Utrecht University

Lecture 2: Modular Decomposition and Quality Attributes –I

Utrecht University

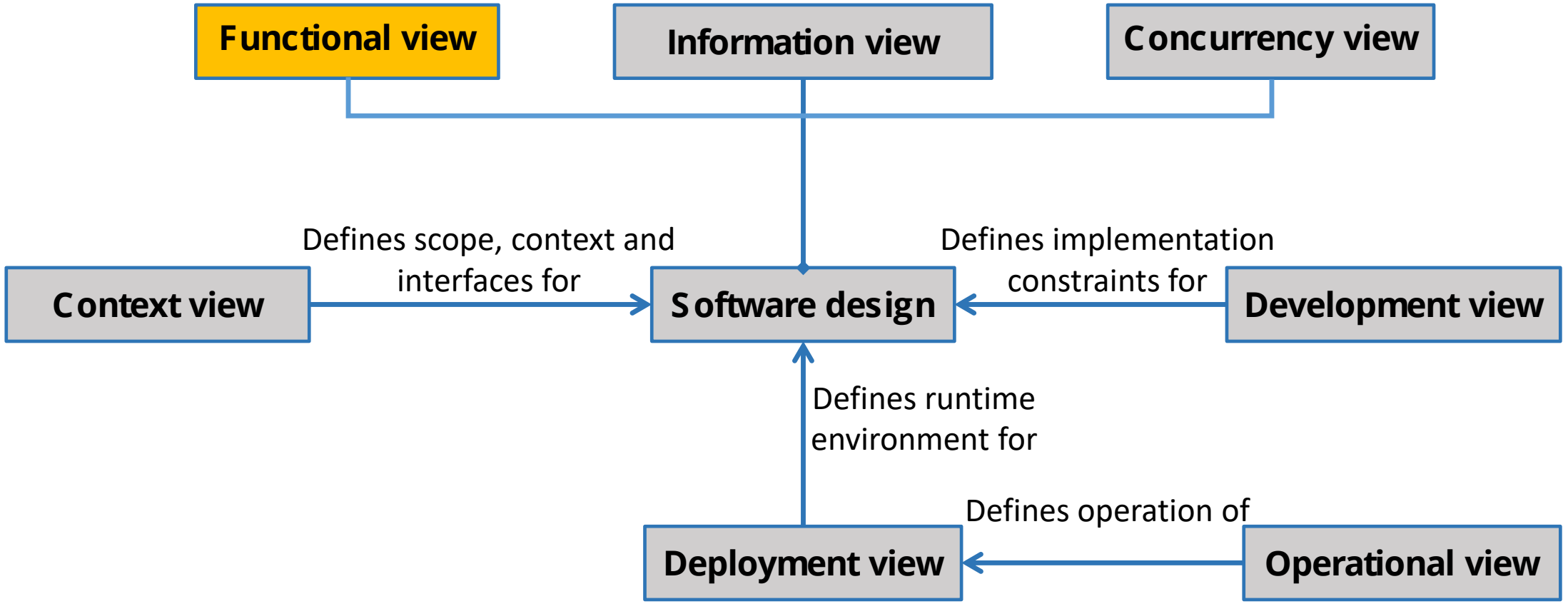
Agenda for today

- 13:15 – 13:45: Functional viewpoint
- 13:45 – 14:45: You: Functional architecture for TrIP
- 15:00 – 15:45: Quality attributes & tactics
- 15:45 – 16:45: Defining QAs in your assignment
- 16:45 – 17:00: Wrap-up

The functional viewpoint

Utrecht University

Viewpoint catalog



Viewpoint:

Collection of patterns, templates and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views

Utrecht University

McIlroy & Functional viewpoint

Utrecht University

McIlroy & Functional viewpoint

Exit

How to participate?

Utrecht University

Copy participation link

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
LLYESU

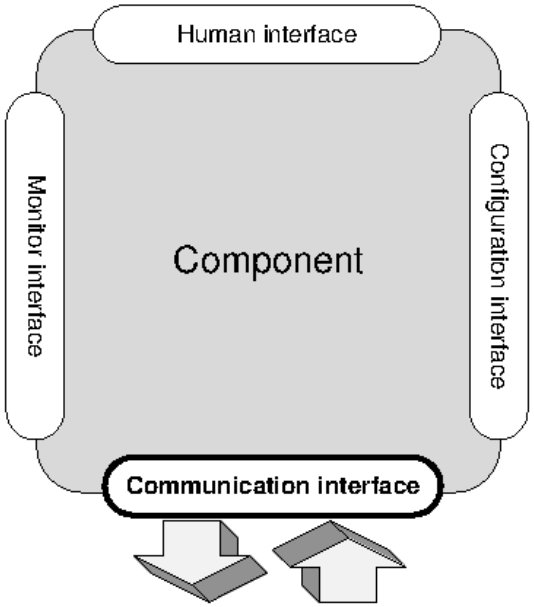
Enable answers by SMS

Utrecht University

(McIlroy, 1968): What did you notice?

Utrecht University

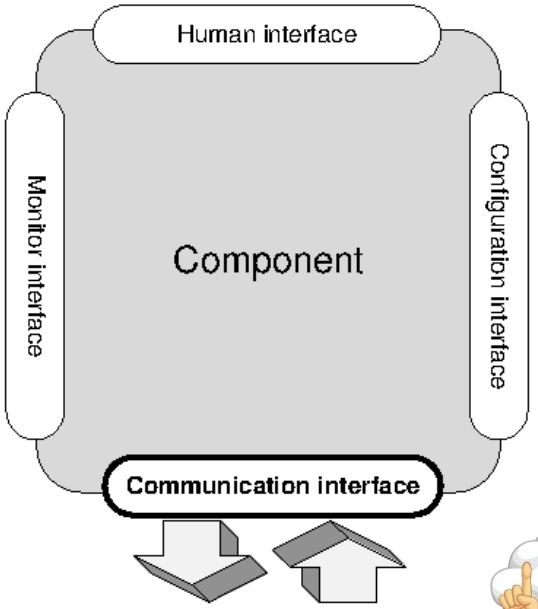
Software components



- Modularization
- Software components
  - **“Family of routines” ➔ A coherent set of functionality**
  - **Configurable (parameters)**
  - **Binding time (design time, run time)**
  - **Interfaces**
- Composition
  - **Treat component as black box**
  - **System sum of the components**

Utrecht University

Software components



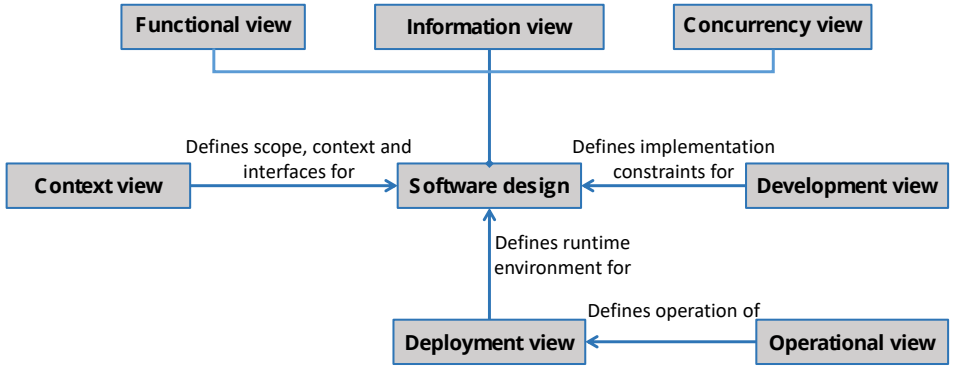
Four interfaces:

