Software architecture: between "rigid process" and "somehow I manage"

Advanced Research Computing Centre, UCL, London

Jasmin Jahić

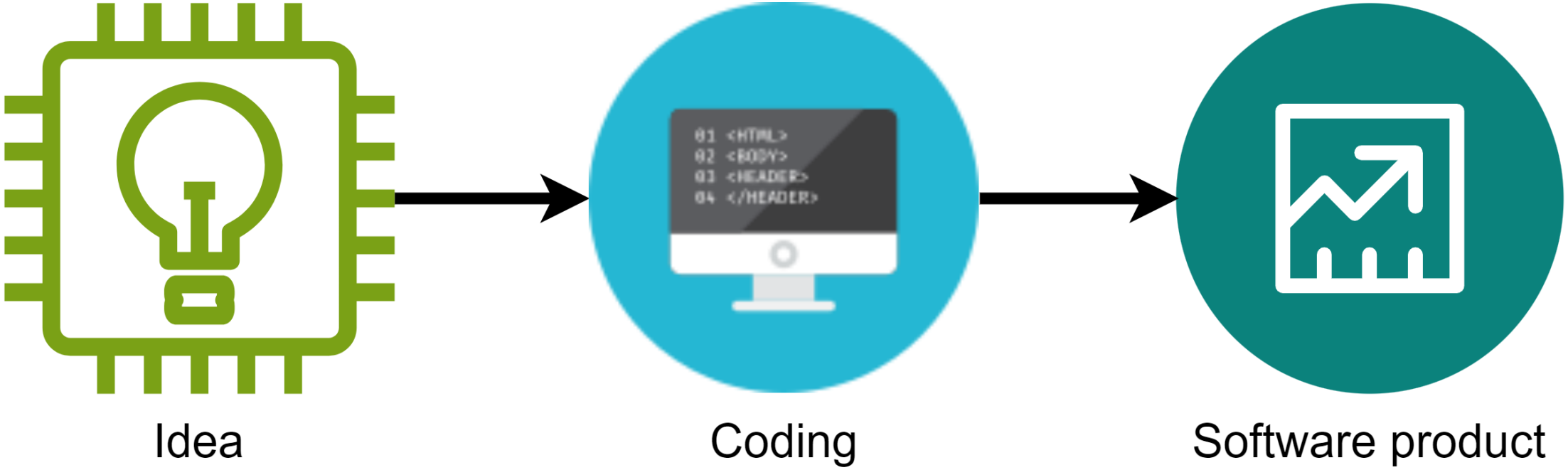
jj542@cam.ac.uk

14.08.2024

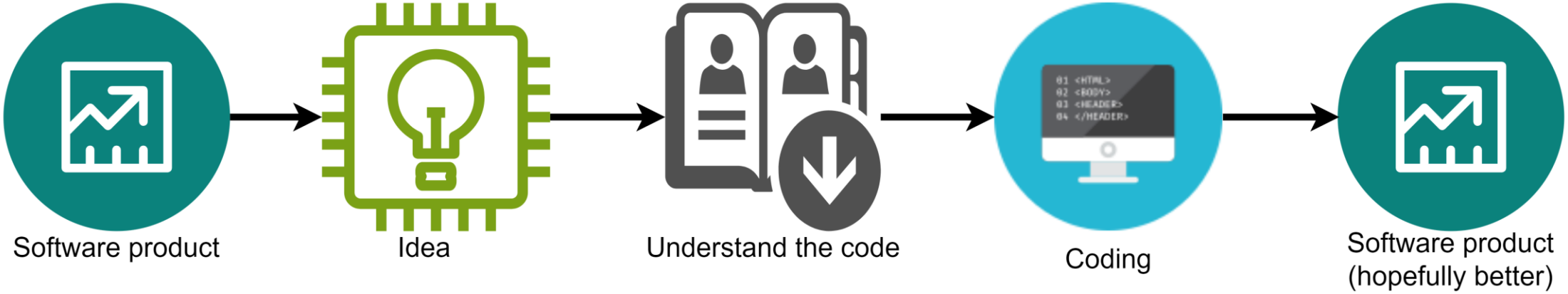
Agenda

- Overview of architectural processes and their desired outcomes
- The overhead that software architecture processes introduce
- Customization
- AI in Software Engineering (and Software Architecture)

