SOFTWARE ENGINEERING RETHOUGHT

Designing cognitive software engineering work systems

Adapting for change with better knowledge

Fabian Fagerholm (fabian.fagerholm@bth.se)

SERL Sweden
LEADING SOFTWARE ENGINEERING

KK-stiftelsen

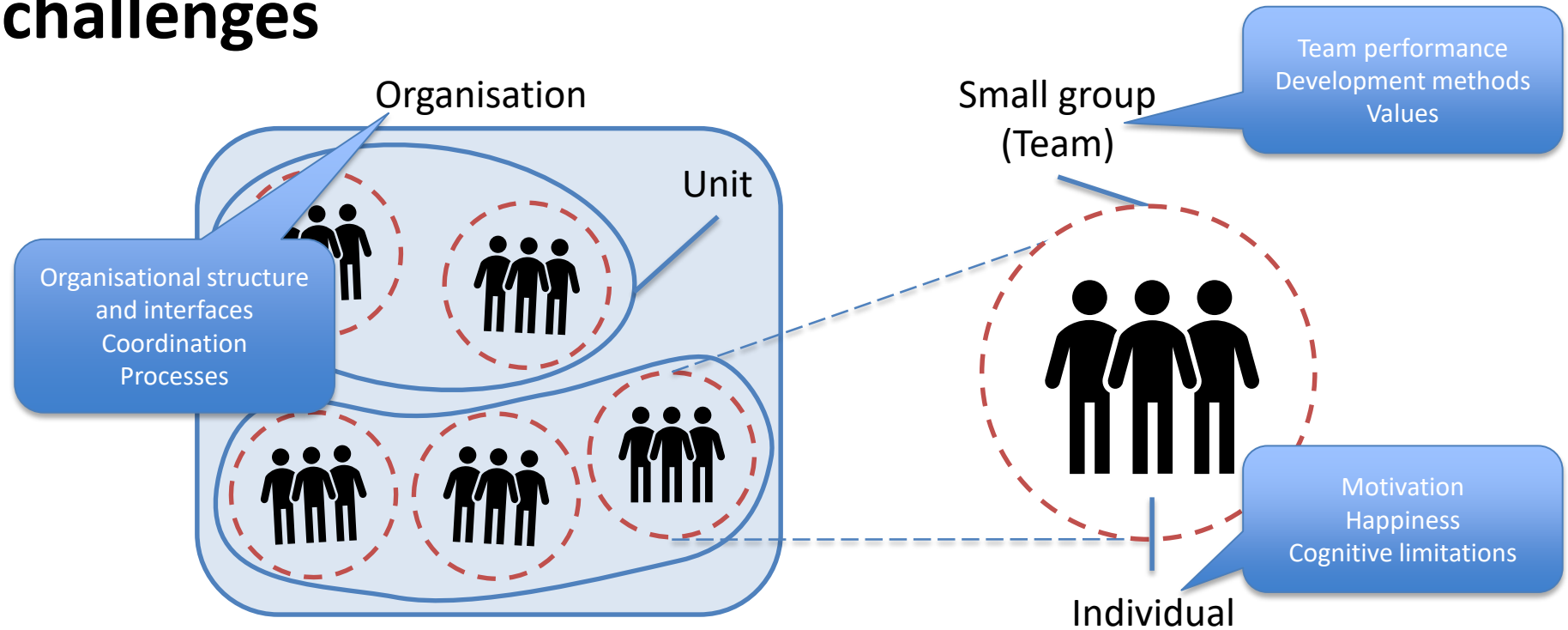
background/introduction

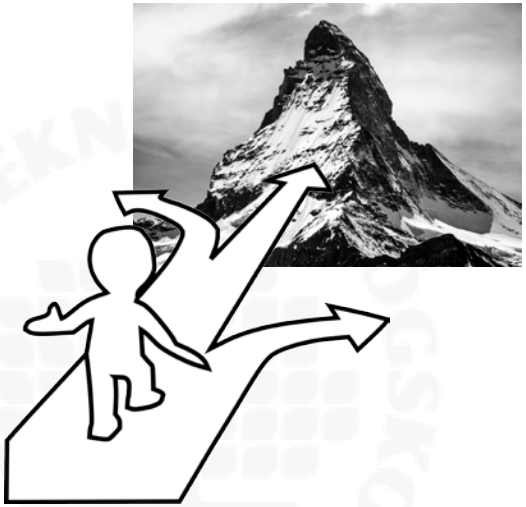
(Fabian Fagerholm)