1. Human interface
   Interface with the user, typically a GUI
2. Configuration interface
   Parameters to deploy it in a specific context
3. Monitor interface
   Logging of the usage of the component
4. Communication interface
   How it communicates with other components
   Often an "Application Programming Interface"

Nomenclature:
if design time, we use the term *module*
If runtime, we use the term *component*

Utrecht University

Functional view



- Functional view:
  Describes the system's runtime functional elements and their responsibilities, interfaces and primary interactions
- Concerns
  Functional capabilities,
  External interfaces,
  Internal structure,
  Functional design philosophy
- Models and views
  Functional Architecture Model

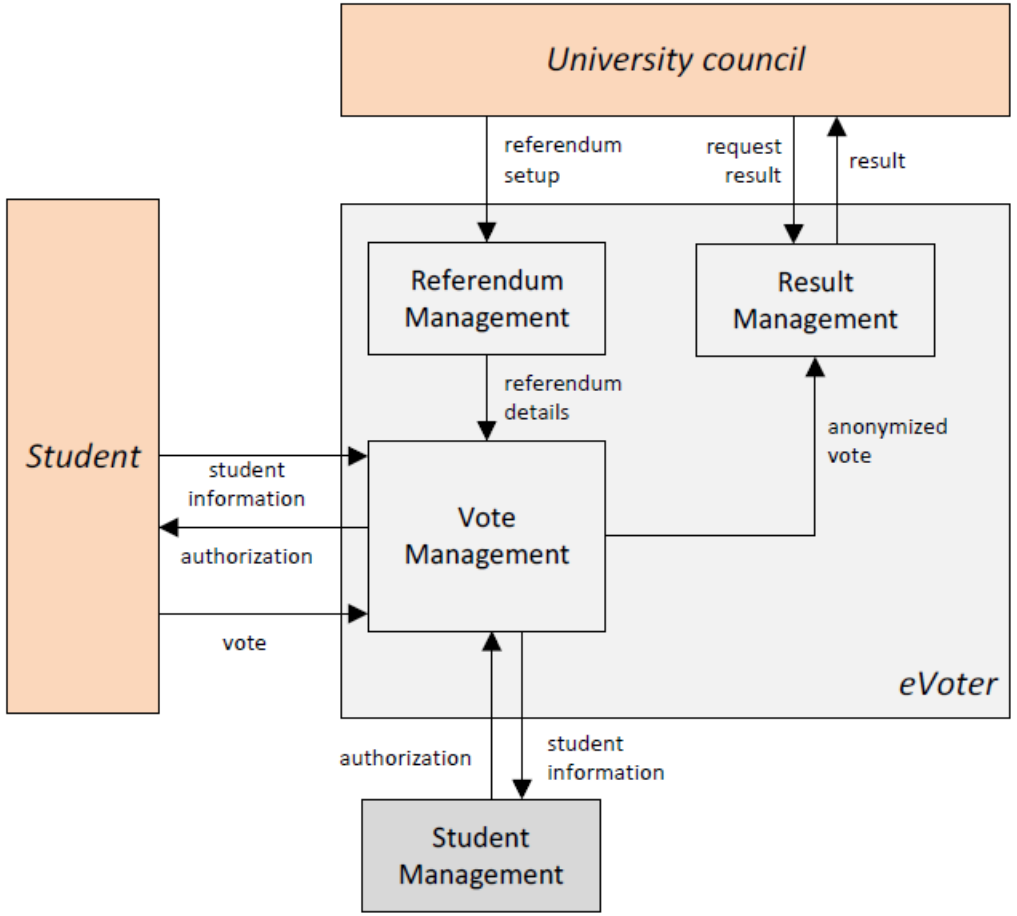
Utrecht University

Functional view – design philosophy

- What are the design principles of your system?
  - **Coherence**
  - **Cohesion**
  - **Consistency**
  - **Coupling**
  - **Extensibility**
  - **Flexibility**
  - **Interdependency**
  - **Separation of concerns**
  - **Simplicity**
- Architecting: trade-off between these principles!

Utrecht University

Functional Architecture Model

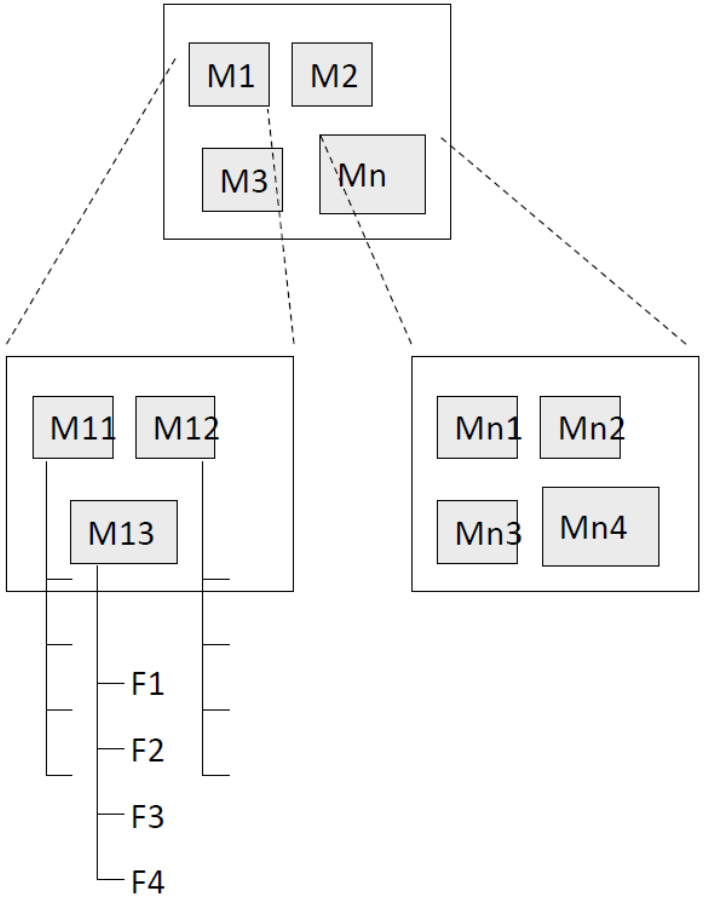


- Decomposition from a usage perspective
- Module
  Coherent set of features bundled within an application to organize its functionality
- Interaction flows
  Interactions between the features to realize some action (typically action as in epic / user story)

Brinkkemper S., Pachidi S. (2010) Functional Architecture Modeling for the Software Product Industry. In: Software Architecture. ECSA 2010. Lecture Notes in Computer Science, vol 6285. Springer, Berlin, Heidelberg.