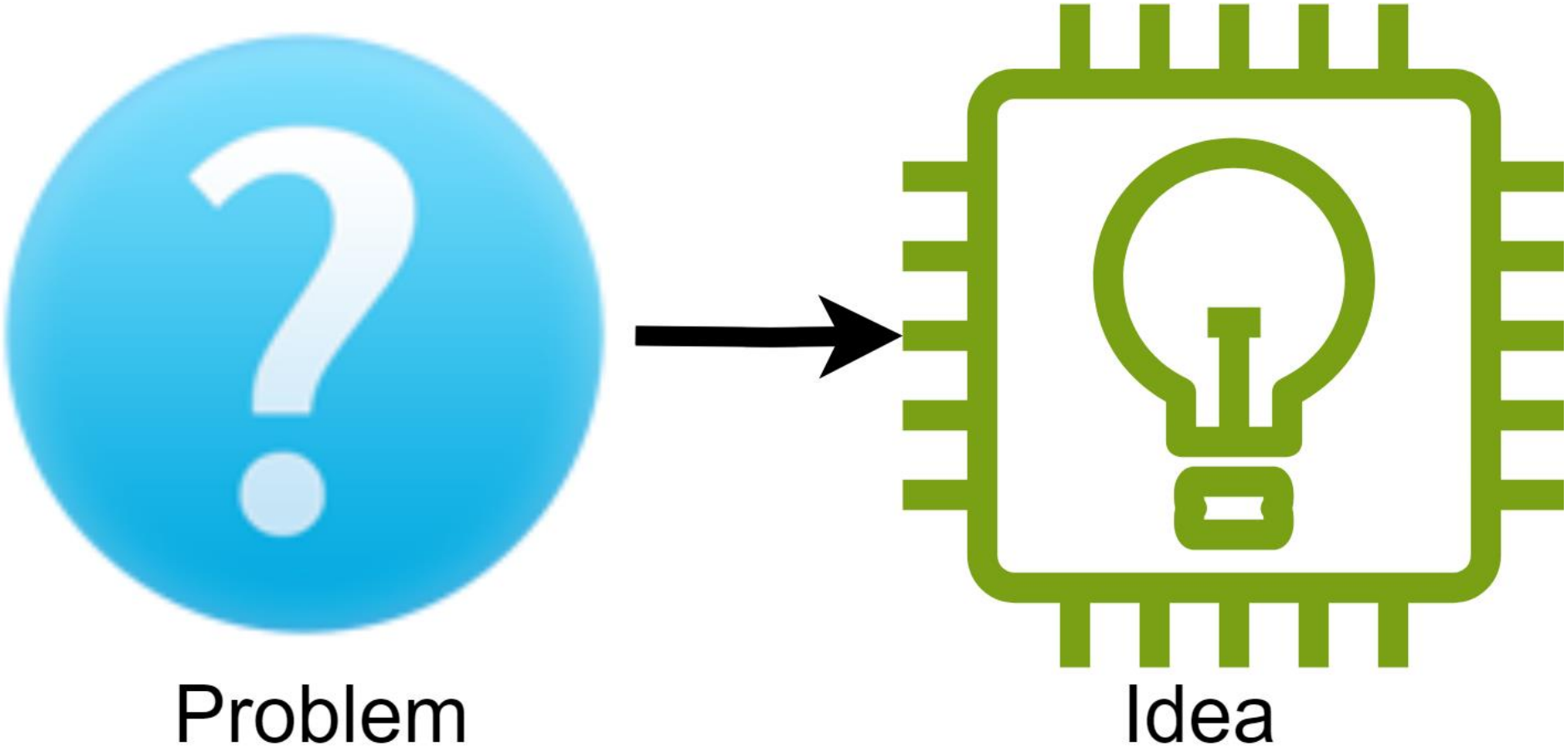
Software engineering – Scenario 1



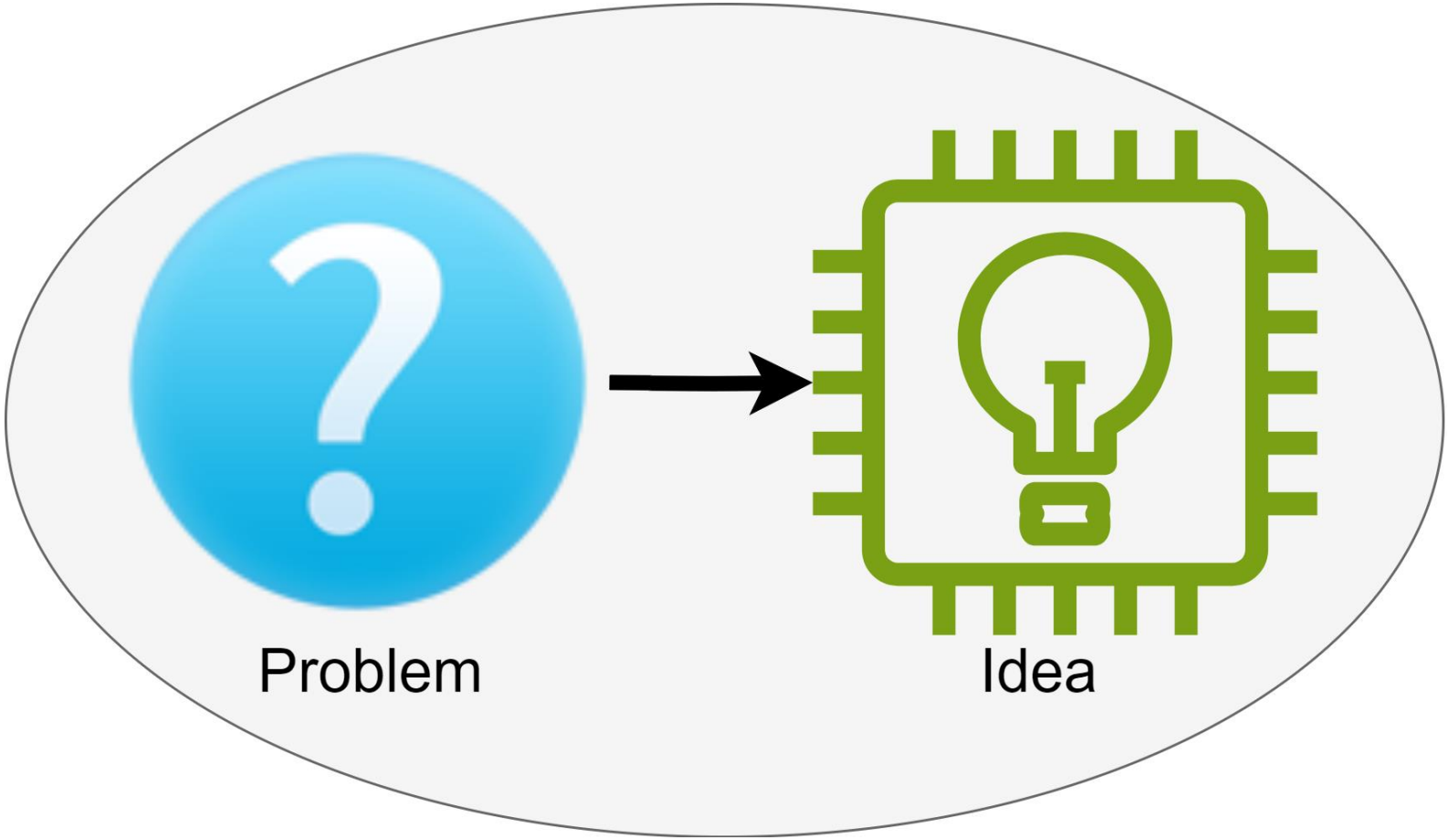
Software engineering – Scenario 2



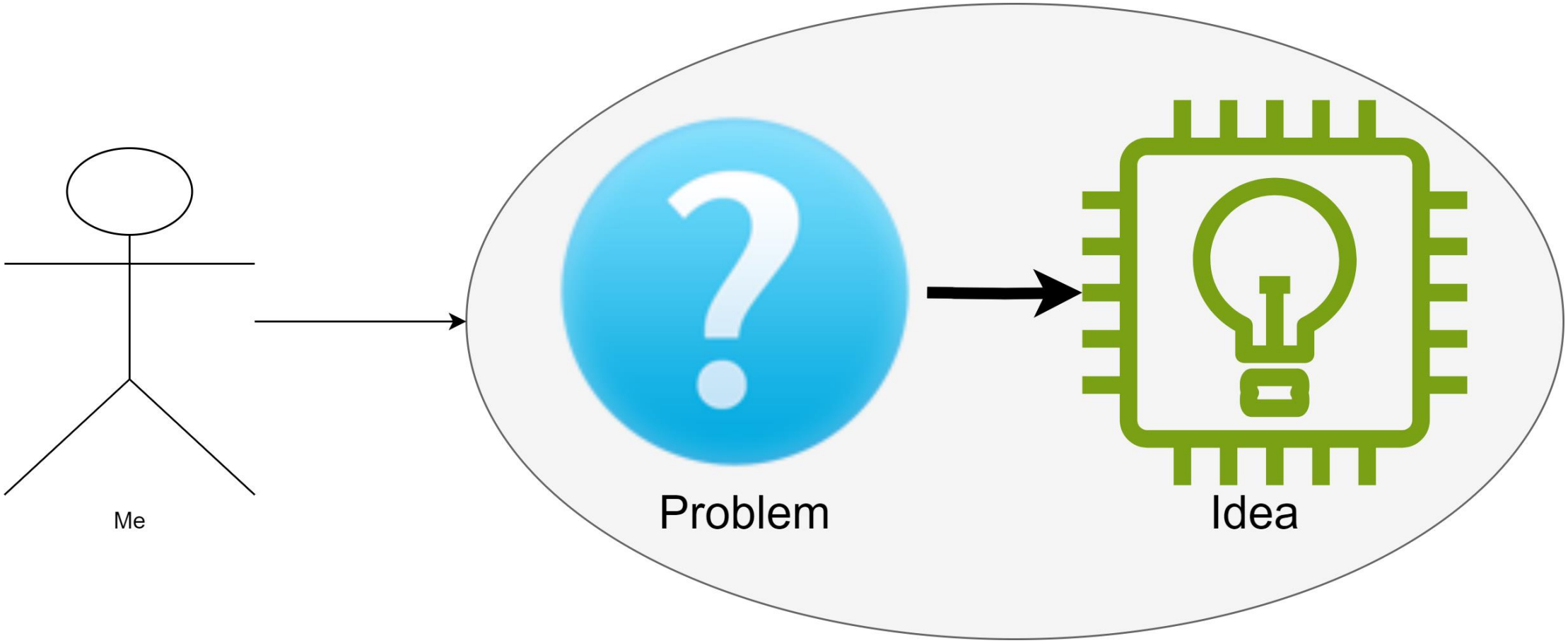
The Problem



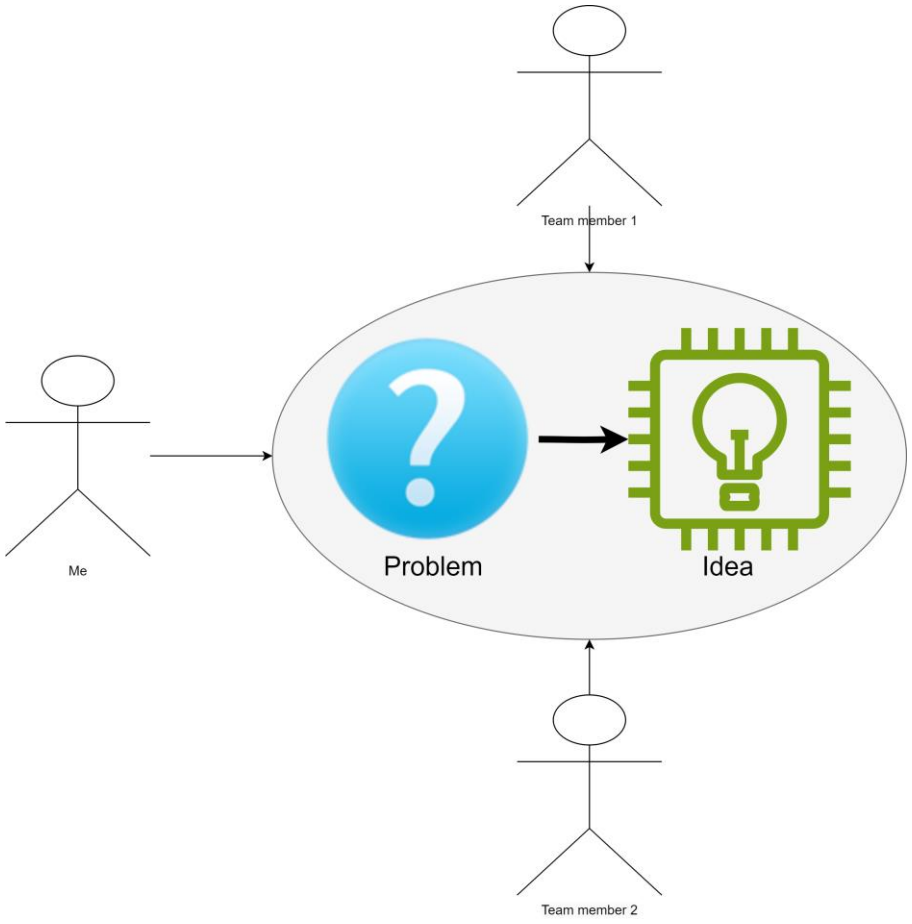
The Problem domain

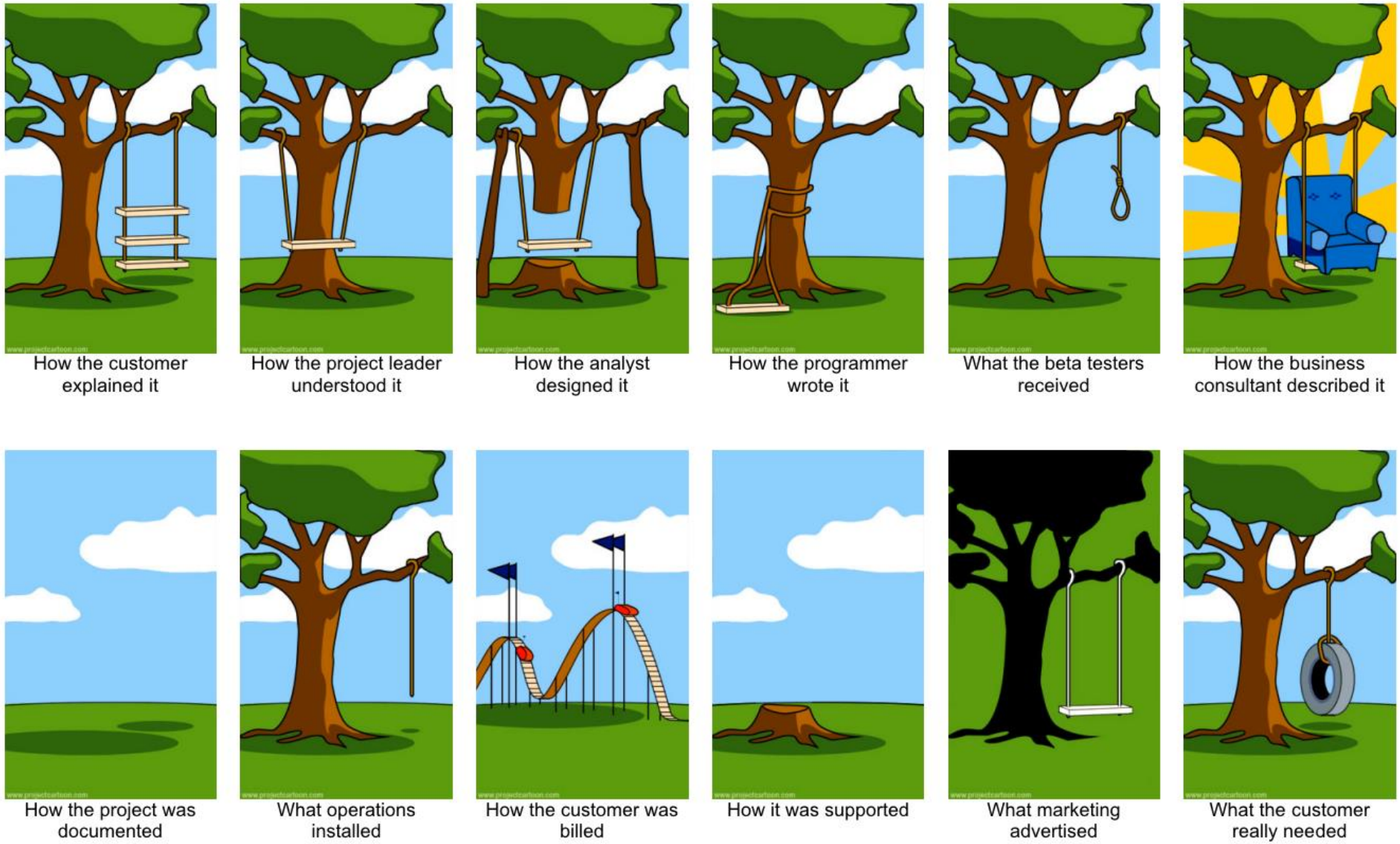


How do I personally understand the problem?



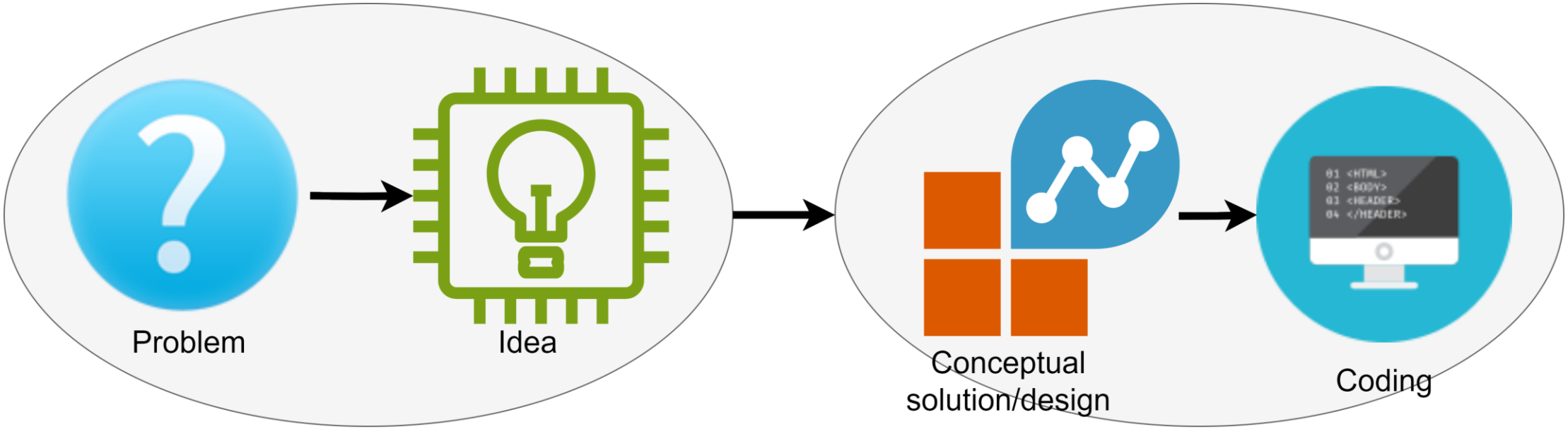
How does a team understand the problem? Common understanding?



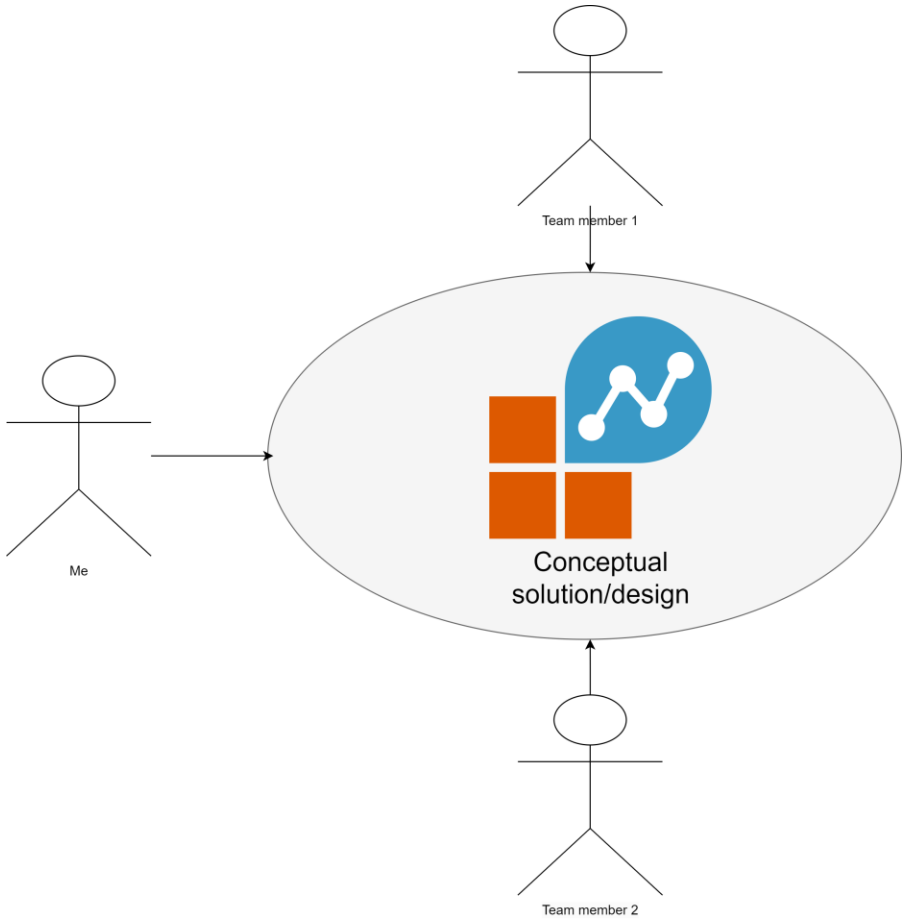


projectcartoon.com

The Solution domain



How does a team understand the solution and what needs to be coded? Common understanding?

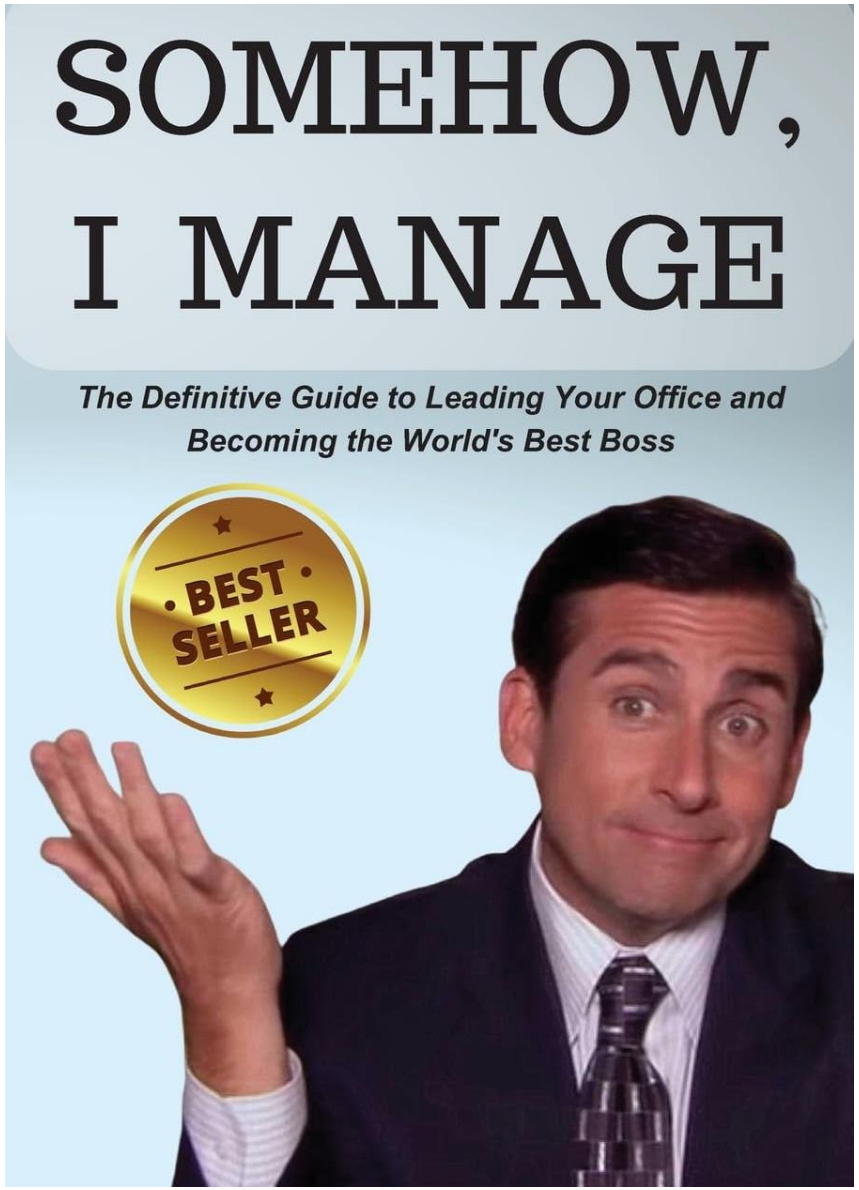


Issues

No common understanding of the problem

No common understanding of the solution

This actually works! (sometimes)



Well, if you are…

Enthusiastic

Knowledgeable

Lucky

Results

I know how to code!