→ Researcher – Engineer

- PhD Computer Science (Software Engineering), M.Sc. Computer Science
- Associate Professor at BTH
- 10+ years in industry (consultant, software architect, project manager, trainer)
- 9+ years in research
  - Human and organizational factors in software engineering
  - Data-driven software product development
  - Continuous experimentation
  - Lean/Agile software development
  - Open Source Software development
- Deputy project leader **SERT**

challenges

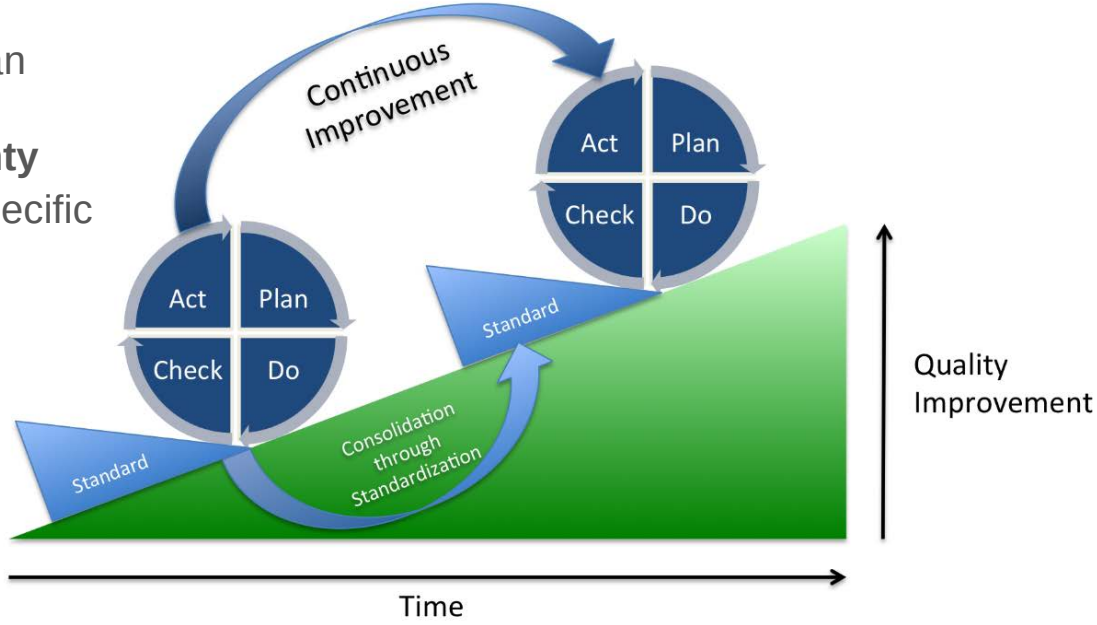




- Our agile approach doesn't scale
  - We can't put all stakeholders in one team
  - We need interfaces between teams and they require coordination
  - We want to design software delivery early on but DevOps adds even more interfaces between teams
- We have waste because our organizational structure doesn't match the way we need to work
- Our work procedures cause mental overload and are not motivating