Utrecht University

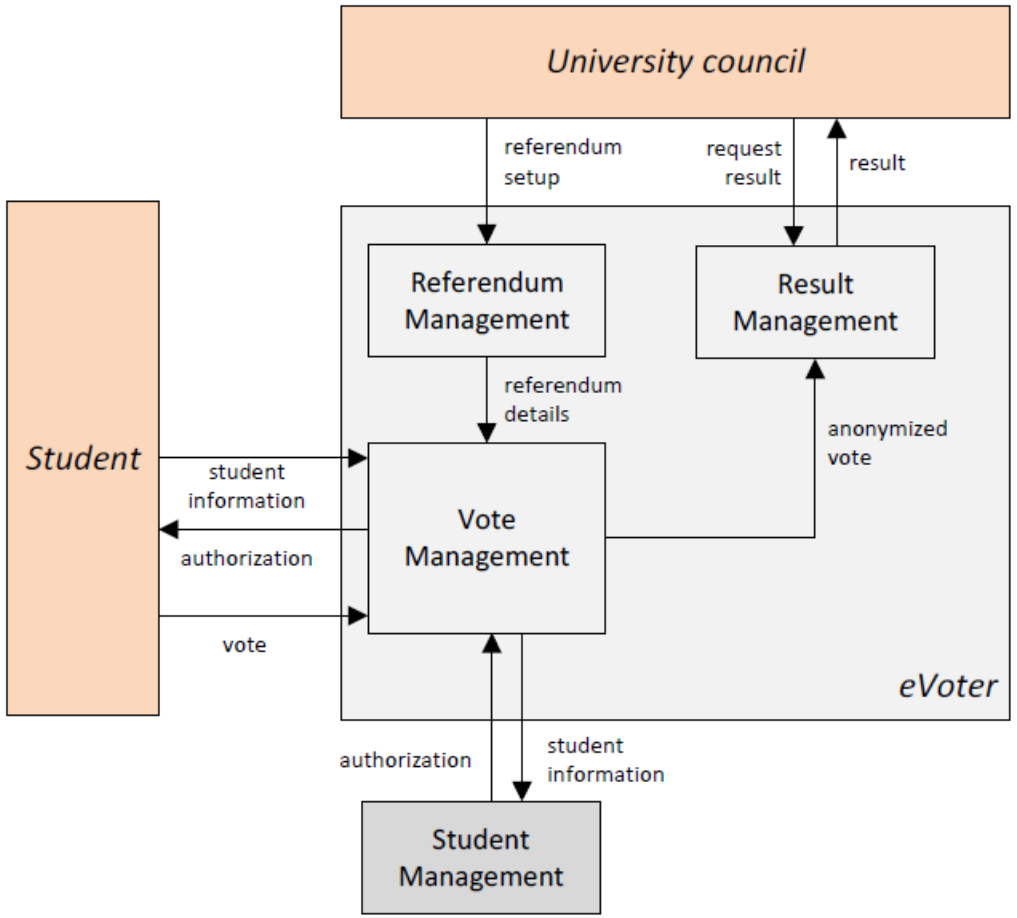
Functional Architecture Model



- Modules are hierarchical (tree)
  - **Typically 2 or 3 layers**
  - **Leaves of the tree: features**
- Feature:
  - **discrete unit of unique functionality of an application that delivers measurable benefit to the user**
  - **Can be organized in feature trees**

Utrecht University

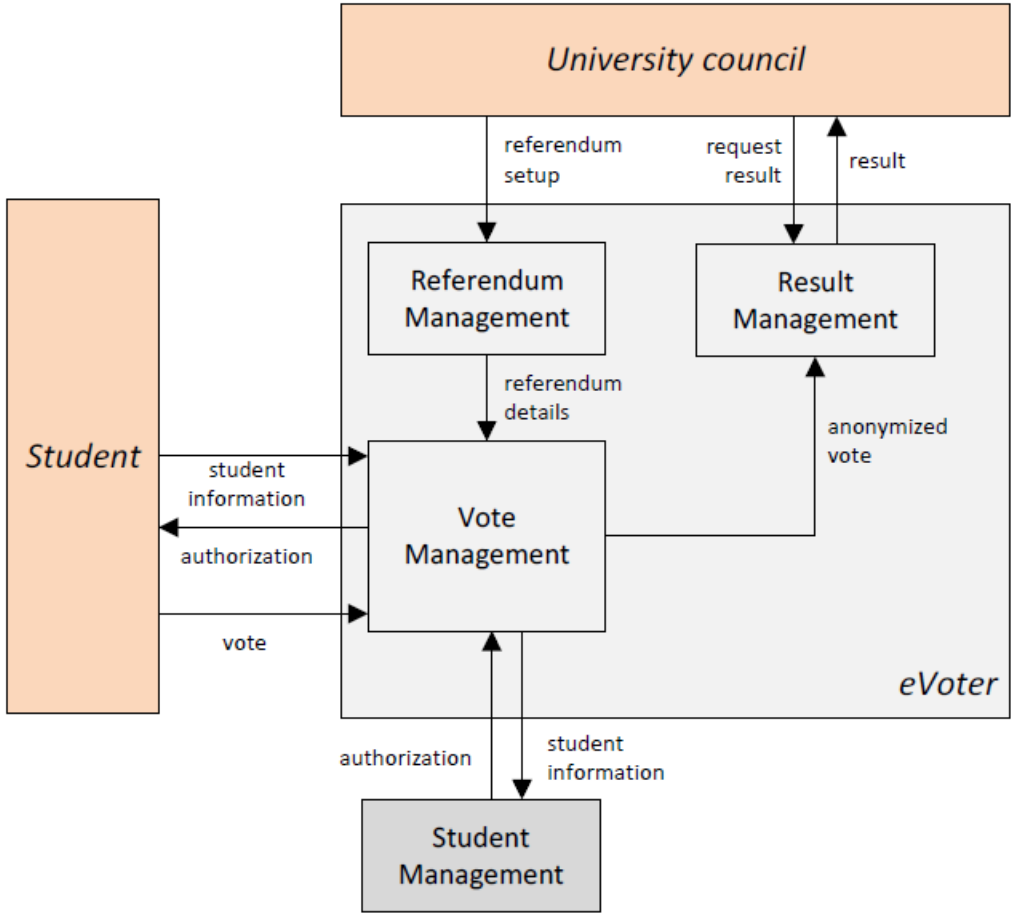
Functional Architecture Model



- Modules are hierarchical (tree)
  - **Typically 2 or 3 layers**
  - **Leaves of the tree: features**
- Feature:
  - **discrete unit of unique functionality of an application that delivers measurable benefit to the user**
  - **Can be organized in feature trees**

Utrecht University

User stories, Use cases and Functional Architecture



Remember: modules are hierarchical, lowest level is a feature!