No one cares Margareth

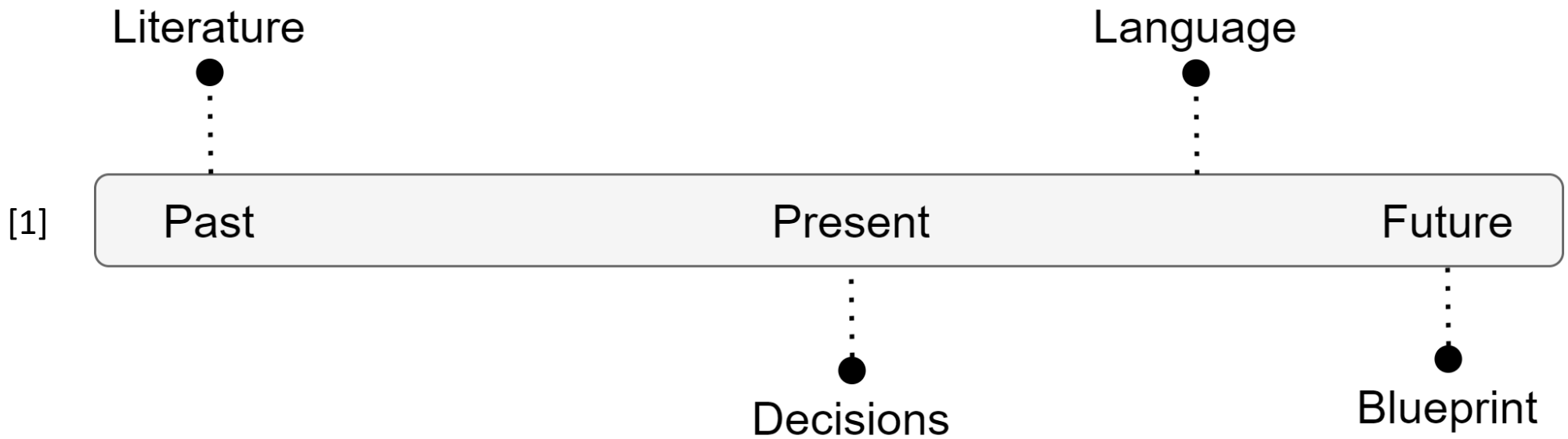
Projects in Software Engineering

- Onboarding new team members
- Taking over an existing project
- Parallelisation of work
- Integration of work created by different teams

- Prioritisation of new features
- Making decisions about new features based on the existing design
- Discussion between different teams
- Discussion between different groups of stakeholders (for example, management and developers)

Additional activities

- Organisation (timeline, division of work, roles and responsibilities)
- Common understanding
- Knowledge transfer

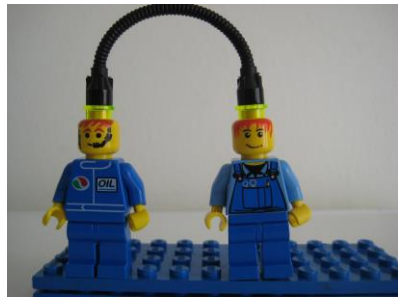


We want to

- Make success reproducible
- Enable incremental development
- Enhance communication between stakeholders
- Support making hard design decisions while managing trade-offs
- Enable reasoning and management of changes as a system evolves
- Establish processes that are reproducible and enhance system quality

Superhero developers

No memory loss



Stay forever or transfer knowledge properly

Can translate between different stakeholders

Common understanding of problems and solutions

Can we really get people with all of these qualities?

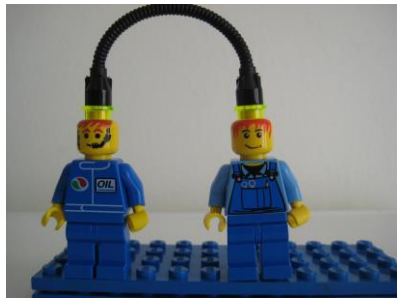
Enthusiastic

Knowledgeable

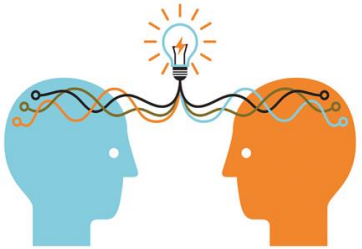
Lucky

„Somehow I manage“ is not good enough

No memory loss



Stay forever or transfer knowledge properly



Can translate between different stakeholders

Common understanding of problems and solutions

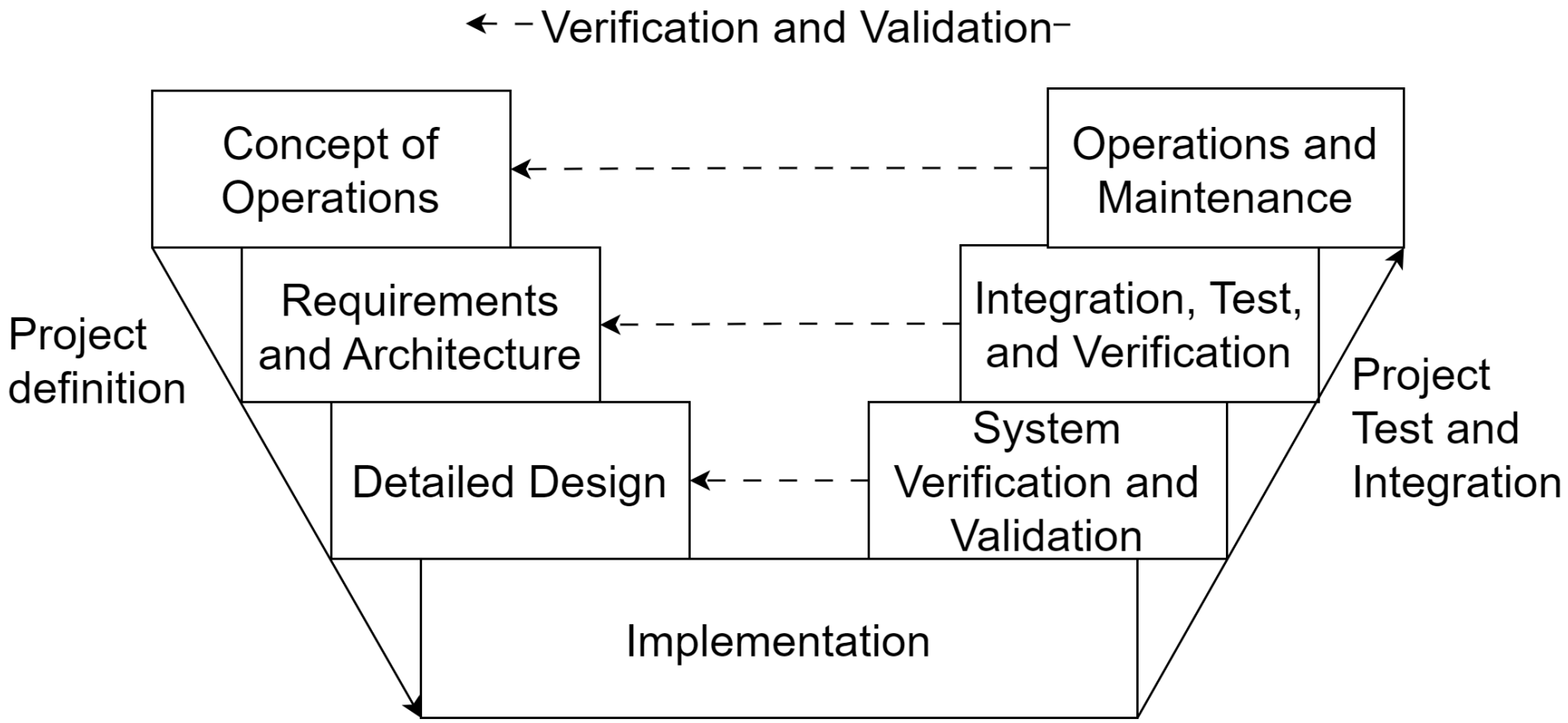
Implicit software architecture

…and document them.

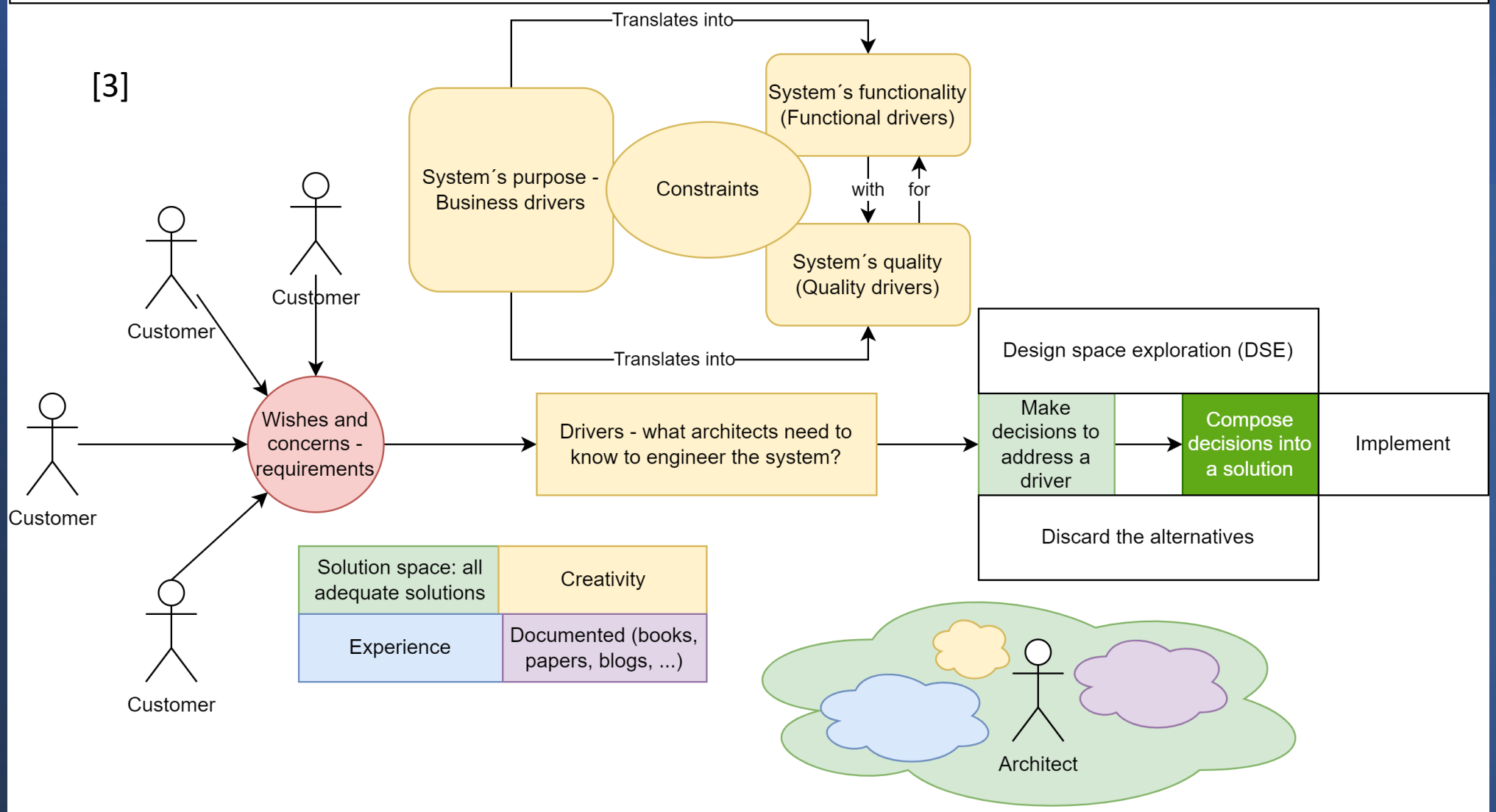
Make all these actions **explicit**…

Impractical but relevant – V model

[2]



Process and methodology



Document

SPECIFICATION OF ARCHITECTURE DRIVERS

Business

Natural language

Links to documents

Increase sale for 15%.

Increase a reputation.

A unique functionality.

Functionality

Use Cases

User Stories / Epics

Template scenario

User registration.

Web shop.

Constraints

Natural language

Use open source.

Use Android.

Do not use QR codes.

Quality

Template scenario

Performance, Maintainability, Extendibility, Safety, Security, Accessibility, Deplorability, Reliability, Scalability

[3]

QUALITY PROPERTIES TEMPLATE

ID	Unique identifier	**Status**	[Open, Defined, Solved, …]
Name	Name of scenario	**Owner**	Responsible for the scenario
Quality	Related quality attribute: exactly one attribute should be chosen.	**Stakeholders**	Stakeholders involved
		Quantification	
Environment	Context applying to this scenario. May describe both context and status of the system.		
Stimulus	The event or condition arising from this scenario.		
Response	The expected reaction of the system to the scenario event.		