pipeline work system

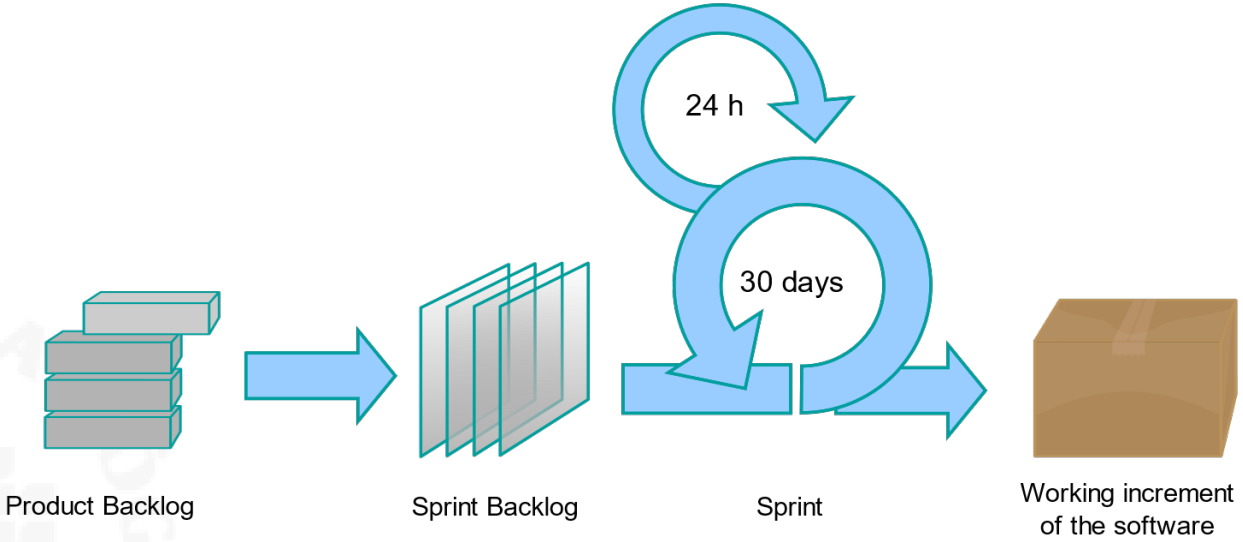
- Each step transforms inputs to outputs
- Artifacts flow in the pipelines, e.g.
  - Requirements
  - Architecture designs
  - Code
  - …
- Advantage: possible to optimize
  - High throughput
  - High quality
- Inflexible: does not adapt easily to changed demands
- Fragile: the optimization achieved can become irrelevant
- The work system focuses on **certainty**
  - Doing well-defined tasks in a specific way