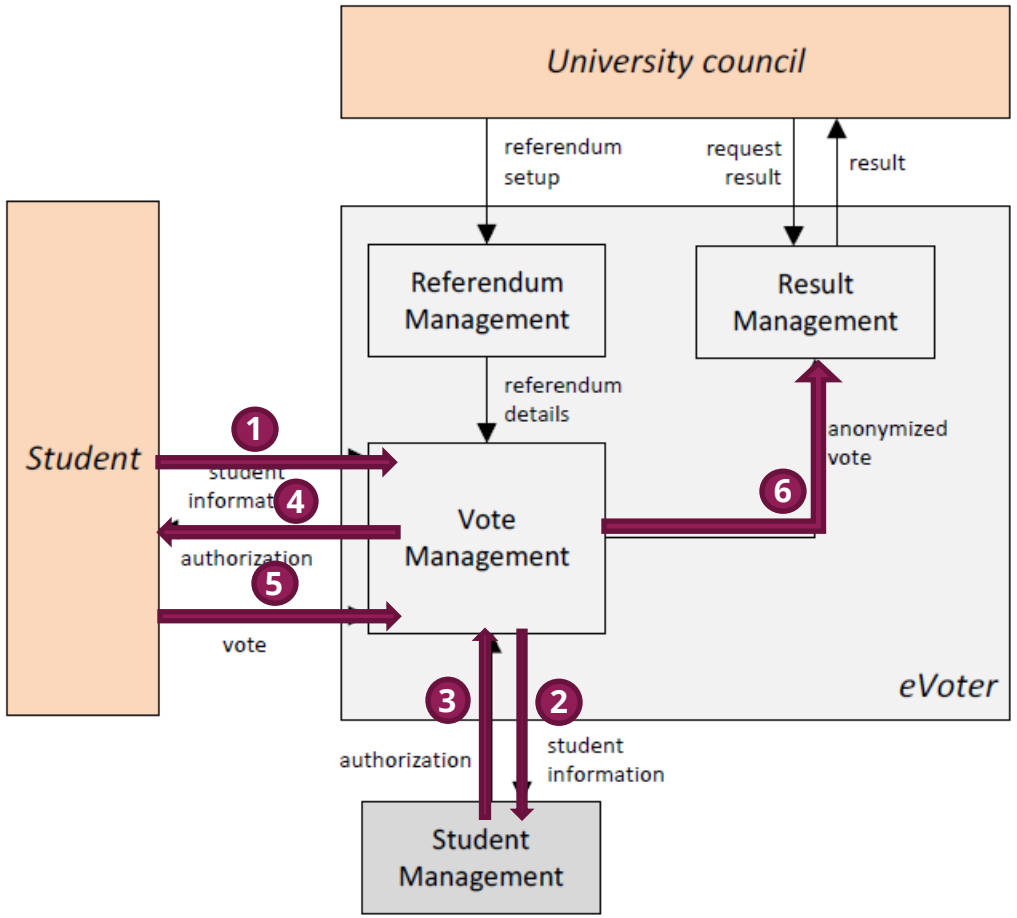
Use case	Epic / User story	ctional Architecture
Goal	Action	Module
Actor	Role	-
Main success scenario	*Acceptance test*	*Sequence of flows*
Alternative paths	-	*Sequence of flows*

- Scenario

  Sequence of flows that together describe how a goal or action is realized.

Utrecht University

User stories, Use cases and Functional Architecture

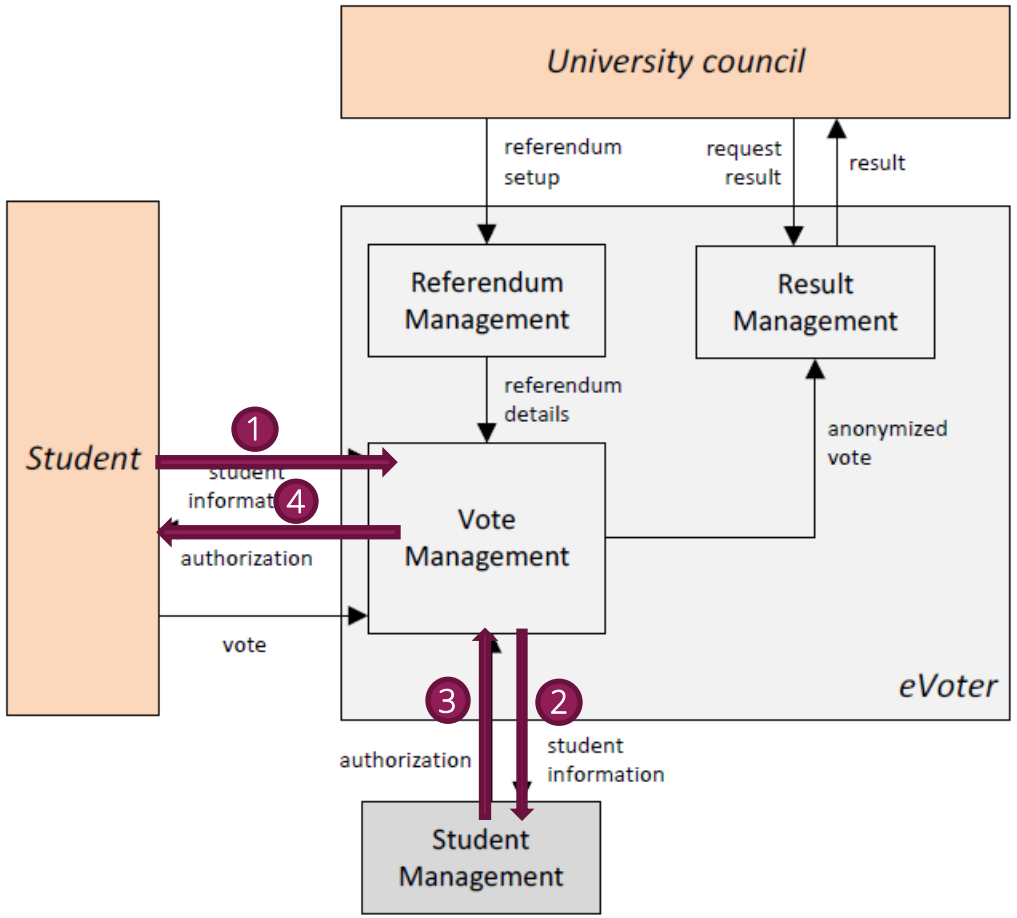


- As a participant, I want to vote anonymously

1. **Student sends student information to eVoter**
2. **eVoter sends student information to Stud. Mgmt**
3. **Stud. Mgmt sends authorization to eVoter**
4. **eVoter sends authorization details to Student**
5. **Student sends their vote to eVoter**
6. **eVoter stores anonymized vote in Result Management**