[3]

DESIGN DECISION TEMPLATE

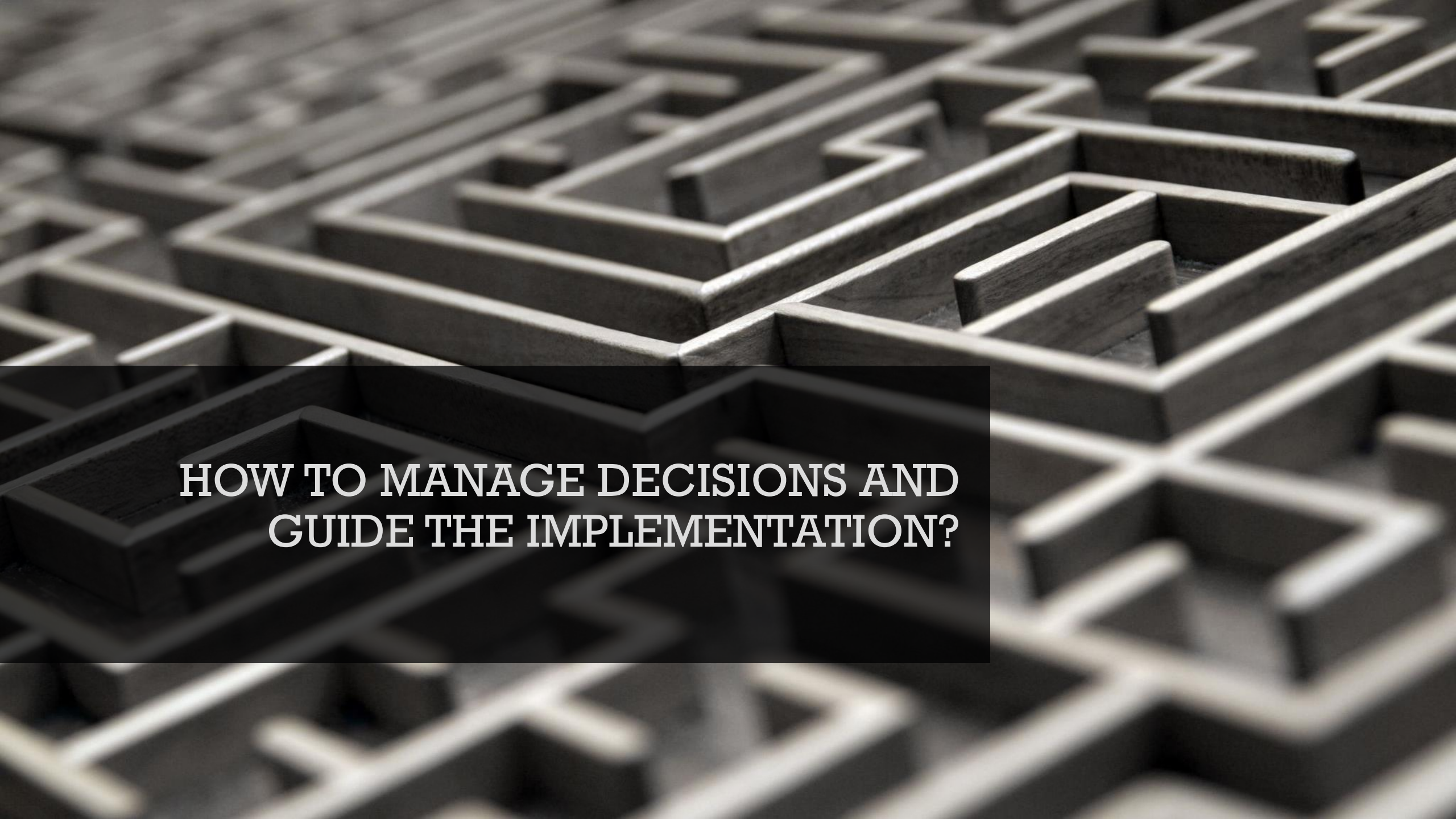
Decision name	
Decision ID	
Description	… …
Rationale (Pros, Advantages)	**Assumptions & Risks (Constraints)**
…	…
Scaling Factors	**Trade-offs**
…	…

[3]

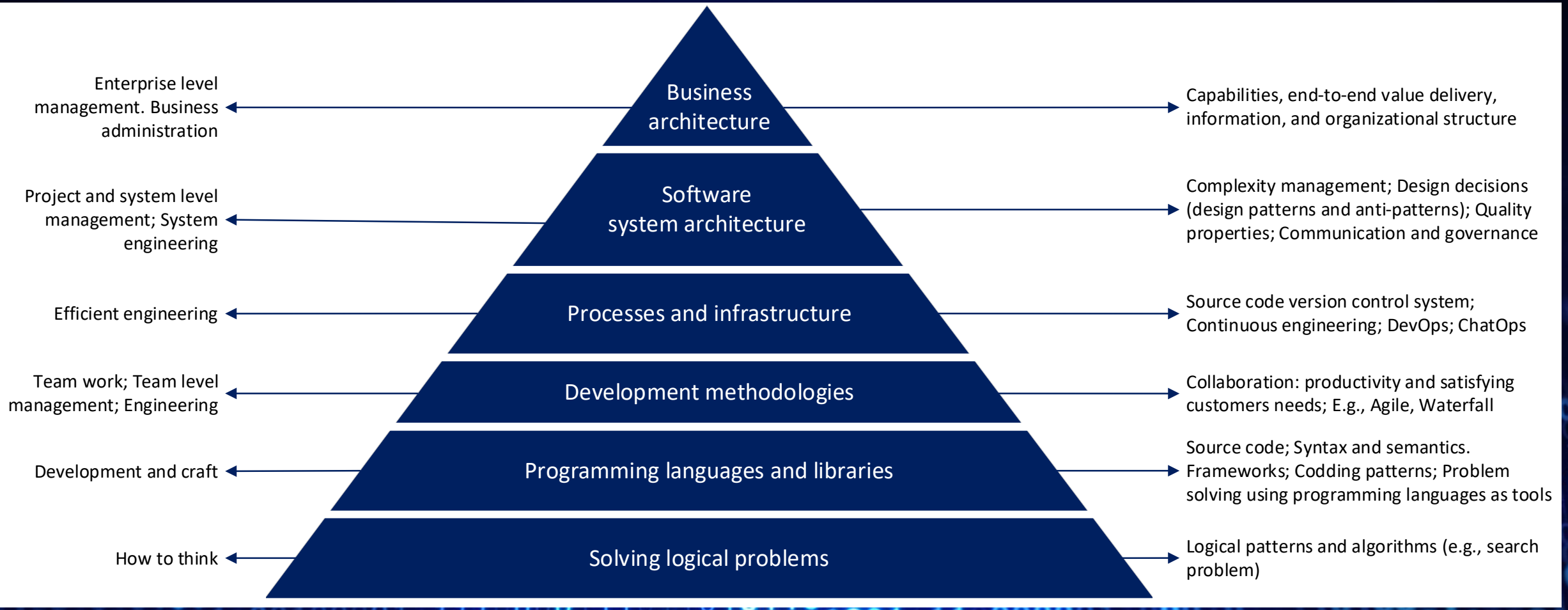
ARCHITECTURE SOLUTION TEMPLATE

Driver name		
Driver ID		
Steps	… The steps necessary to fulfil the scenario….	
Decisions	Accepted	Rejected

Rationale (Pros)	**Assumptions and Risks (Cons)**
…	…
Scaling Factors	**Trade-offs**
…	…



HOW TO MANAGE DECISIONS AND GUIDE THE IMPLEMENTATION?



SOFTWARE SYSTEM ARCHITECTURE

Decisions

ABSTRACTION IS THE KEY

ARCHITECTURE COMPONENTS

- "In the Component-based Software Engineering (CBSE) discipline, components are seen as standalone service providers, being, therefore, more abstract than objects and classes." (Bass, Clements, & Kazman, 1997), (Garlan, Monroe, & Wile, 2000), (Medvidovic & Taylor, 2000), (Roshandel, Schmerl, Medvidovic, Garlan, & Zhang, 2004).
- "A component is a software element that conforms to a component model and can be independently deployed and composed without modification according to a composition standard." (Heineman & Councill, 2001).
- "A component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third-parties." Szypersky (Szyperski C. , 2002).

ARCHITECTURE COMPONENTS

- Stand alone service providers
- Independent deployment
- Independent executable entities
- Explicit context dependencies (if any)

- More abstract than classes and objects.
- Should be reused
- Components are software elements that conform to a component model

- Have interfaces

COMPONENT INTERFACES

- "A collection of service access points, each of them including a semantic specification" (Bosch, 2000)
- "Mechanisms to define assembly constraints in the part model before assembling the component into the assembly" (Smith, 2004)
- "An entity provided or realized by a component, which comprises a set of operations performed by a hardware or software element in the system." (IBM, 2012)

COMPONENT INTERFACES

- Service access point
- Assembly constraint
- Set of operations performed by a hardware or software element in a system.
- Coupling

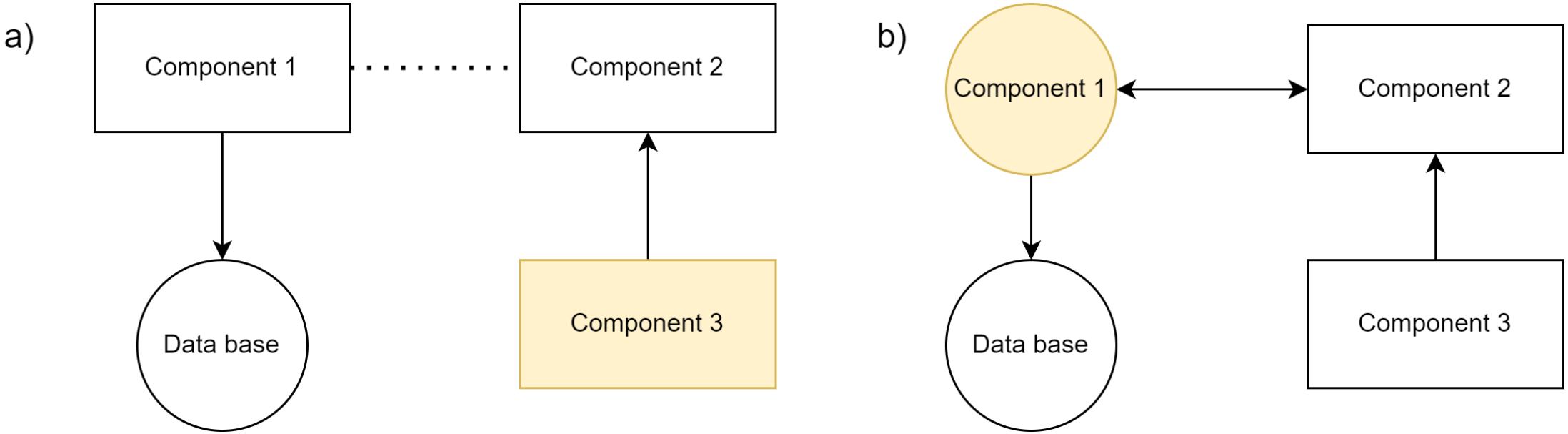
- Interface types:
  - Provided interface
  - Required interface

TEXT IS NOT THE BEST ABSTRACTION

Decisions

Visualisation

Visual documentation – what is the difference?



Modelling profiles

- Visual elements
- Their definitions
- Colours, symbols, size, position, etc.

- Example: Human-computer interaction

- UML?

Diagram(s) using the modelling profile

ONE, TWO, FIVE?

WHO WILL USE IT?

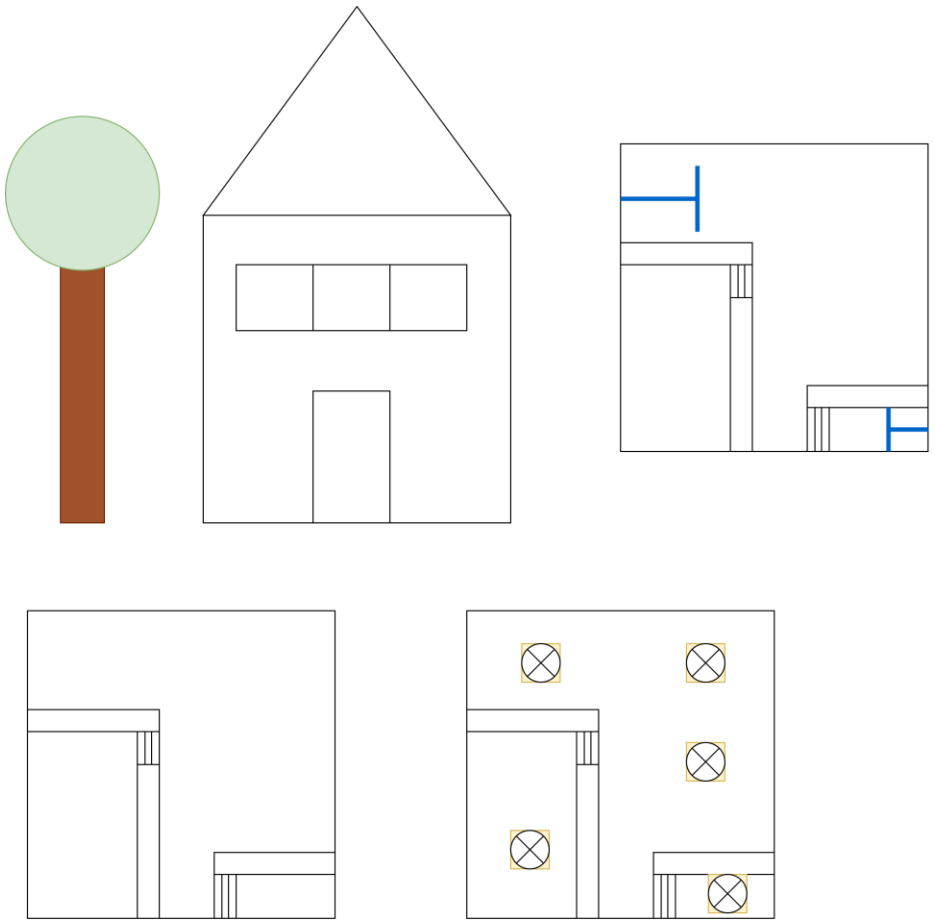
WHY?

