

ProAgile

Reduce cognitive
load with your
software
architecture

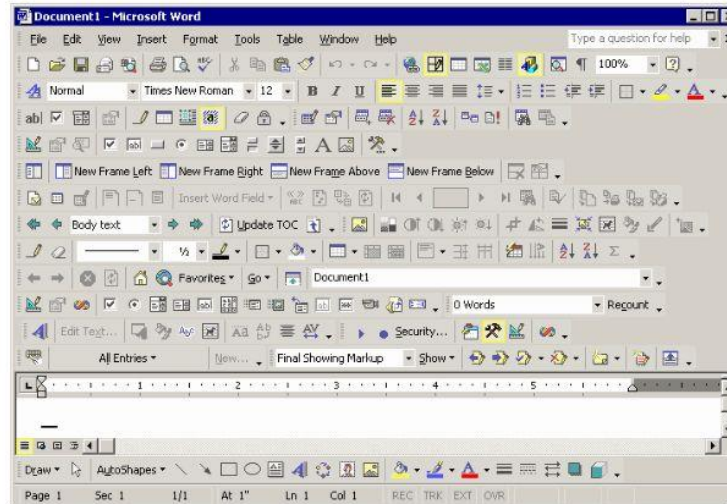
David Sundelius

 [@davidsundelius](https://twitter.com/davidsundelius)

Bad systems

Neglected UI, security, stability, performance and accessibility

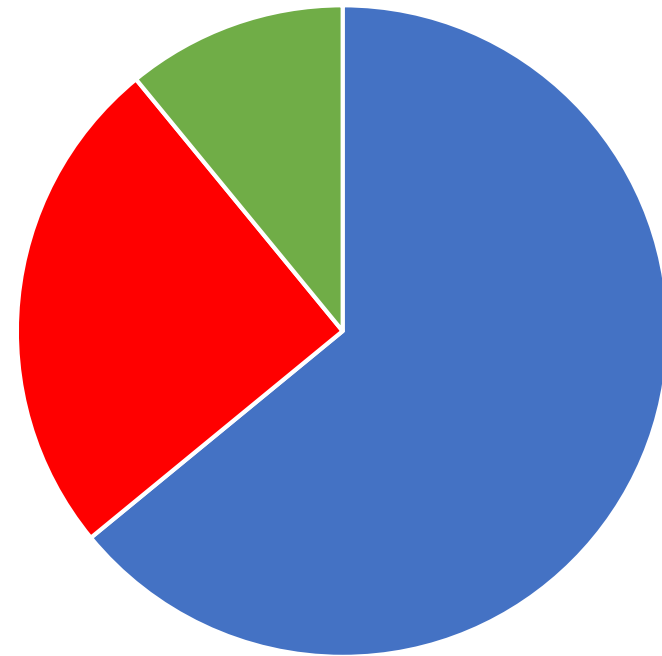
Why?



Definition: Cognitive load

“Cognitive load refers to the total amount of mental effort being used in the working memory”

- Intrinsic
- Extraneous
- Germane



Cognitive overload

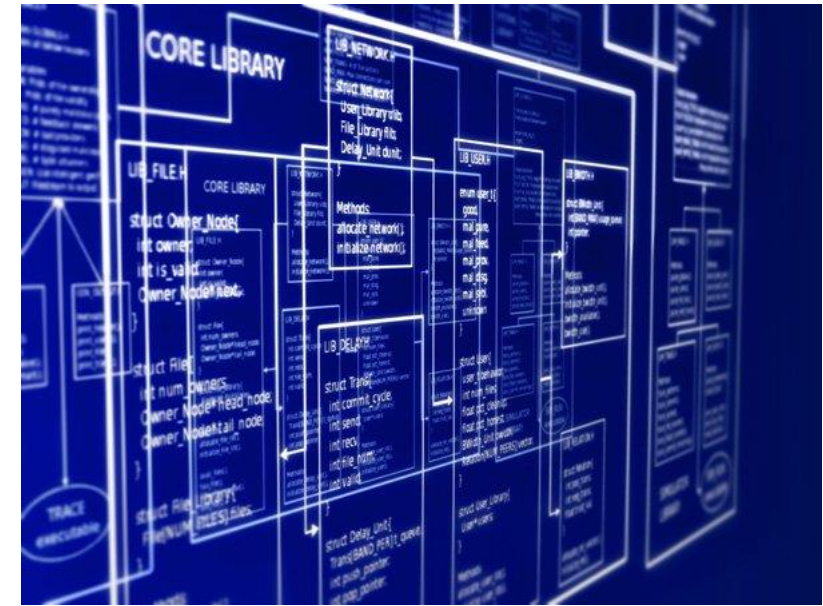
- Resignation
- Stereotyping
- Psychological issues

- Individual differences



Architecture

- *“...fundamental structures of a software system...”*
- Decisions made to optimize certain parts of the software
- Examples on impacts of cognitive load



Too low/high test coverage

- NOT a great way of measuring quality
- Too low: worry about braking vital fuctionality
- Too high: maintaining tests becomes an issue