certainty and change

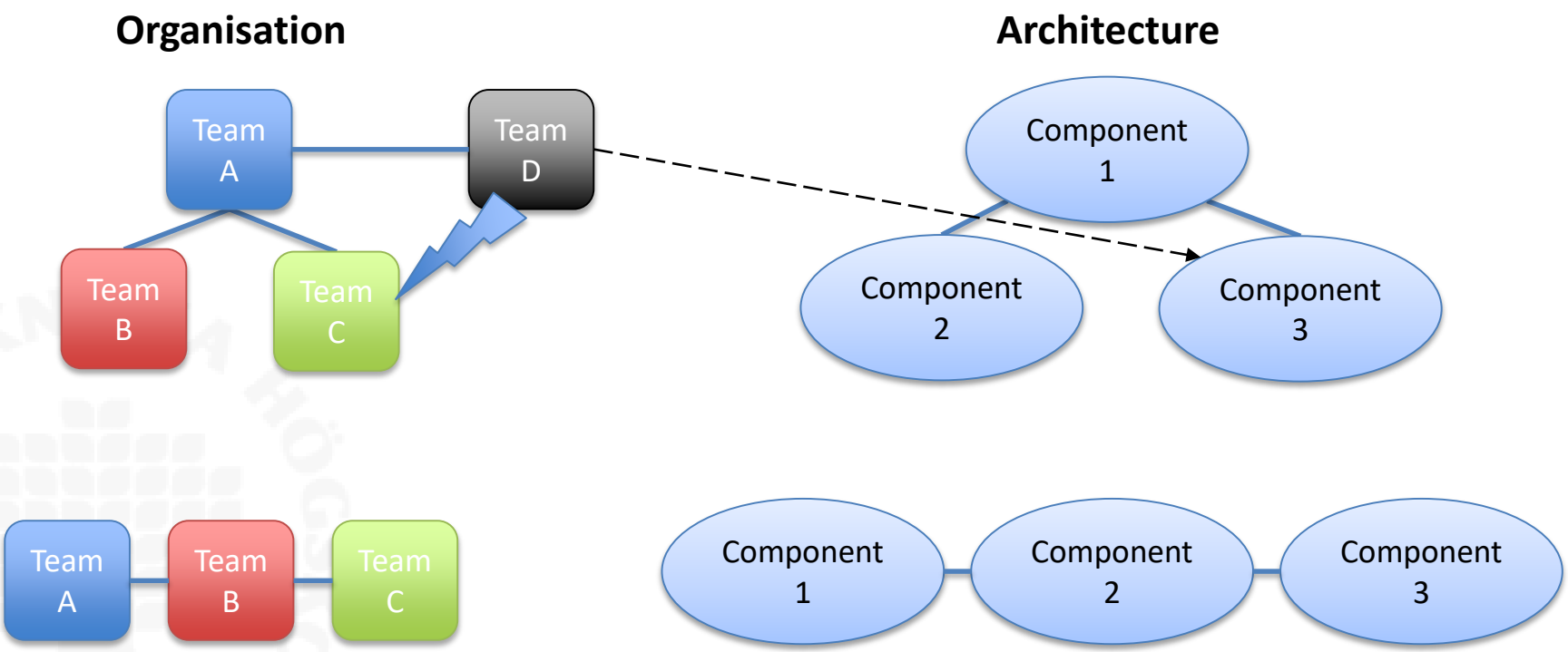
accepting some uncertainty

- Agile/Lean
  - Limit uncertainty through short cycles
  - Adapt quickly to new demands
- Scaling problems
- Large-scale agile frameworks add more structure
  - Going back to pipelines?



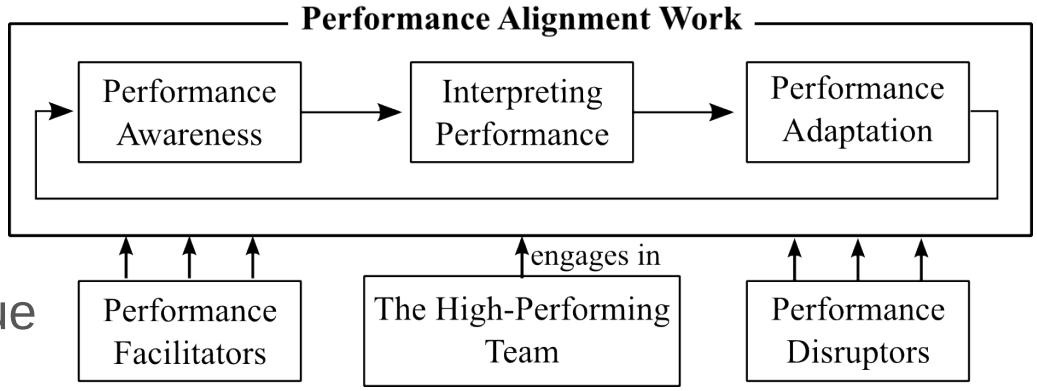
why can't we scale?

- Conway's law: The architecture will reflect the communication structure in the organization
- Scaling requires the right organizational and architectural structure
  - Detect mismatches, refactor
  - Human-computer co-design for evolving the organization and architecture



team performance

- Views on performance
  - Efficiency: doing things fast
  - Effectiveness: doing things that matter – producing **VALUE**
- Measuring often leads to undesired behavior
  - Goodhart's law: When a measure becomes a target, it ceases to be a good measure
- Need to access the team's ability to align their performance
  - New methods for situational awareness and providing teams with the visibility they need to understand value



well-being for developers

- Large study on software developers' happiness
  - 1318 participants sampled from GitHub
  - 88 countries, 75% working professionally
- Sources for unhappiness:
  - Internal: causes related to personal states or originated in own behaviors (437 occurrences)
  - External: causes external to individual developers, by which developers are affected but have little or no control of (1843 occurrences)