Decisions

Visualisation

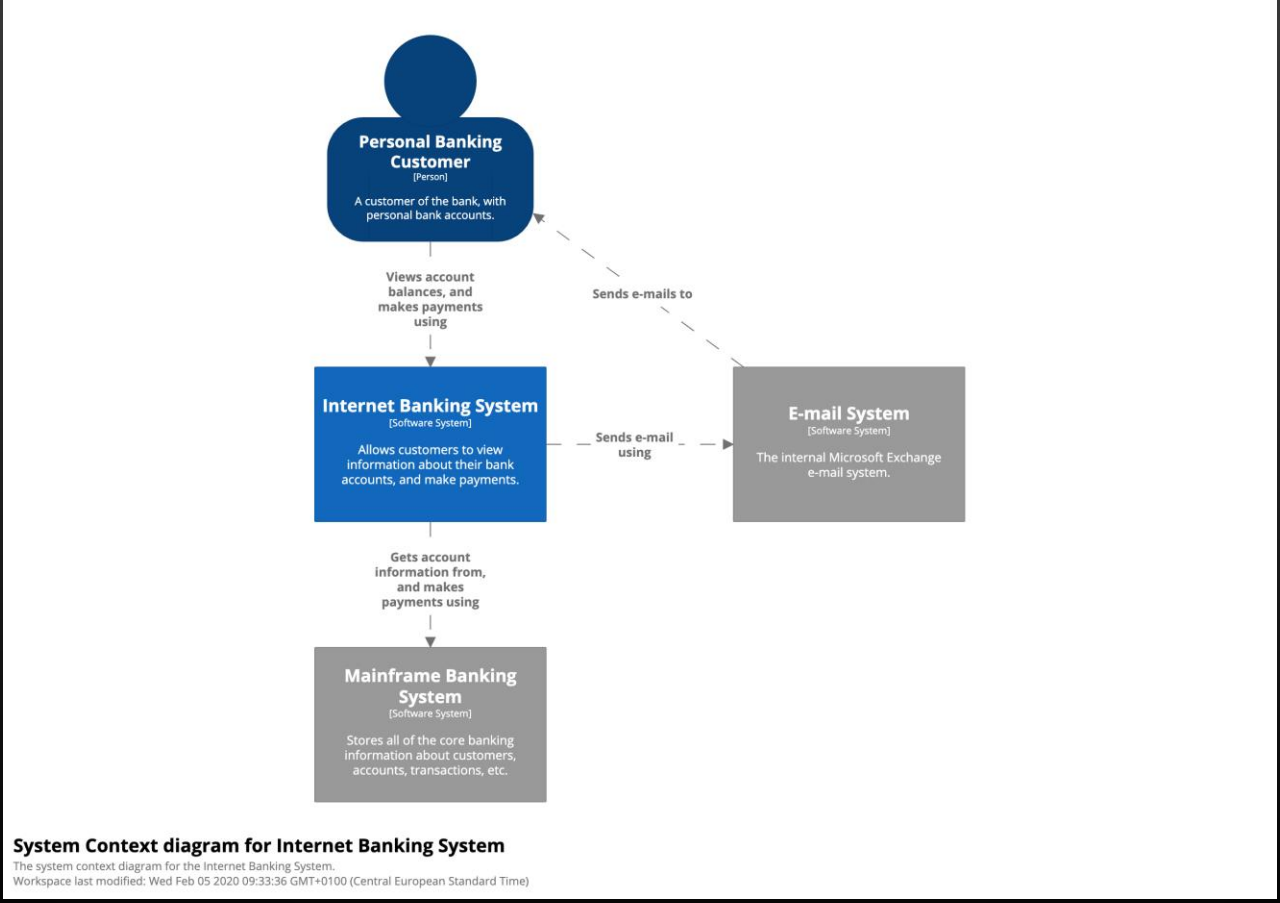
Perspective

Architectural views [4]



[5]

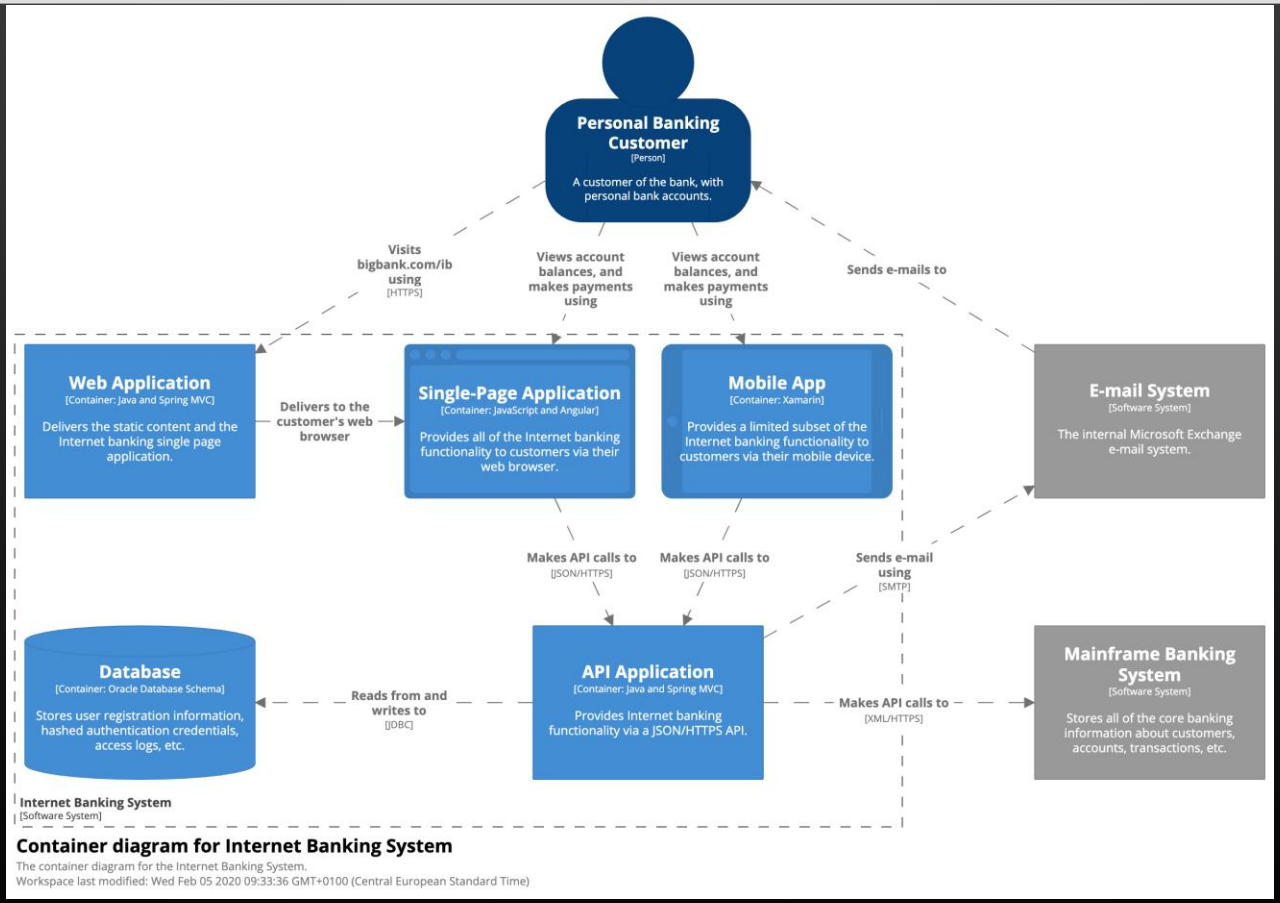
C4 ARCHITECTURE VIEWS - CONTEXT



https://c4model.com/

[5]

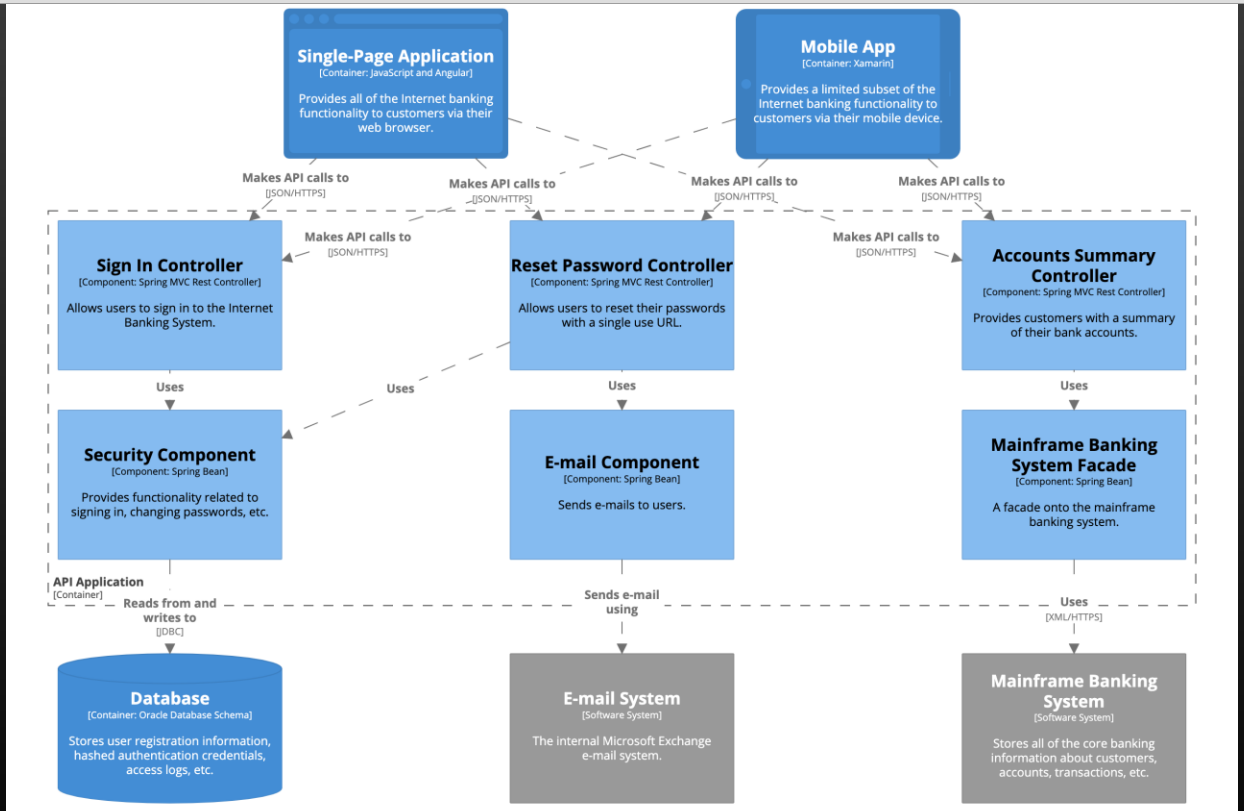
C4 ARCHITECTURE VIEWS - CONTAINER



https://c4model.com/

[5]

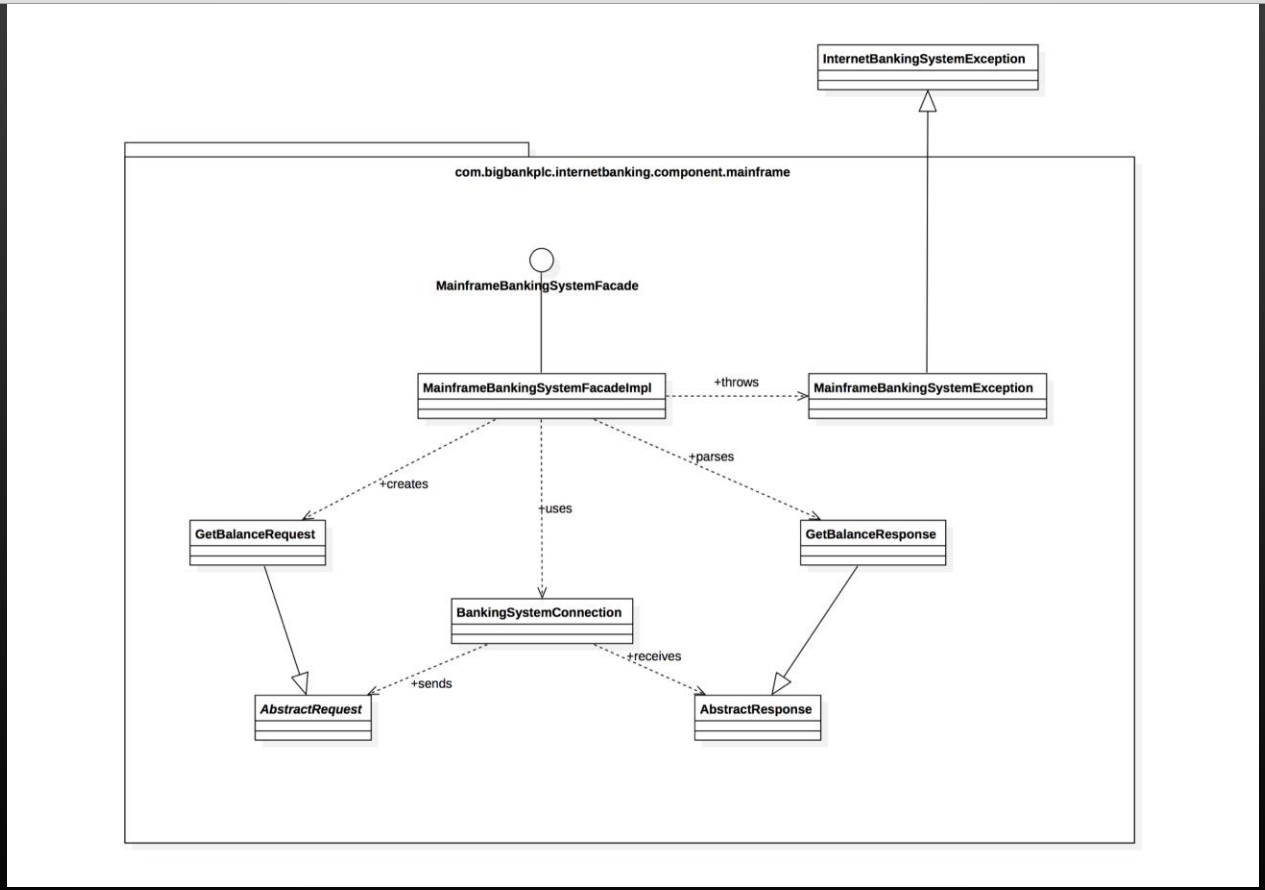
C4 ARCHITECTURE VIEWS - COMPONENT



https://c4model.com/

[5]

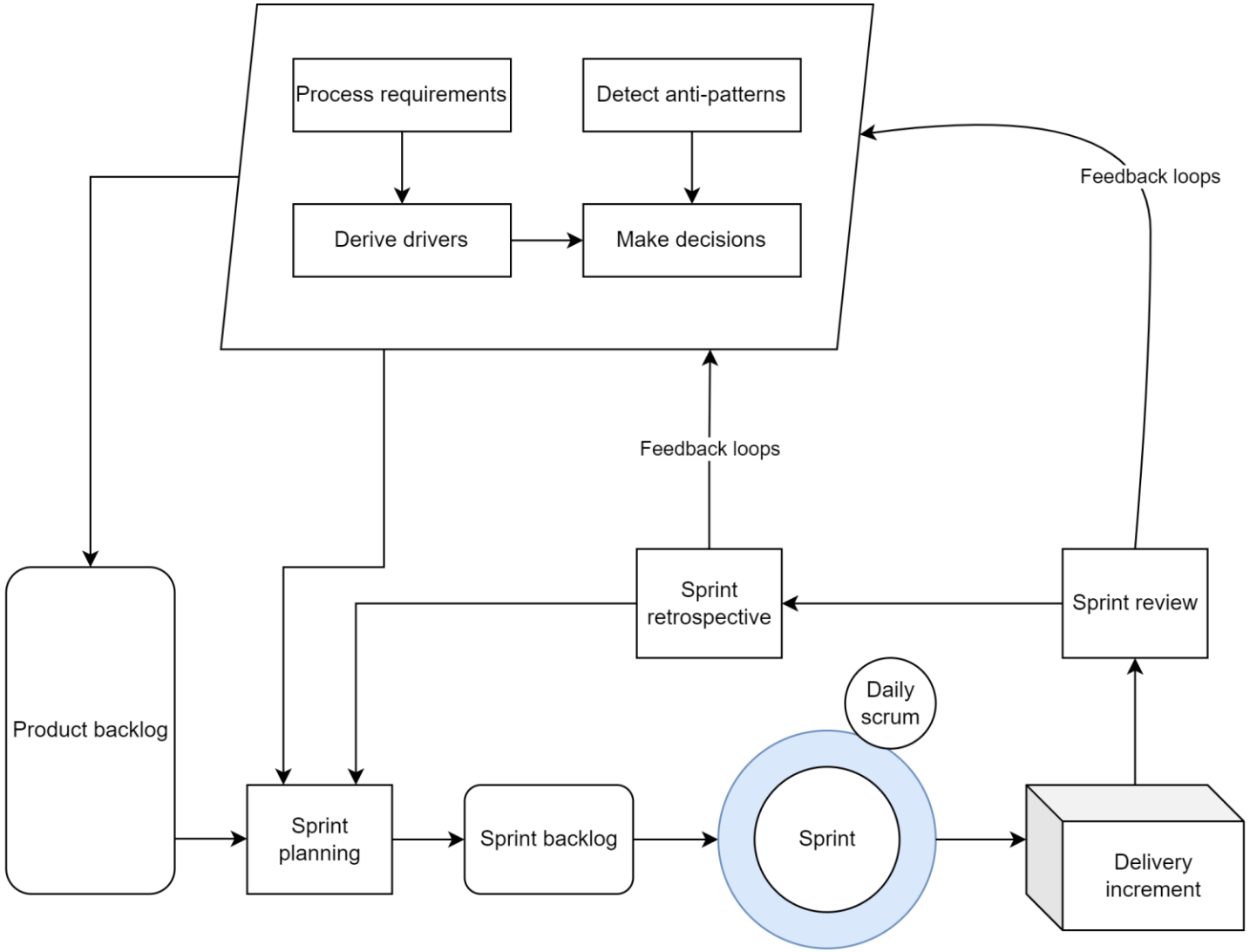
C4 ARCHITECTURE VIEWS - CODE



https://c4model.com/

[6] [7] [8]

A process



Explicit software architecture sounds great!

