

IN5140

Smart processes and agile methods in Software Engineering

Group session 12

- DevOps, systems thinking and dealing with complexity

Agenda

- Repetition of Lecture
- Previous Exam Questions

Questions?

Speak out loud, or type in chat

Repetition from lecture

Summary

- Why system thinking is needed?
- The ability to change is influenced by factors such as culture, (transformational) leadership, structure within the organisation
- How to adapt to complexity?
- Some keywords
 - Cynefin framework
 - Team topologies
 - Conway's law
 - DevOps, team boundaries with cross functionality

4 key metrics to measure delivery performance

Q1 : What is System Thinking?

Adapting to Complexity in a rapidly changing world

The world is changing faster

- Rapid change brings more **VUCA** (Volatility, Uncertainty, Complexity, Ambiguity) -> increase complexity
- Failing to understand complexity -> create more issues!
 - a. How to deal with complexity?
 - i. Using Cynefin Framework : choose the right approach based on the problem domain

Systems Thinking and Ability to Change

- Systems thinking is a holistic approach to understanding complex systems and how different parts interact to create a larger whole
- Every system can change but often resists change.
 - a. The ability to change is influenced by factors like culture, leadership, and structure within the organization.

System Thinking

Every system changes

-> Some more than others

- Every system has resistance to change

-> Some more than others

- A systems ability to change is a product of qualities from the systems culture, leadership and structure (processes)

-> Processes that someone has dedicated their energy into creating and optimizing

- Optimized processes can also be the very things limiting the ability to change

What is a process?

A process is a series of sequential actions or steps taken to achieve a particular end. It involves the systematic progression of activities that transform inputs into outputs, guiding the way tasks are performed to reach a specific goal efficiently and effectively.

Tools

Tools and decisions made

Tool choices

- Azure DevOps
- GitHub(?)
- (Almost) A single organization
- Different processes for each team

Issue

- We need a tool to manage tasks and our process
- We need better CI/CD tools
- We need to work better across teams
- Different types of teams have different needs

Q2 : Describe Cynefin Framework

Find problem domain with the Cynefin Framework

Or explained with Cynefin



Q3 : Describe

a. What is DevOps

b. three ways of DevOps practices

Deliver value with DevOps

DevOps (development and operations)

- deliver software rapidly and reliably, continuously providing values to users.

How to deal with Complexity?

- The three ways of DevOps practices
 - 1st way (**flow**-systems)
 - 2nd way (amplify **feedback** loops)
 - 3rd way (culture of **continuous** experimentation and learning)

Q4 :

- a) Describe two measure for measuring “flow” of software delivery performance and**
- b) Describe two measure for “reliability” in terms of software delivery performance**

Measuring software delivery performance

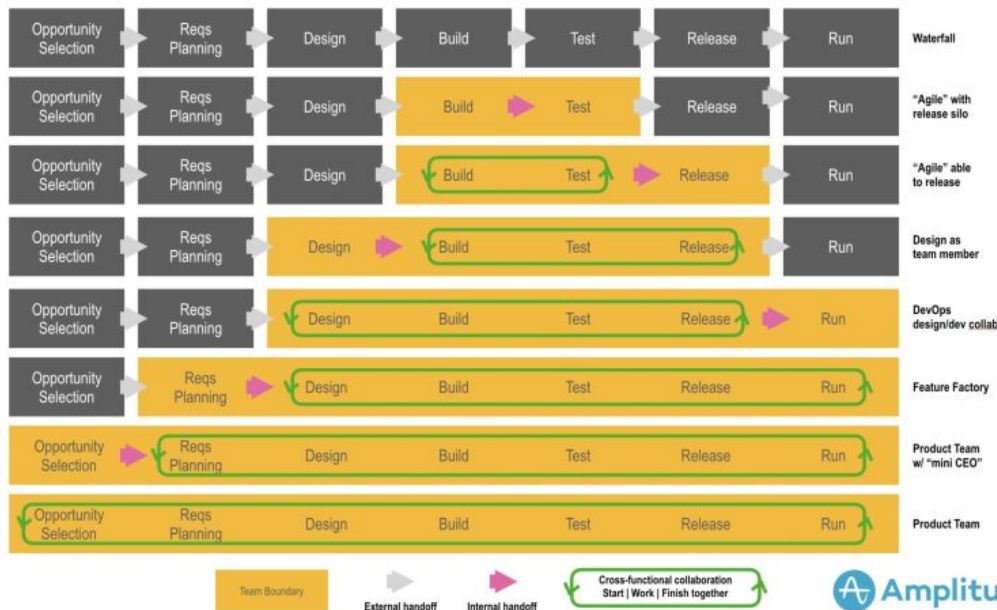
- For **flow** : How quickly software is delivered
 - Development frequency
 - Lead time for change
- For **reliability** : to assess system stability
 - Time to restore service
 - Change failure rate

Q5 : What is team boundaries?