Utrecht University

User stories, Use cases and Functional Architecture



- As a participant, I want to vote anonymously

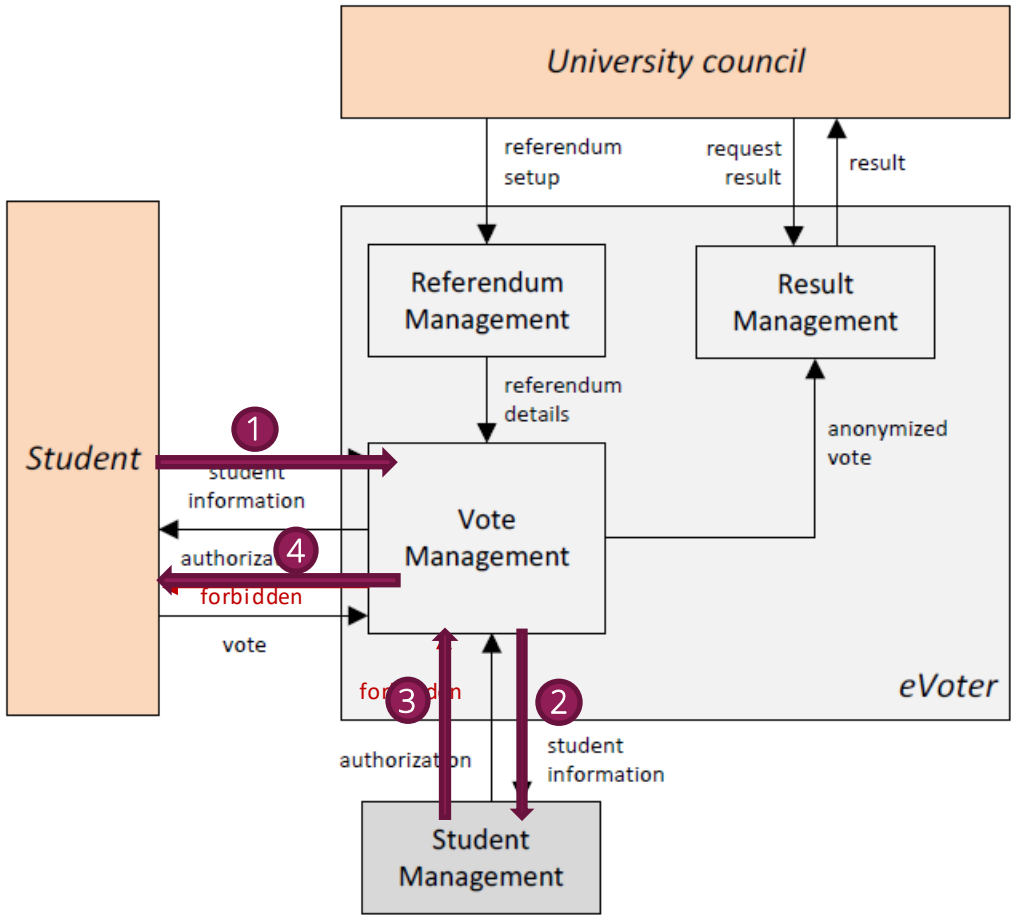
1. **Student sends student information to eVoter**
2. **eVoter sends student information to Stud. Mgmt**
3. **Stud. Mgmt sends authorization to eVoter**
4. **eVoter sends authorization details to Student**
5. **Student sends their vote to eVoter**
6. **eVoter stores anonymized vote in Result Management**

- Alternative path

1. **Student sends student information to eVoter**
2. **eVoter sends student information to Stud. Mgmt**
3. **Stud. Mgmt sends denial of the student to eVoter**
4. **eVoter denies student access**

Utrecht University

User stories, Use cases and Functional Architecture



- As a participant, I want to vote anonymously

1. **Student sends student information to eVoter**
2. **eVoter sends student information to Stud. Mgmt**
3. **Stud. Mgmt sends authorization to eVoter**
4. **eVoter sends authorization details to Student**
5. **Student sends their vote to eVoter**
6. **eVoter stores anonymized vote in Result Management**