Who does this? Almost no one.

Overhead (documenting abstractions and processes)

- Creating the first version (including the setup and templates).
- Maintenance effort:
  - Update as the implementation/process changes.
  - Ensure consistency (especially problematic in large documents)
- Consuming the documentation:
  - How easy it is to find the information that stakeholders need?
  - How easy it is to understand the information?
- Follow a process
- **Overhead vs benefits?**

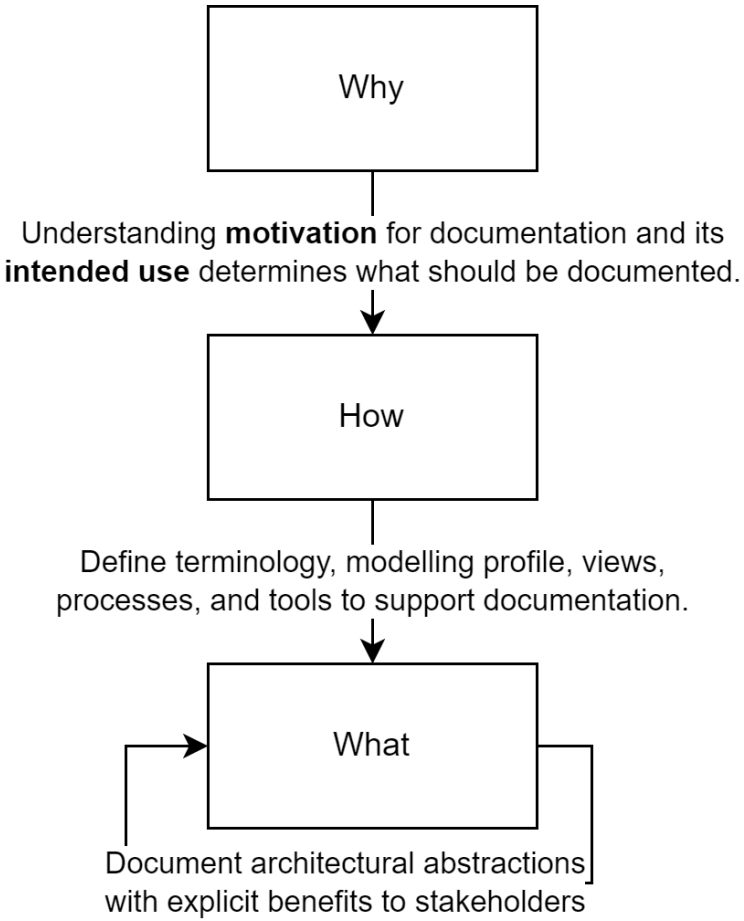
Actually, about that coding Margareth…

Explicit vs Implicit architecture

- We always do the same set of things (either explicitly or implicitly)
  - Problem
  - Solution
  - Coding
  - Process
  - …
- And we can fail in both cases (in a different way)

Customization

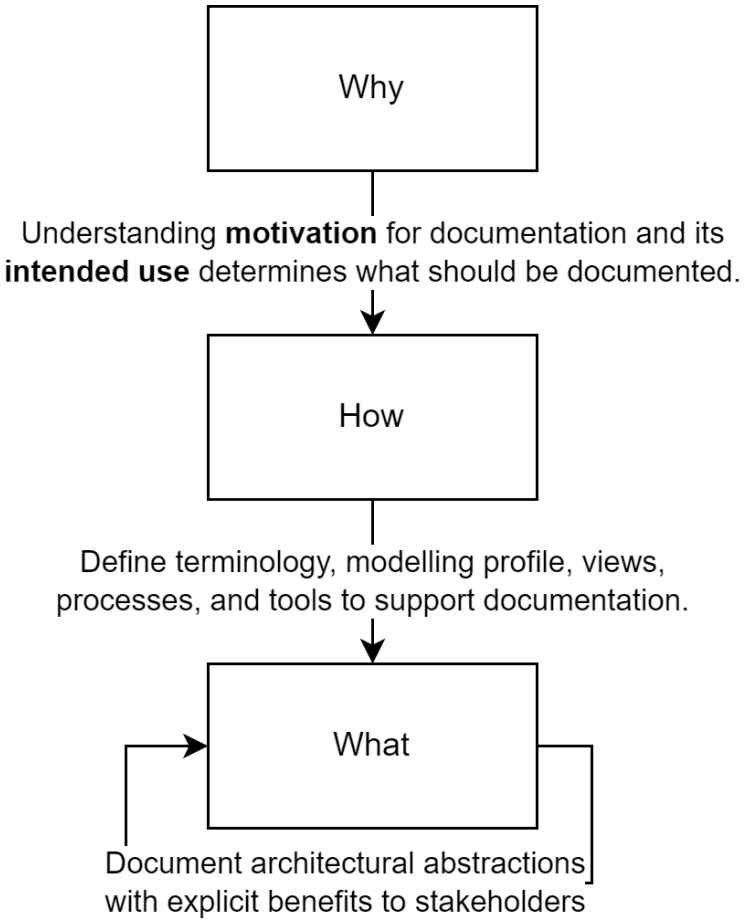
What are we trying to solve? Why are we trying to make something explicit?



Why?

Motivation	E.g., onboarding new members.
Producer	E.g., Team lead and a senior software engineer
Consumer	E.g., New team member
What would consumer gain from this?	E.g., setup IDE, setup plugins, download code.
To what benefit would consumers use this knowledge?	E.g., ready to compile code, run tests.

What are we trying to solve?

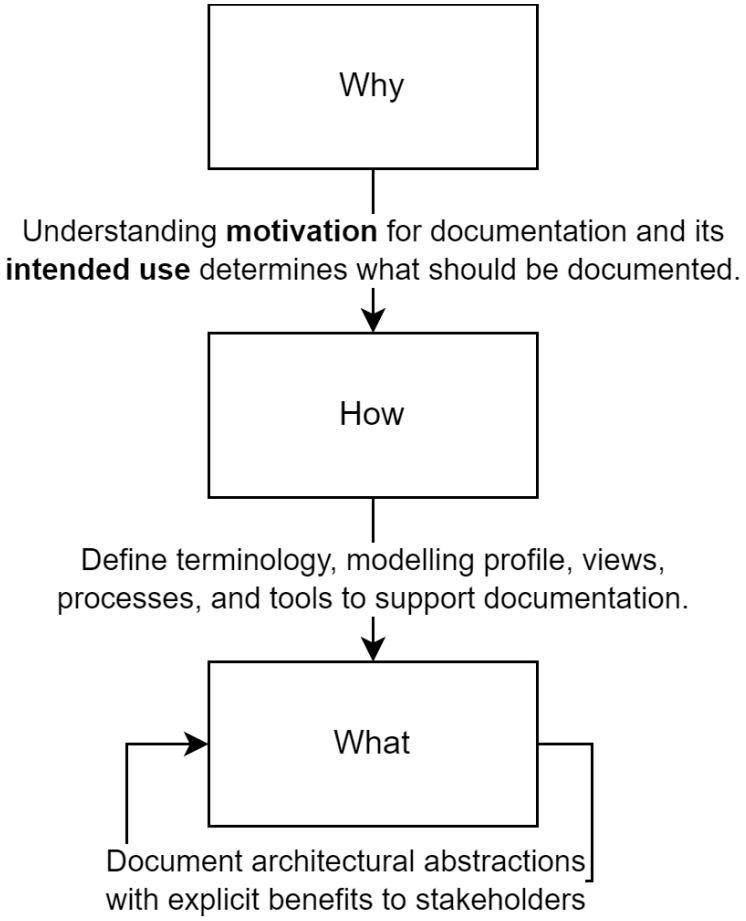


How? (considering why)

- Modelling profile
- Views
- Process for ensuring traceability and consistency
- Process for introducing changes
- Tools
- Documentation formats

What are we trying to solve?

Motivation	E.g., onboarding new members.
Producer	E.g., Team lead and a senior software engineer
Consumer	E.g., New team member
What would consumer gain from this?	E.g., setup IDE, setup plugins, download code.
To what benefit would consumers use this knowledge?	E.g., ready to compile code, run tests.




Modelling profile


Views


Process for ensuring traceability and consistency


Process for changing the process


Tools


Documentation formats

Cost/benefits analysis

Understand the need

Customize existing practices

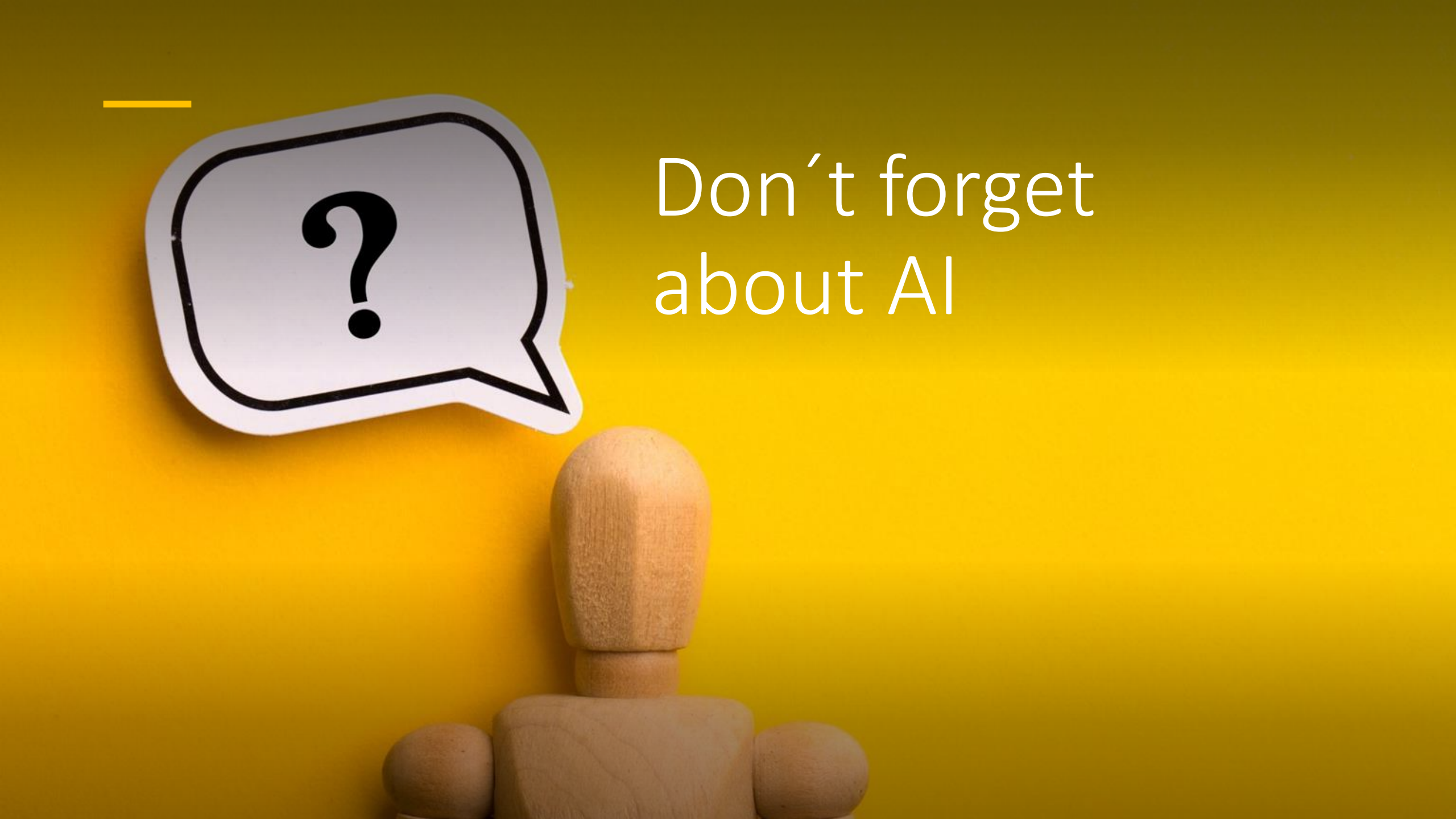
Maximize benefits/minimize overhead

Oh wait, what is software architecture?