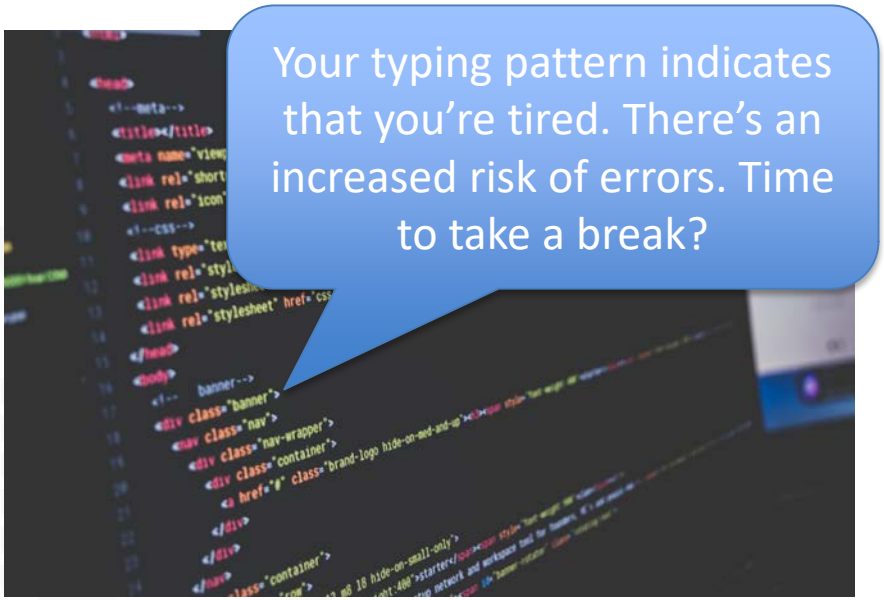
Main category	Sub-categories
Artefact and working with artefact (788)	Code and coding (217) Bug and bug fixing (194) Technical infrastructure (151) Requirements (99)
Process-related factors (544)	None
People (416)	Colleague (206) Manager (122) Customer (49)
Other causes (95)	None

We could influence these

- Develop more advanced tools
- Make the development process more flexible

cognitive work systems in the profile

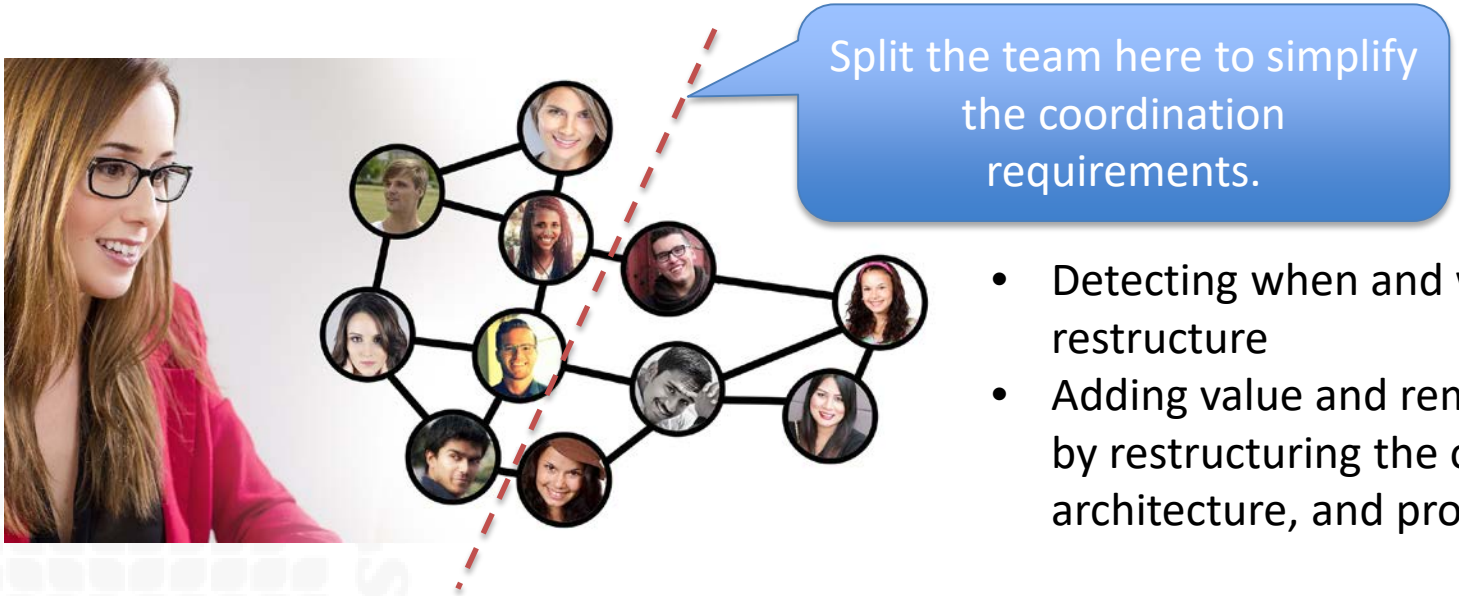
- **SP4: Cognitive software engineering development models**
  - Goal: Maximize human potential
  - Use knowledge of psychology to develop new work methods
    - When do errors occur?
    - What is motivating and demotivating?
    - How do feelings influence technical decisions?
  - Package insights into tools: cognitive augmentation