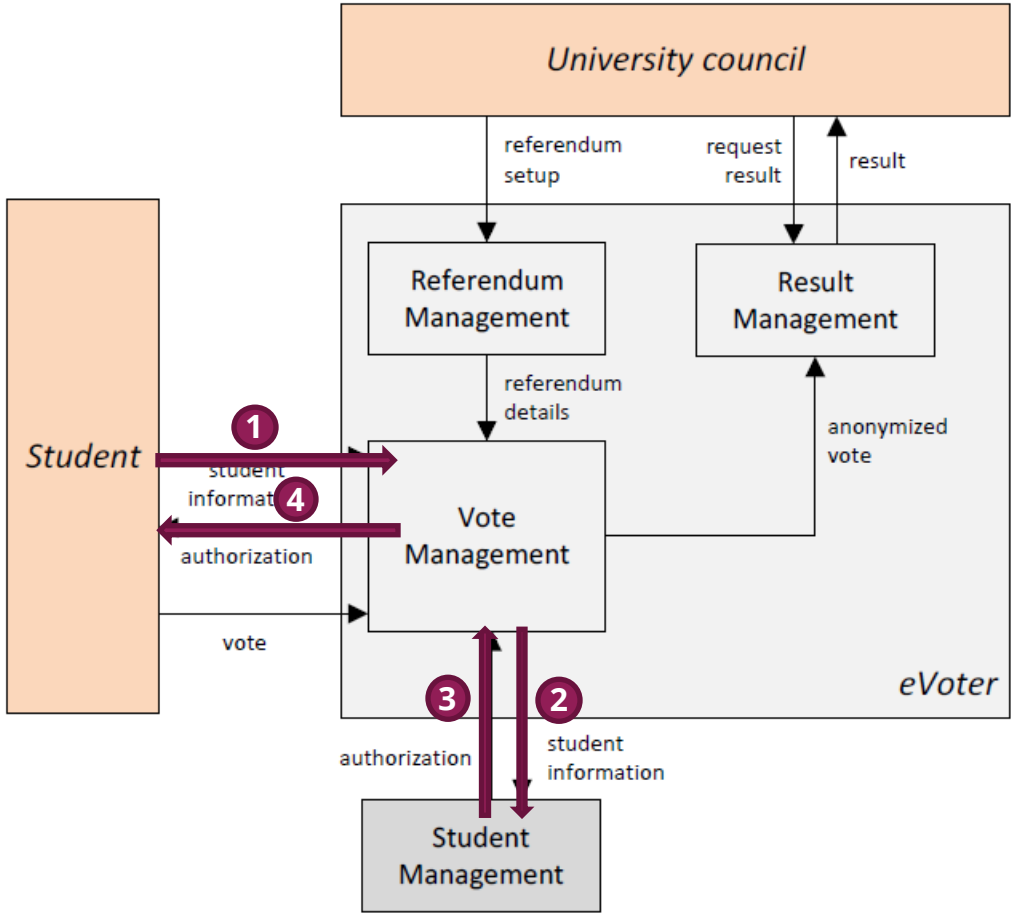
- Alternative path

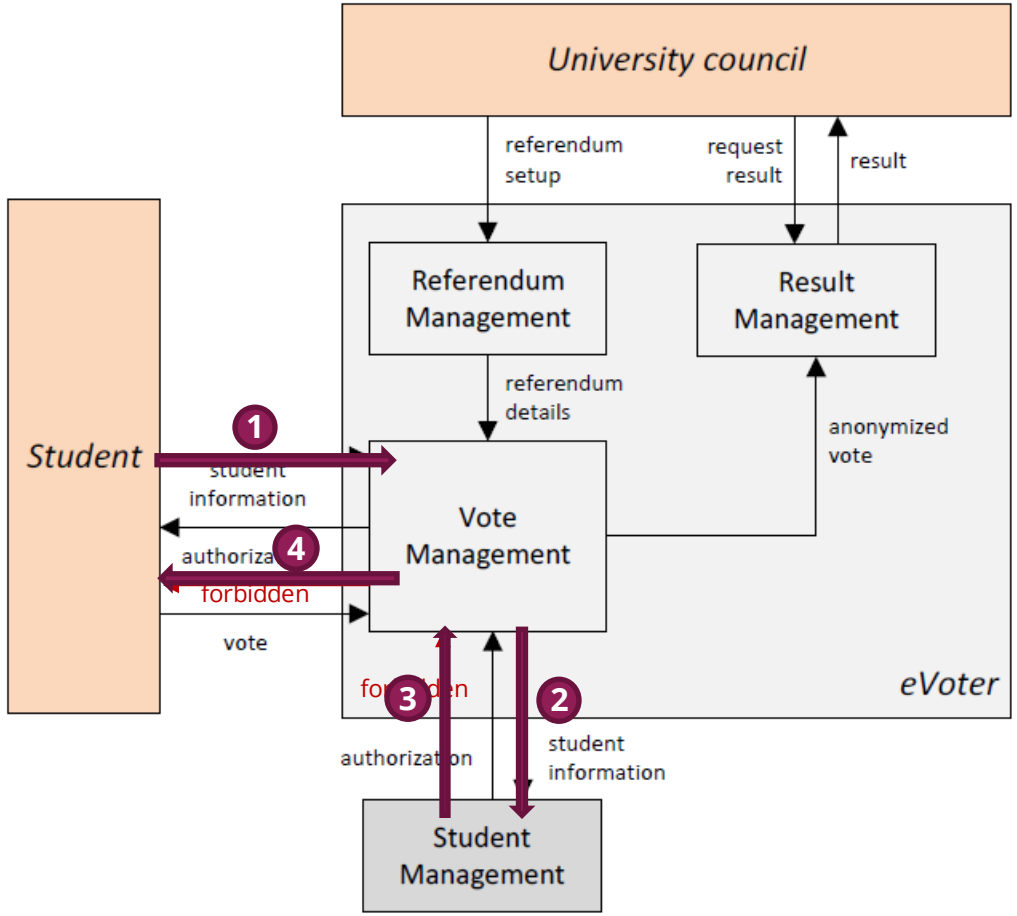
1. **Student sends student information to eVoter**
2. **eVoter sends student information to Stud. Mgmt**
3. **Stud. Mgmt sends denial of the student to eVoter**
4. **eVoter denies student access**

Utrecht University

User stories, Use cases and Functional Architecture

Which alternative do you prefer?

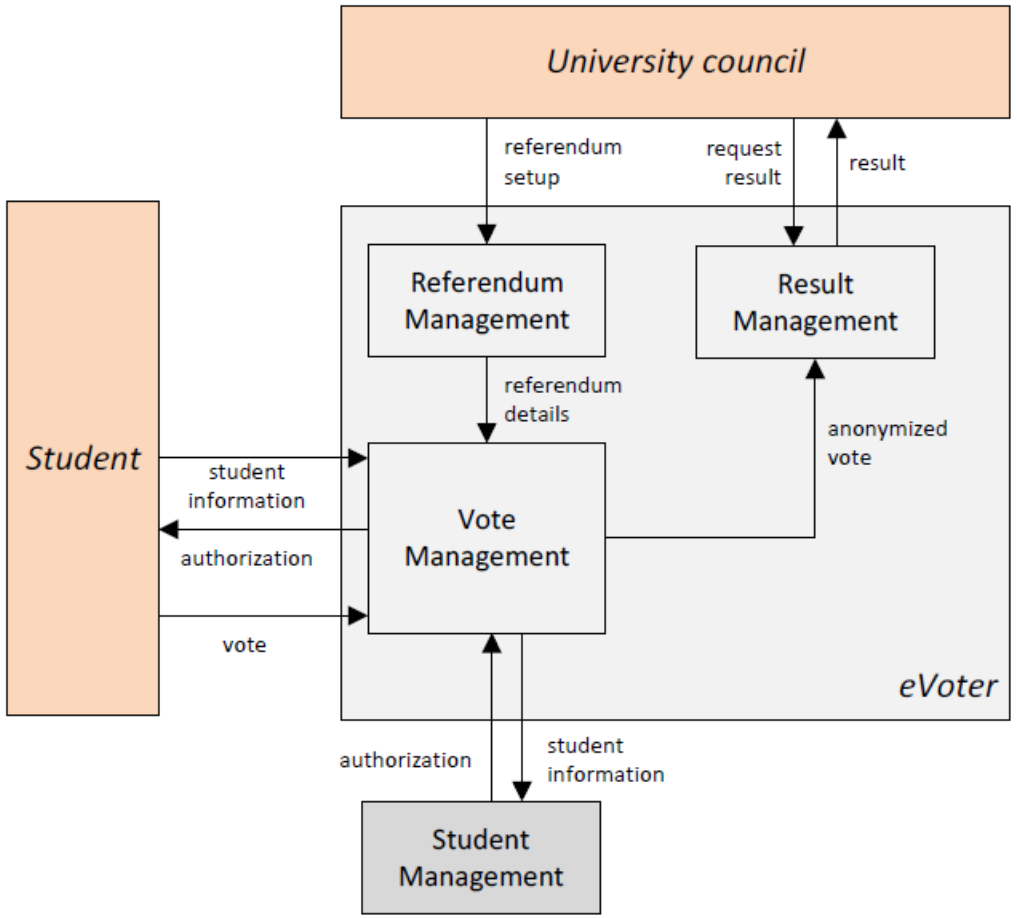




There are no clear rules: this is the creative part of architecture!

Utrecht University

Where do you start?