Overdocumentation

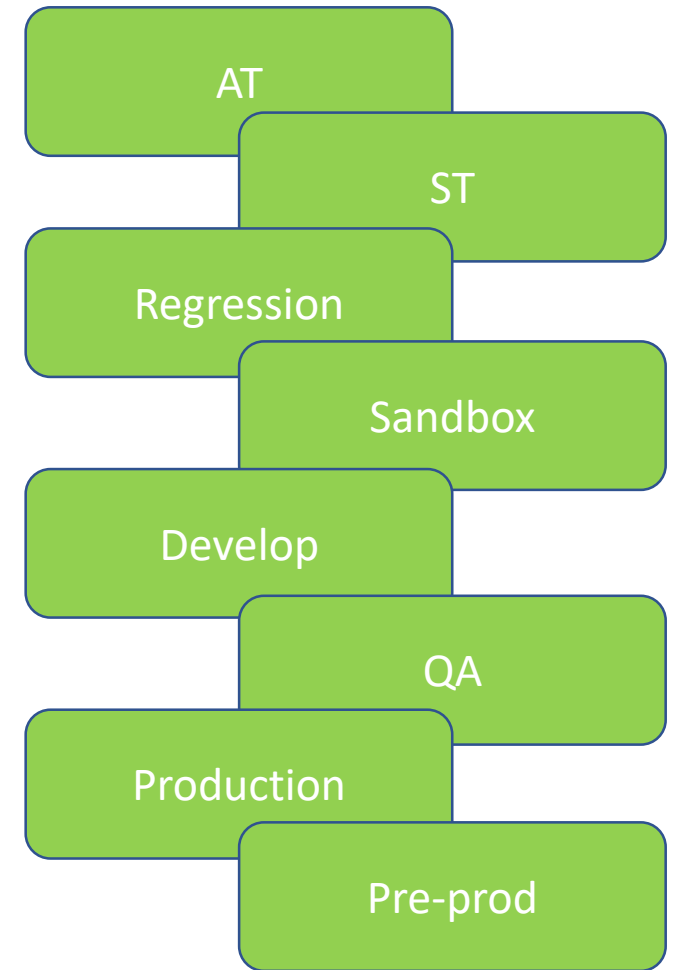
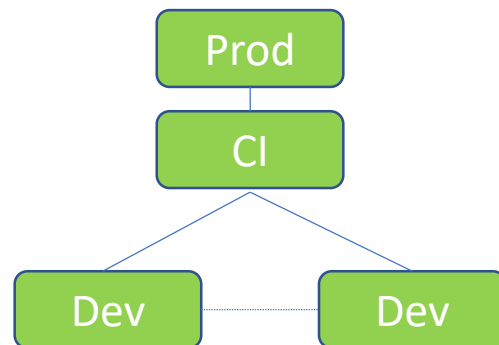
- Is it updated?
- Traceability
- Remembering what's important

Manual repetitive work

- Deployability
- Testability
- Handovers

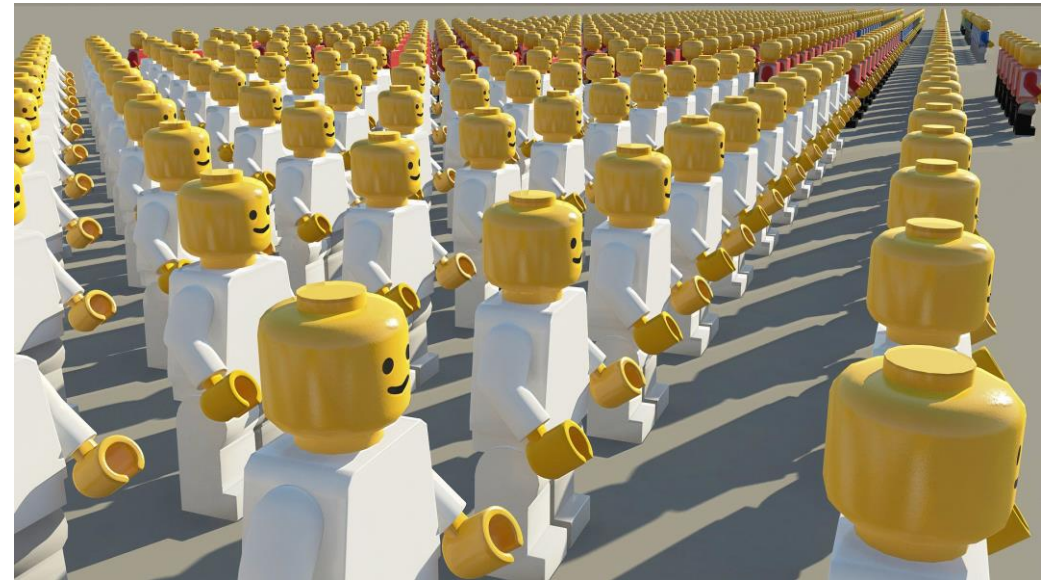
Environment overload

- In what state is the AT-environment at the moment?
- Is it going to work together with feature X?
- Will this run in production?



DRY – Shared code

- What impact will my change have?
- Is it worth the work making a change?
- Overengineering



”Build for the future” - platforms

- Unused code
- Generalization
- Focusing on the future features



Keep good things for later

- Is this used?
- Increases complexity
- Is it safe to change or remove?



Too few/large releases

- Tracking all changes
- Maintenance after release
- Synchronization/communication

Vague or implicit agreements

- Quality standards
- Interfaces
- Decision making

Abstraction overload

- Verbose code
- The metaunderstanding of the abstraction
- Harder to get an overview



What does this have to do with agility?

- *"Simplicity--the art of maximizing the amount of work not done--is essential."*
- *"The best architectures, requirements, and designs emerge from self-organizing teams."*
- *"Continuous attention to technical excellence and good design enhances agility."*

Buzzwords

- Monoliths or microservices
- Cloud based or on-premises servers
- Machine learning or blockchain

How can we improve?

- Optimize architecture on reducing cognitive load at a team level
- Find the flow of value and organize teams and architecture
- Discuss cognitive load within the team and try experiments to reduce it
- Spread knowledge within the team (business, technical and tools)
- Simplify for the humans, not the machines

- Developers with less cognitive load -> Better quality, more technical agility

ProAgile

Reduce cognitive
load with your
software
architecture

David Sundelius

 [@davidsundelius](https://twitter.com/davidsundelius)