Impact of team structure and collaboration on software delivery

Team boundaries

And cross functionality on different levels



Team boundaries : who is responsible for what takes and where one teams **responsibilities** end and another's begin

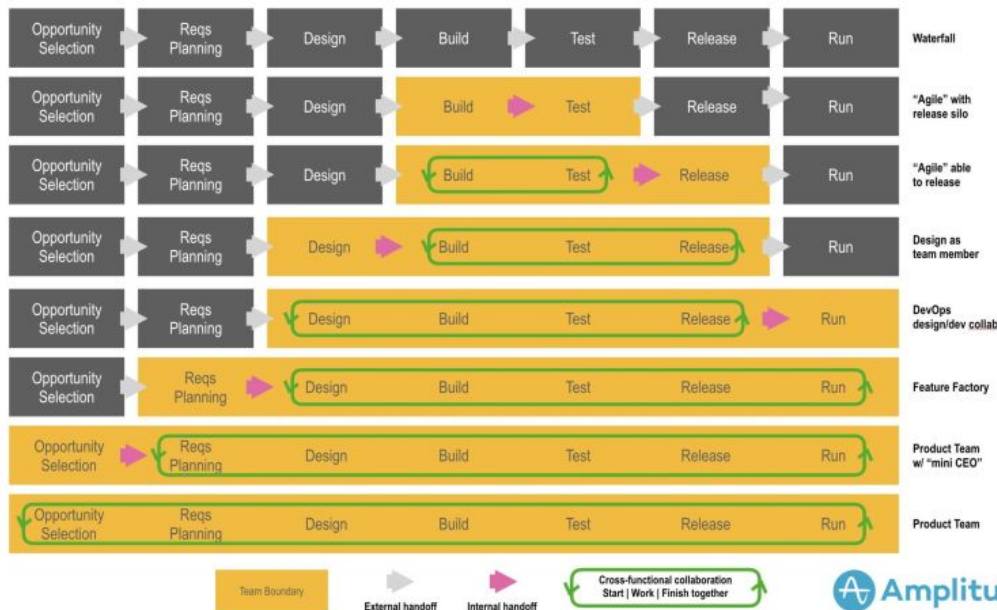
Q6 : Discuss in terms of how structure (team boundary) impacts

- a) the cross functional collaboration,**
- b) low/high flexibility**
- c) whether making the changes easy or not**

Impact of team structure and collaboration on software delivery

Team boundaries

And cross functionality on different levels



Team boundaries : who is responsible for what takes and where one teams **responsibilities** end and another's begin

Discuss it in terms of :

How is the structure (team boundary)

How is the cross functional collaboration

low/high flexibility?

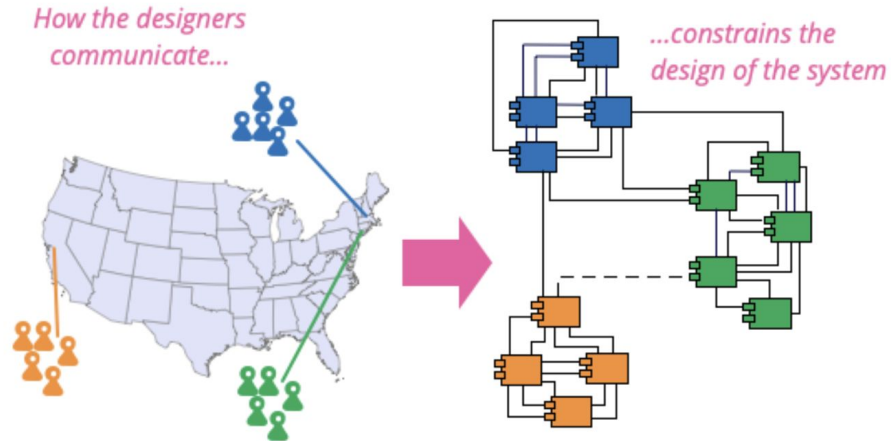
making the changes easy?

Q7

Describe the Conway's law.

Conway's law

- System will mirror the **communication patterns of the organization** building the system
- It suggests that organizational structure impacts software architecture.



Team topologies

To organize for flow, they have defined team topologies

Principles and practices for fast and sustainable stream of flow

Reduce cognitive load for developers

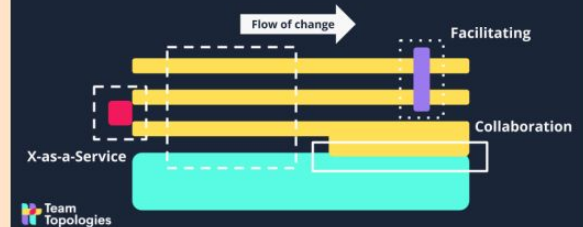
Team Topologies

4 fundamental topologies

- Stream-aligned team
- Enabling team
- Complicated Subsystem team
- Platform team