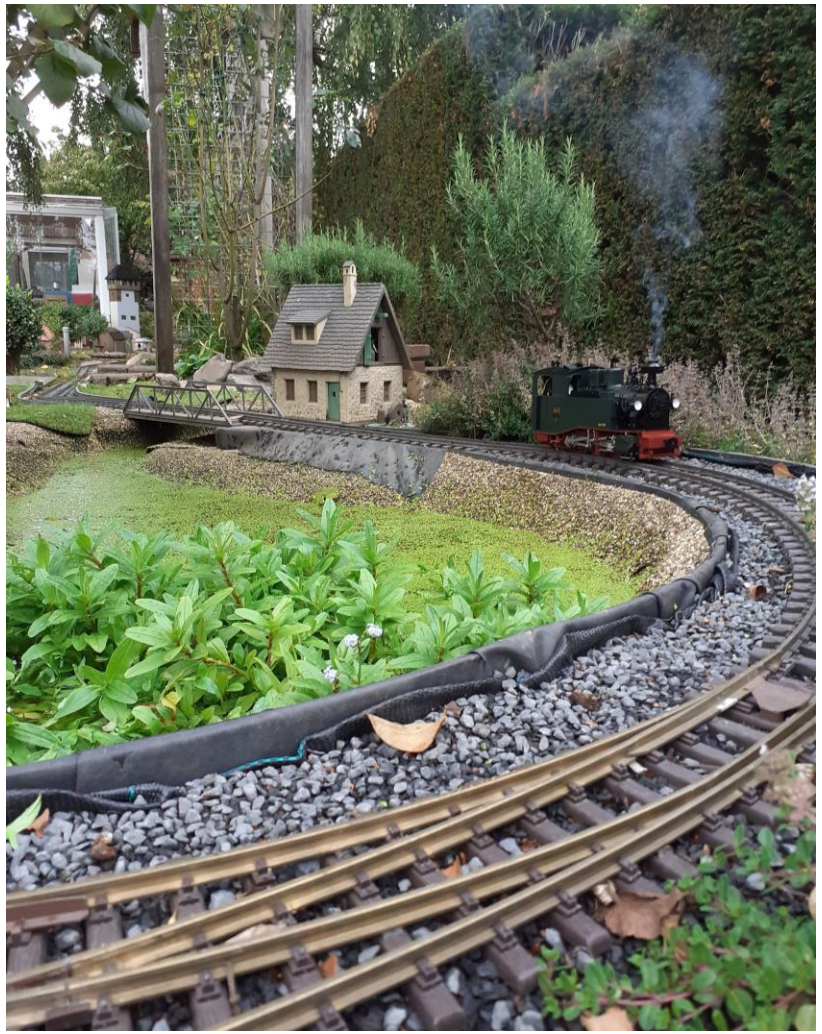
Use case	User story	Functional Architecture
Goal	Action	(sub)Module
Actor	Role	-
Main success scenario	*Acceptance test*	*Sequence of flows*
Alternative paths	-	*Sequence of flows*

- Iterative!
- Checks:

  Are the main features of the FA covered?
  Is each user story covered by a module?

Utrecht University

For now: assignment time!



- Create your functional architecture
- Checks:
  Are the main features of the FA covered by USs?
  Is each US covered by a module in the FA?
- Add more user stories if required or remove unused ones

Utrecht University

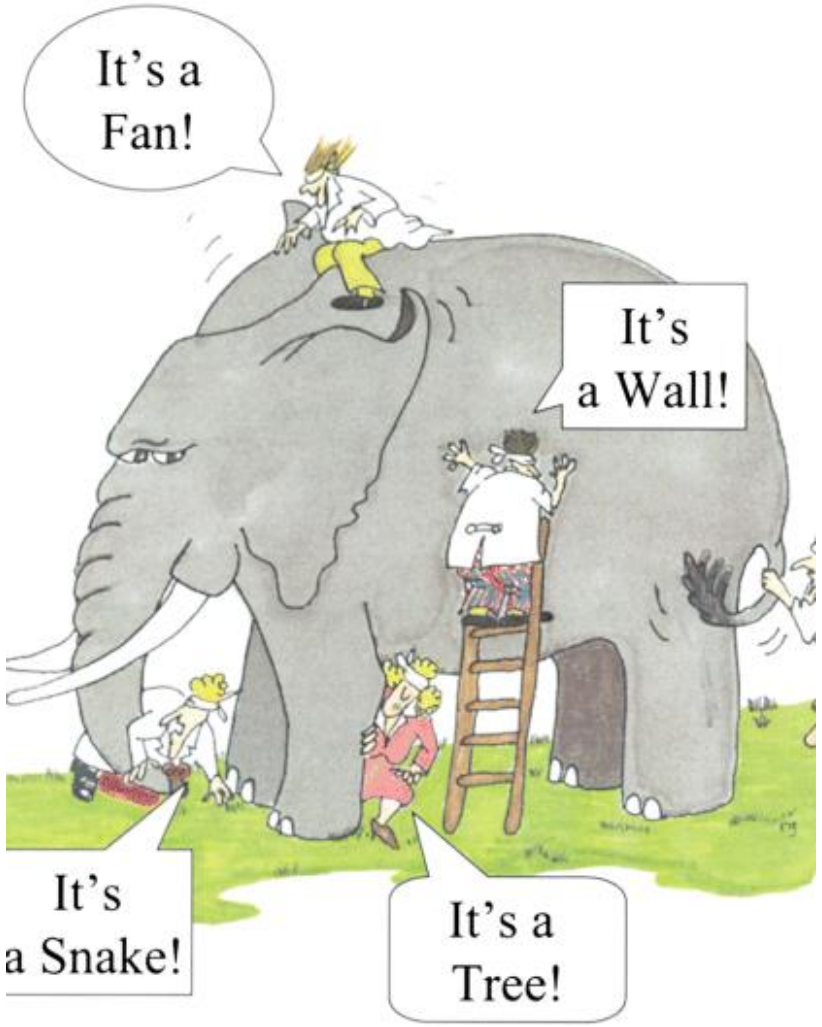
Agenda for today

- 13:15 – 13:45: Functional viewpoint
- 13:45 – 14:45: You: Functional architecture for TrIP
- 15:00 – 15:45: Quality attributes & tactics
- 15:45 – 16:45: Defining QAs in your assignment
- 16:45 – 17:00: Wrap-up