- Possibilities in all kinds of software engineering activities
  - Requirements
  - Programming
  - Testing
  - Delivery …
- We need to study specific activities in detail to learn what kind of assistance is needed and how to enable trust

cognitive work systems in the profile

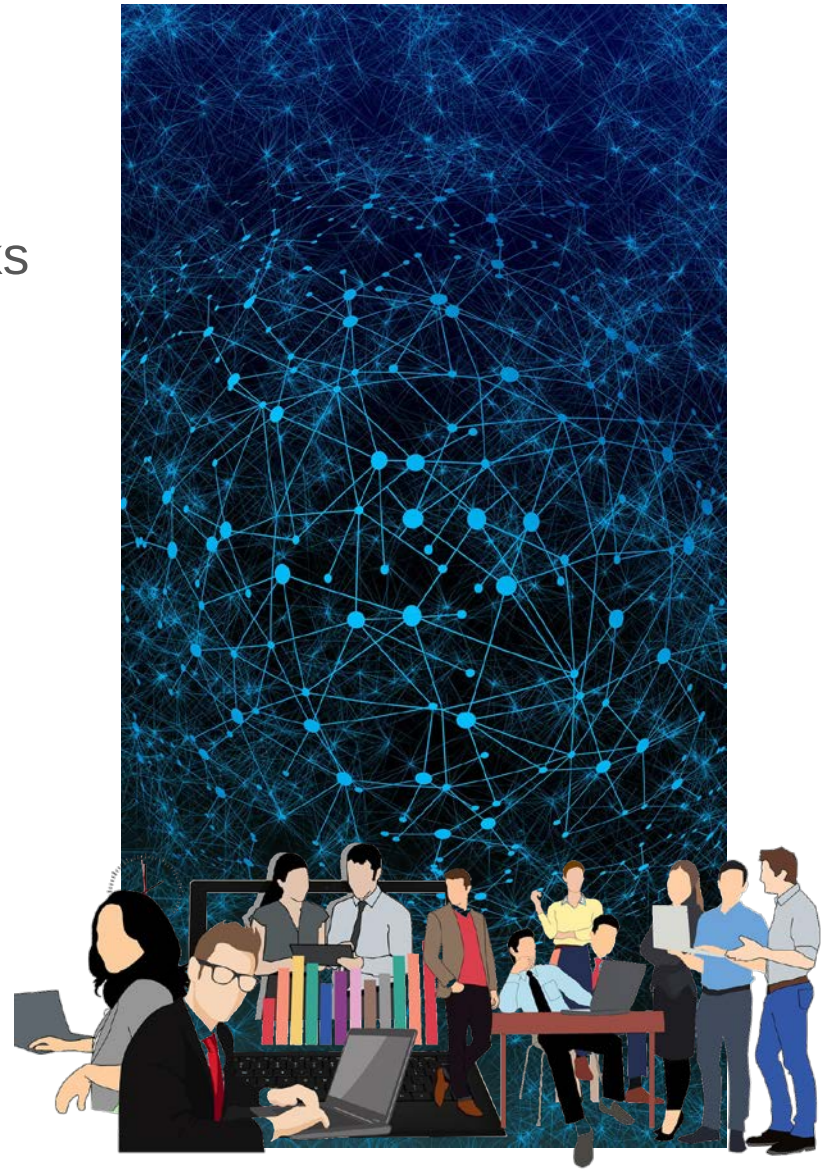
- **SP5: Study and improve LeaGile handling of organizational and team interfaces**
  - Goal: Greater flexibility and development speed while maximizing value
  - New ways of scaling agile
  - Aligning organization, process, and architecture



- Detecting when and what to restructure
- Adding value and removing waste by restructuring the organisation, architecture, and processes

cognitive work systems in the profile

- Vision: A machine-supported work system
  - Make the process interactive
  - Machine support for specific tasks
  - Certainty through knowledge
  - New methods and software tools help make sense of the complexity

- Contact
  - Fabian Fagerholm (fabian.fagerholm@bth.se)

Thank you for your attention!