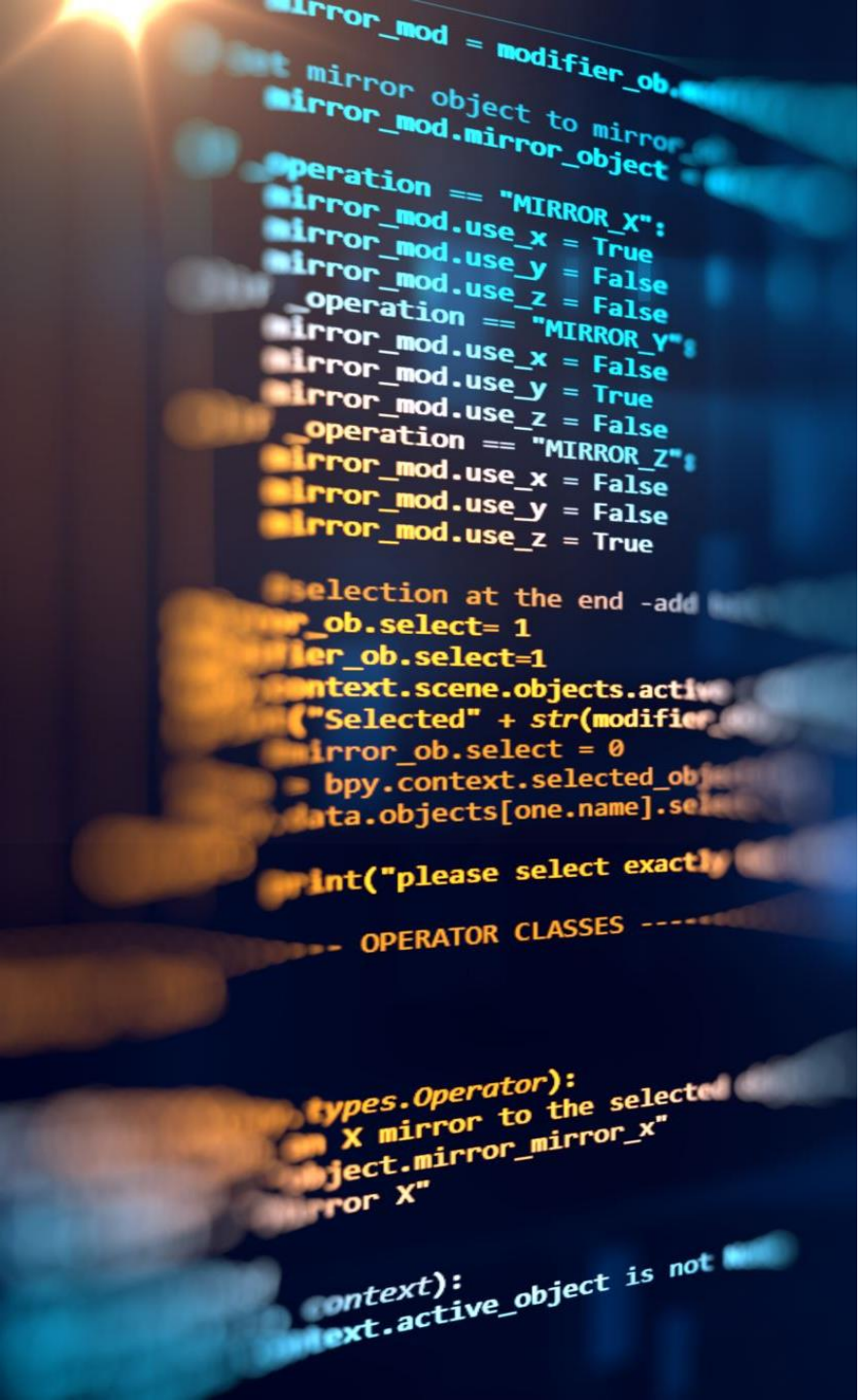
It depends…

Software architecture

- “The software architecture of a system is the set of structures needed to reason about the system. These structures comprise software elements, relations among them, and properties of both.”, Bass, Clements, & Kazman, Software Architecture in Practice, Fourth Edition, 2021
- “Software Architecture is the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution” (ISO, 2011)
- “Software architecture is the set of components needed to reason about the system, design decisions behind those components, and their discarded design alternatives.”, Jasmin Jahić



Don´t forget about AI

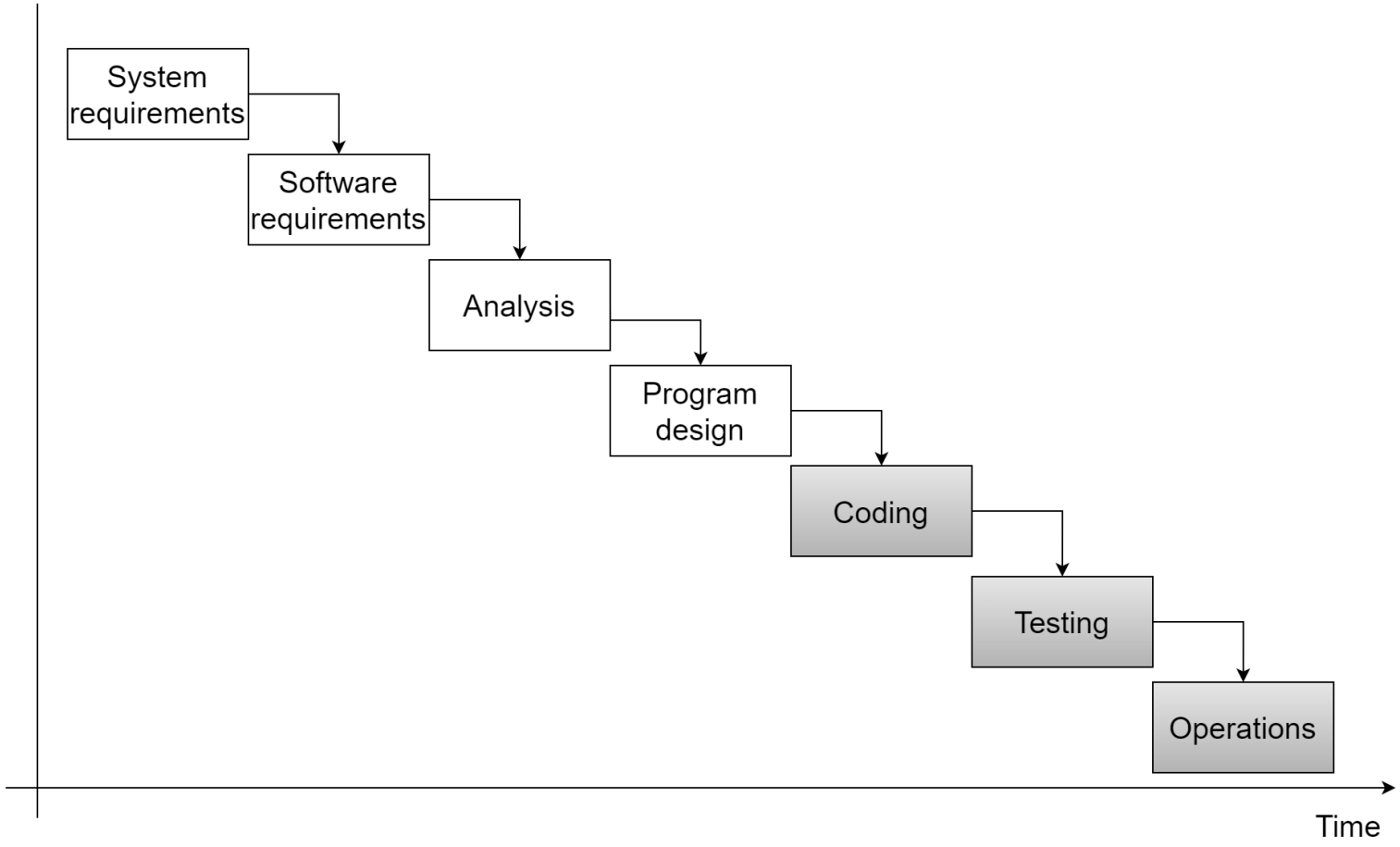


AI in Software Engineering

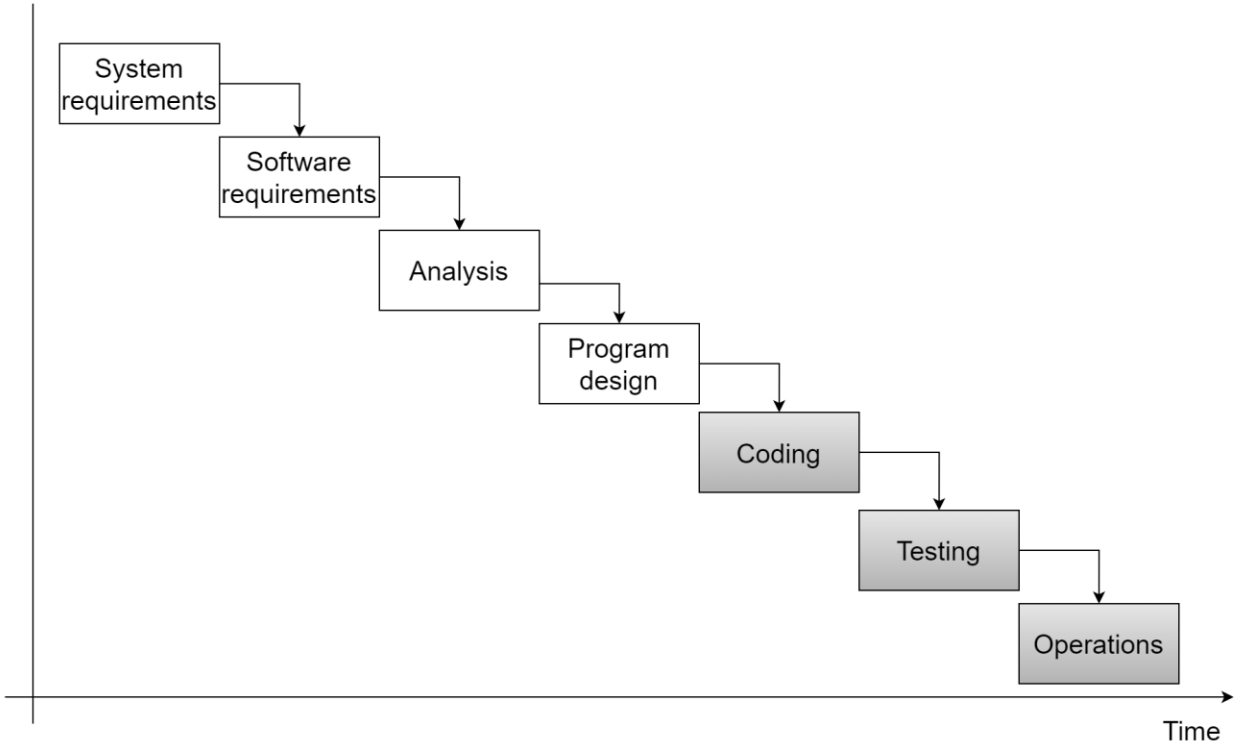
- Software Architecture
  - Need to understand context around problems and solutions
- Coding
- Testing
- Deployment
- Delivery
- Other ops

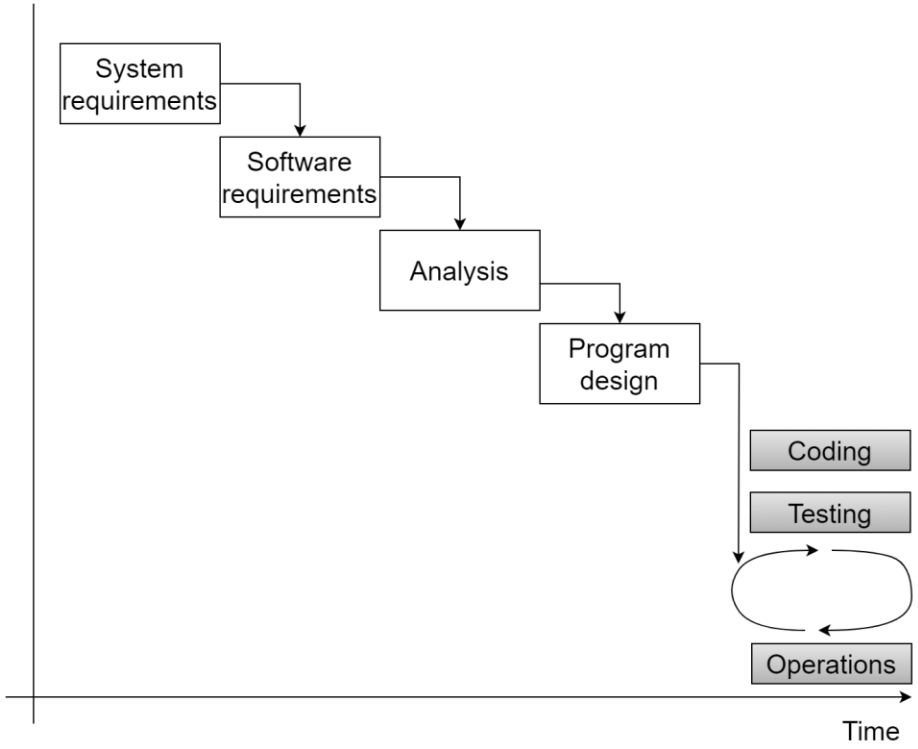
How a t-shirt stopped this autonomous car in its tracks - https://www.carexpert.com.au/car-news/how-a-t-shirt-stopped-this-autonomous-car-in-its-tracks

Waterfall

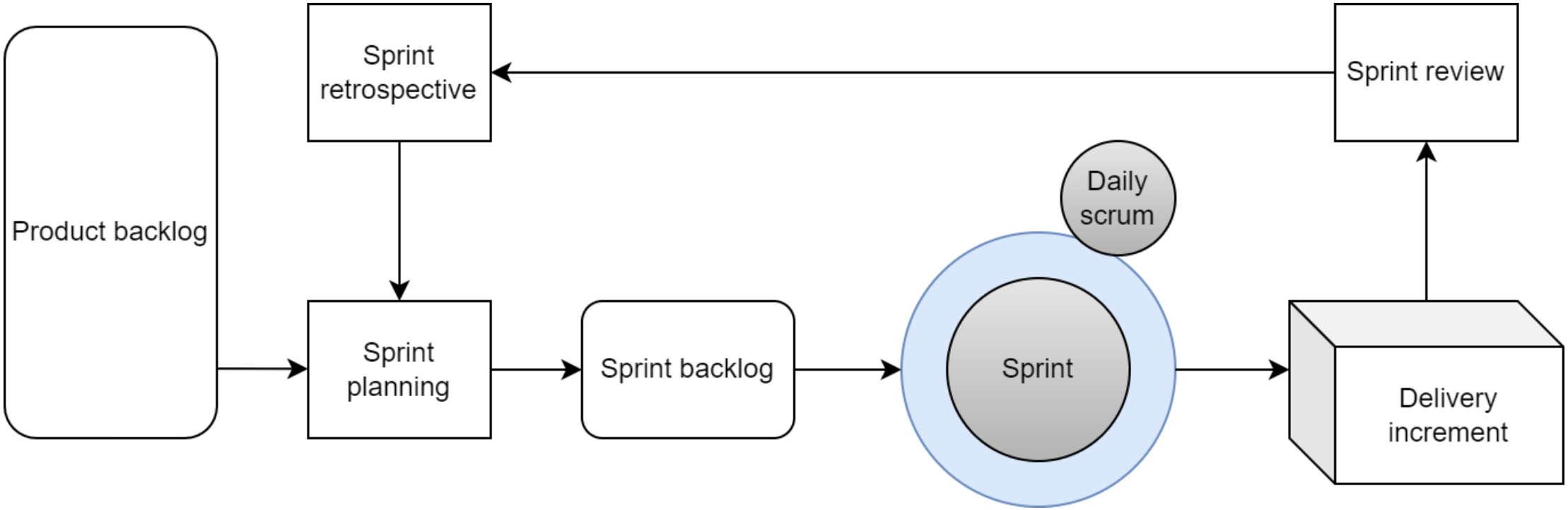

System requirements
Software requirements
Analysis
Program design
Coding
Testing
Operations
Time

