Process



- **https://www.pentalog.com/blog/the-devops-engineer-job-explained**

How will we work together

- **Duties of a Devops engineer**
  - https://www.instituteforapprenticeships.org/apprenticeship-standards/devops-engineer/

- **Social / Process**
  - Duty 2 Initiate and facilitate **knowledge sharing and technical collaboration** with teams and individuals, with a focus on supporting development of team members.
  - Duty 3 Engage in productive **pair/mob programming** to underpin the practice of peer review.
  - Duty 4 Work as part of an **agile team**, and explore new ways of working, rapidly responding to changing user needs and with a relentless focus on the user experience. Understand the importance of continual improvement within a blameless culture.
  - Duty 13 **Accept ownership** of changes; embody the DevOps culture of 'you build it, you run it', with a relentless focus on the user experience.
  - Duty 11 Keep up with cutting edge by committing to **continual training and development - utilise web resources for self-learning**; horizon scanning; active membership of professional bodies such as Meetup Groups; subscribe to relevant publications.

- **Taking responsibility**
- **Pair Programming**
- **Self learning**
- **Investigate, Experiment, Choose and Commit**
- **Communicate**
  - Documentation
  - Project Planning
  - Messaging
  - Video conferences
  - email etc.

How we will design

- **Design / Program**
  - Duty 1 Script and code in at least one **general purpose language** and at least one **domain-specific language** to orchestrate infrastructure, follow test driven development and ensure appropriate test coverage.
  - Duty 8 Evolve and **define architecture**, utilising the knowledge and experience of the team to design in an optimal user experience, scalability, security, high availability and optimal performance.
  - Duty 9 Apply leading **security** practices throughout the Software Development Lifecycle (SDLC).

- **Java EE technologies**
  - building on COM504
  - Messaging JMS
  - Web Servlets
  - Jersey / Jaxb
  - Spring
  - Junit
  - Maven

- **You will be provided with examples**

How we will deploy our solution

- **Deploy and Monitor**
  - Duty 5 Build and operate a **Continuous Integration (CI) capability**, employing **version control** of source code and related artefacts.
  - Duty 6 Implement and improve **release automation & orchestration**, often using Application Programming Interfaces (API), as part of a continuous delivery and continuous deployment pipeline, ensuring that team(s) are able to deploy new code rapidly and safely.
  - Duty 7 **Provision cloud infrastructure** using APIs, continually improve infrastructure-as-code, considering use of industry leading technologies as they become available (e.g. Serverless, Containers).
  - Duty 10 Implement a good coverage of monitoring (metrics, logs), ensuring that alerts are visible, tuneable and actionable.
  - Duty 12 Look to **automate** any manual tasks that are repeated, often using APIs.

- **Devops tools**
  - Docker / docker compose
  - CI/CD
  - Version Control
- **Cloud deployment**
- **Test Harnesses**
- **Monitoring solutions**

Team Exercise - discuss

- **Without naming and shaming, attaching blame or abdicating responsibility**
  - what would you have liked to have known before joining the class
  - what could you have done better personally
  - what could the team have done better
  - what could the whole class have done better
  - what could the lecturer have done better

My Feedback

- **Possibly the first time you have had to work within a development team**
  - students often
    - work to own schedule,
    - don't share work,
    - don't document code
- **Take responsibility for your own learning**
  - don't expect to be taught rather than taking responsibility for own learning.
- **Release early, release often**
  - important to publish your code to share your thinking and progress with your team
  - important to show thought leadership - teaching the rest of the team
- **Think about the needs of the whole project**
  - it is never just about your own team – for this to work everyone needs to succeed
  - discussing interfaces / contracts
  - reviewing code and designs across teams
- **It always takes longer than you think**
  - de-risk through experiments.
  - Do the most important things first
  - refactor later if you don't know everything now
- **Be tenacious**
  - Expect problems and don't give up

Have fun…