3 core interaction modes

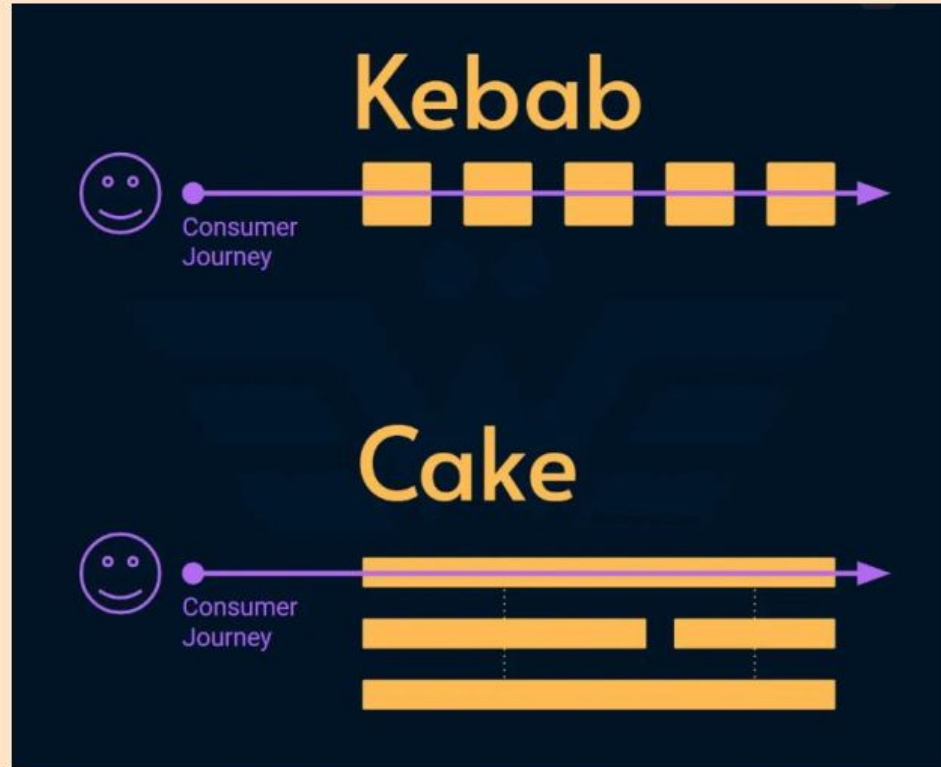


Common DevOps mistakes

- **Oversimplification:**
 - Trying to simplify complex systems too much, which can overlook important details.
- **Automating what is fluid and unpredictable :** Attempting to automate processes that are inherently fluid and unpredictable, which may lead to issues.
- **Overly focusing on tools:** failing to change in other demensions
- **Lack of System Thinking:** Ignoring the overall system view, which leads to isolated improvements rather than optimized outcomes across the entire system.

Remember there is a tradeoffs! prioritizing "flow of value" can sometimes have cost of other important factors.

Kebab organizations vs Cake organizations



By Alex Ewerlöf

<https://blog.alexewerlof.com/p/kebab-vs-cake>

News from Dora State of DevOps Report 2024

While AI boosts individual **productivity**, It can **negatively impact overall software delivery performance** due to larger code batches and changes -> RISK

Importance of **transformational leadership** is required

- Embrace uncertainty and inspire change
 - Leaders *motivate* employees to achieve higher performance by appealing to their *values* and sense of *purpose*, facilitating organizational change

Previous Exam Questions

(2022)

DevOps

Briefly explain the concept of DevOps. (4p)

Fill in your answer here

When should you perform testing in DevOps? (1p)

Fill in your answer here

(2019)

IN5140 H19 Exam Task 1.7

What does **DevOps** stand for?

- ☐ Development and Operations
- ☐ Development and Optimisation
- ☐ Deployment and Operations
- ☐ Deployment and Optimisation

(2018)

No	Question	Answer A	Answer B	Answer C	Answer D
4	Which one of the following methodologies does least impact the establishment of DevOps methodology?	Waterfall Software Delivery	Agile Software Delivery	Continuous Software Delivery	Lean Manufacturing

(2018)

7	What does DevOps stand for?	Deployment and Optimisation	Development and Operations	Deployment and Operations	Development and Optimisation
---	-----------------------------	-----------------------------	----------------------------	---------------------------	------------------------------

(2016)

Discuss the challenges of working agile in the development teams (using **DevOps**) when the project contract is based on the waterfall model.

Possible Explanations

Main challenge is connected to the planning stage which is completely different in waterfall and agile process models. In the waterfall model, planning and design of the system is done in the early beginning of the development process, while in agile (DevOps) scenario planning is continuous.

Unfortunately, developer teams will have to follow milestones mentioned in the contract. However, developers won't be forced to invest too much time into design by following waterfall model (meaning, they can start development earlier, in contrast to the waterfall model) ...

Possible Explanations

... **DevOps** is also embracing short release cycles, which confronts with the waterfall model where releases and milestones are much less frequent. Short release cycles also imply fast feedback loop in which product owner should be involved. Based on the feedback, development teams can adjust goals and move forward faster. Since waterfall model is enforced by contract, feedback is expected only on releases listed in roadmap, which is not sufficient for **DevOps**.

In order to adapt waterfall to **DevOps**, development team can prioritize minimal set of absolutely essential features and deliver Minimum Viable Product (MVP) as soon as possible.