Waterfall in the Age of AI

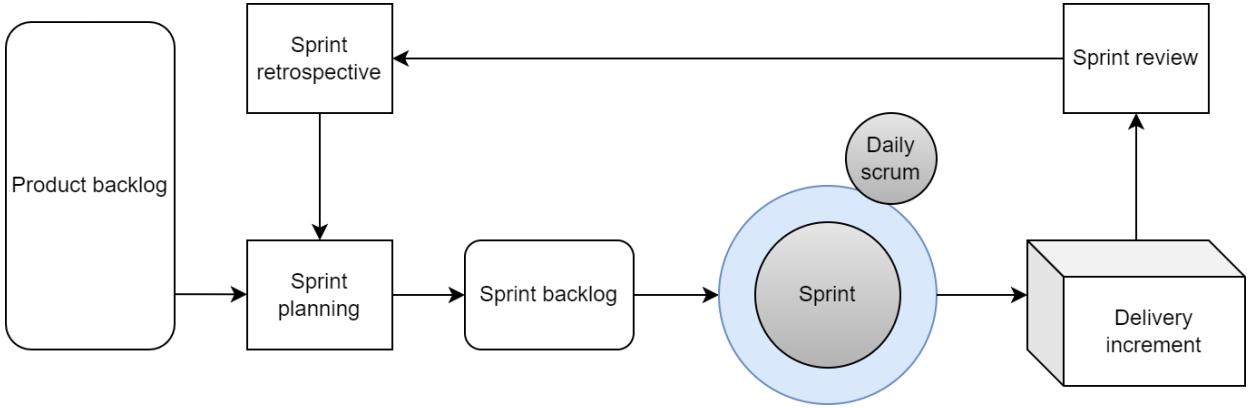


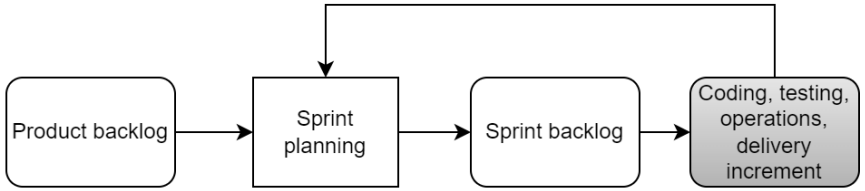


Agile Scrum

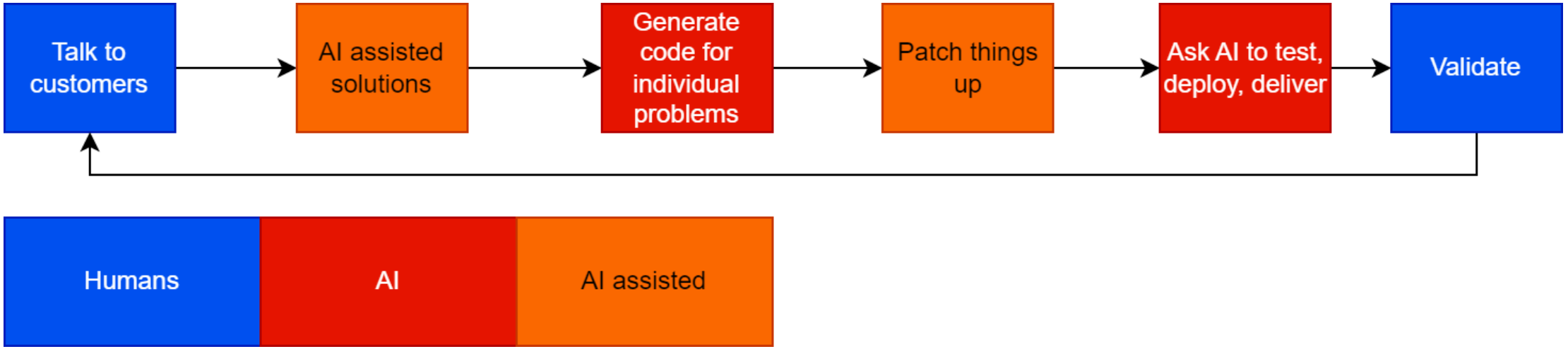

Product backlog
Sprint retrospective
Sprint review
Daily scrum
Sprint planning
Sprint backlog
Sprint
Delivery increment

Agile Scrum in the Age of AI





What is the Value that Humans bring?



I know how to code!

No one cares Margareth – we have AI

Typical Example of Digitalisation

Bakery – Two Pigeons

Bakery Digitalisation

Online presence

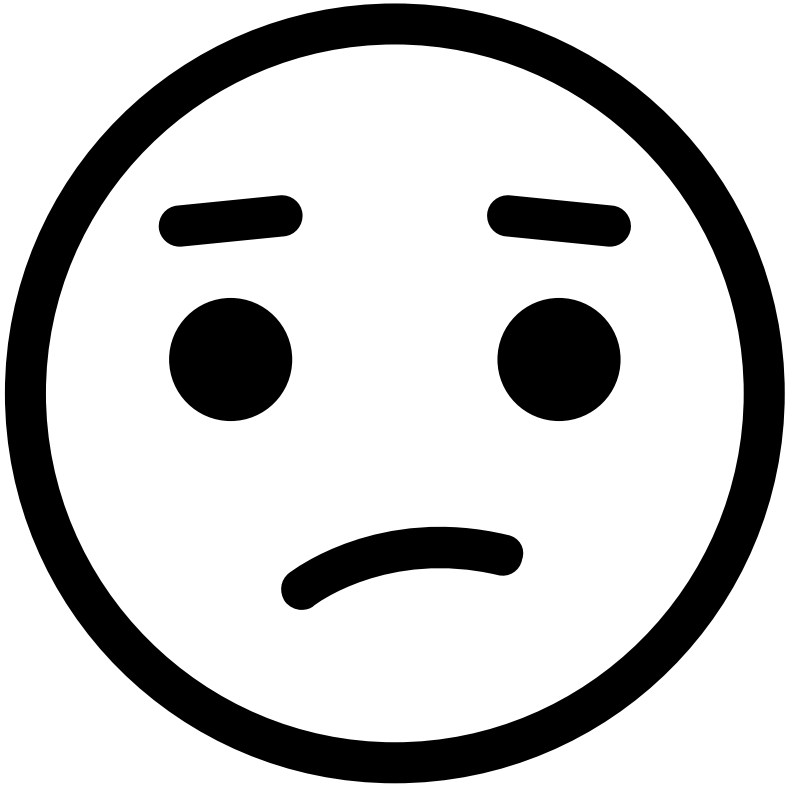
Website (presentation, orders)

Social media (updates)

Accounting software

How can AI help a bakery owner with digitalisation (in future)?

- Go to GitHub, there is a CoPilot
- It will generate all the code (Node.JS, JavaScript, can even create a database)
- Then, you just need to patch it (use ChatGPT) and deploy it
- Oh, you will need a server
- And hosting
- And perhaps a cloud-based solution



- „According to the German Bread Institute, over 3,000 different types of bread and other baked goods are sold in Germany every day.“

- https://www.germany.travel/en/experience-enjoy/german-bread-and-baked-goods.html#:~:text=German%20bread%20and%20baked%20goods&text=According%20to%20the%20German%20Bread,sold%20in%20Germany%20every%20day.

We know Software Engineering will change

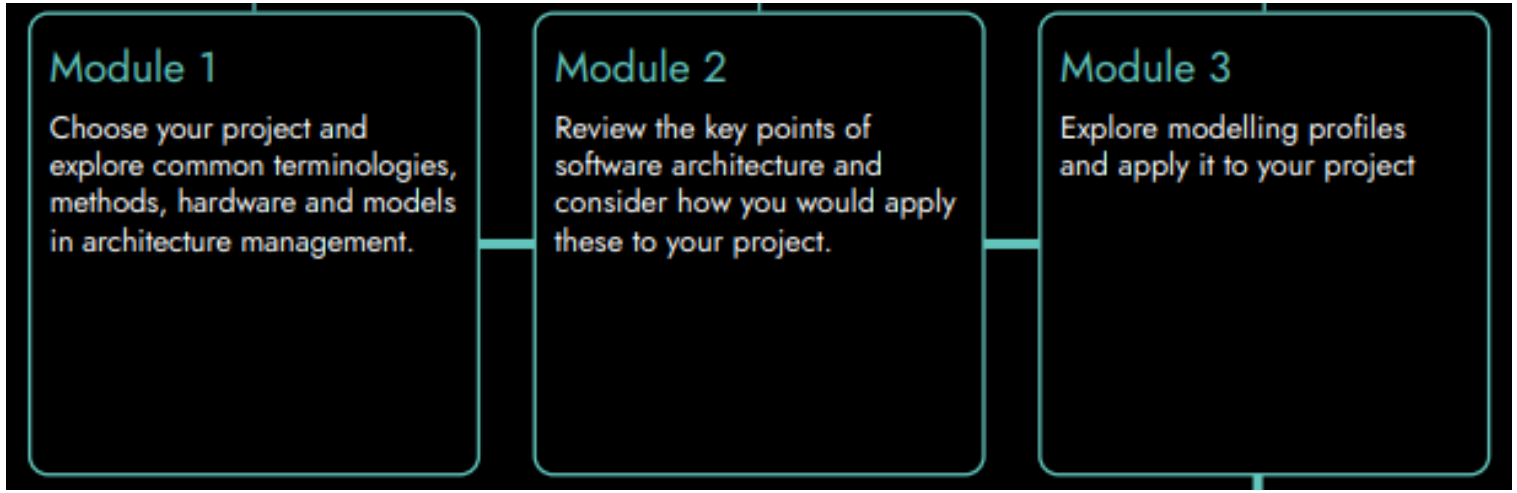
We just do not know how

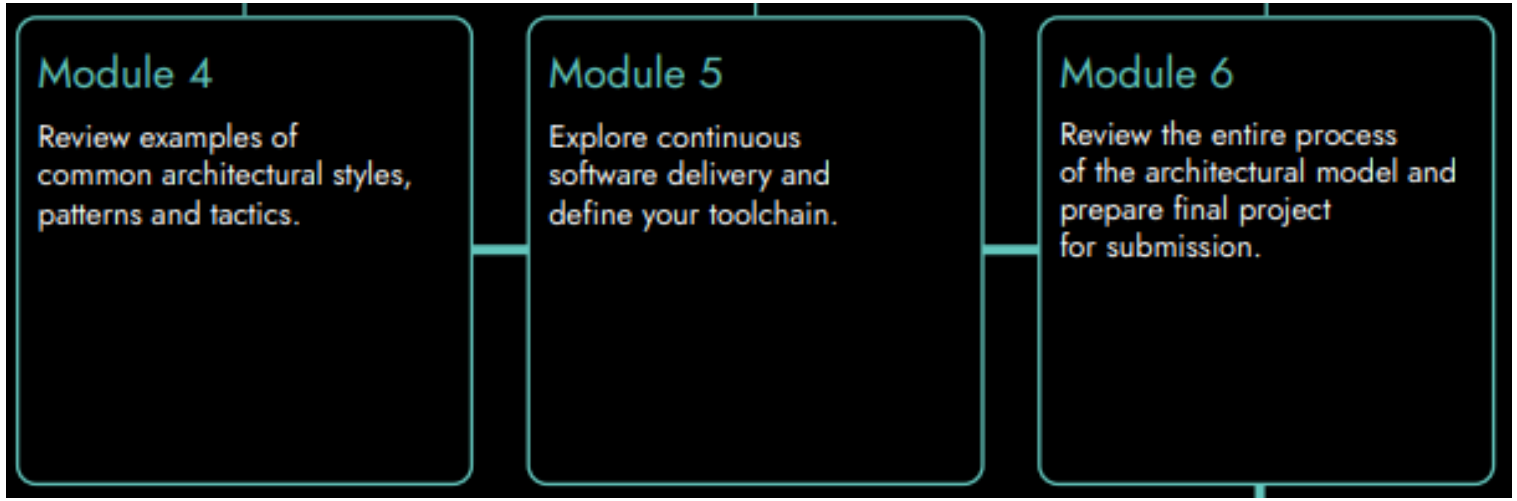
Ongoing Projects

- AI and Software Architecture: reusing solutions [9]
- Benefits and overheads: AI vs human, a coding project.
- AI in Software Engineering manifesto: what do we hope to gain from AI in Software Engineering?
- jj542@cam.ac.uk

Managing Software Architecture

- https://advanceonline.cam.ac.uk/courses/managing-software-architecture





References

- [1] K. Smolander, "Four metaphors of architecture in software organizations: finding out the meaning of architecture in practice," Proceedings International Symposium on Empirical Software Engineering, Nara, Japan, 2002, pp. 211-221, doi: 10.1109/ISESE.2002.1166942.
- [2] Concept of Operations, Leon Osborne, Jeffrey Brummond, Robert Hart, Mohsen (Moe) Zarean Ph.D., P.E, Steven Conger, example https://web.archive.org/web/20090705102900/http://www.itsdocs.fhwa.dot.gov/jpodocs/repts_te/14158.htm
- [3] Jens Knodel and Matthias Naab. 2016. Pragmatic Evaluation of Software Architectures (1st. ed.). Springer Publishing Company, Incorporated.
- [4] Philippe Kruchten: The 4+1 View Model of Architecture. IEEE Softw. 12(6): 42-50 (1995)
- [5] https://c4model.com/
- [6] Continuous Architecture in Practice: Software Architecture in the Age of Agility and DevOps, Murat Erder, Pierre Pureur, Eoin Woods
- [7] F. Helwani and J. Jahić, "ACIA: A Methodology for identification of Architectural Design Patterns that support Continuous Integration based on Continuous Assessment," 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), Honolulu, HI, USA, 2022, pp. 198-205, doi: 10.1109/ICSA-C54293.2022.00046.
- [8] Brian Fitzgerald, Klaas-Jan Stol, Continuous software engineering: A roadmap and agenda, Journal of Systems and Software, Volume 123, 2017
- [9] Jasmin Jahic, "State of Practice: LLMs in Software Engineering and Software Architecture", International Conference on Software Architecture (Hyderabad, India) (ICSA 2024)

Questions