Quality Attributes: I

Utrecht University

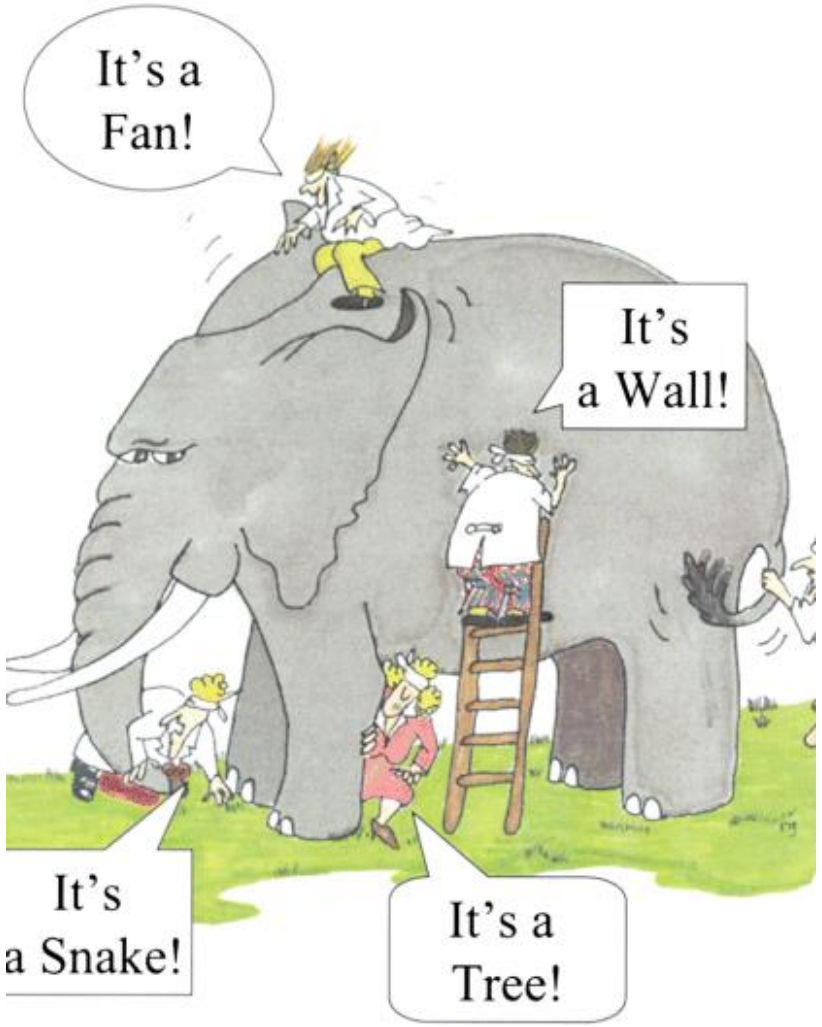
Software Architecture



- The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both. *(Bass, Clements & Kazman 2003)*
- Software architecture is the composition of a set of architectural design decisions *(Jansen & Bosch, 2005)*

Utrecht University

Software Architecture



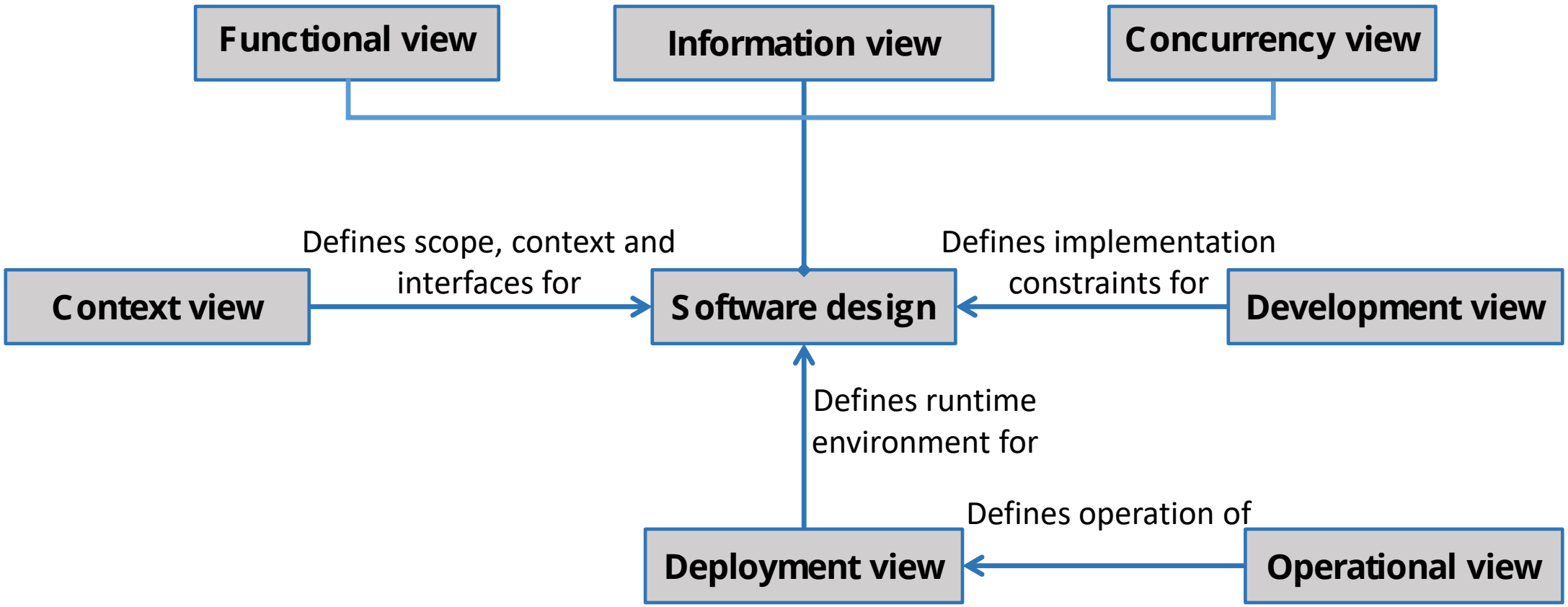
- An architecture consists of architectural elements and relations between them
- Views are used to depict these relations and to document the architecture
- Views address concerns of the stakeholder:

  How do you show a concern of a stakeholder is addressed?
  By one or more views!
  When do you add a view to the documentation?
  When it addresses a concern of a stakeholder!

- Many different views exist. Therefore, we order them in viewpoints. Each viewpoint addresses a specific set of concerns. We have 7 of them.

Utrecht University

Viewpoint catalog

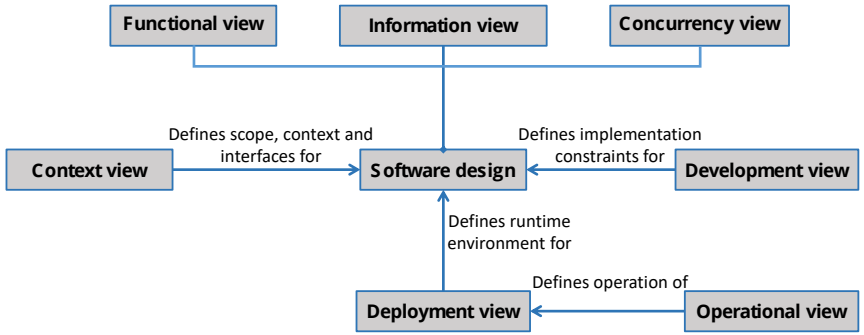


Viewpoint:

Collection of patterns, templates and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views

Utrecht University

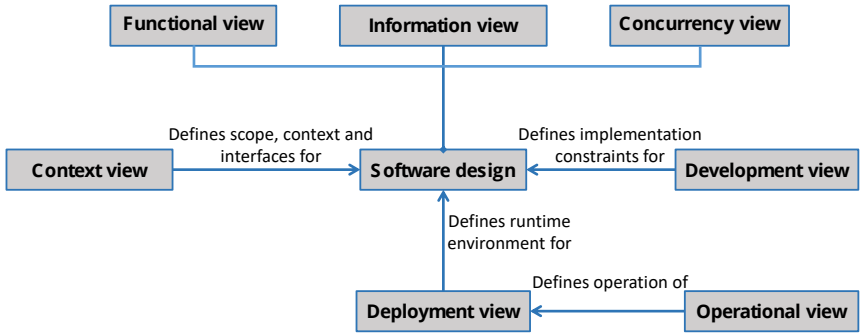
Information view



- Information view:
  - **Describes the way the system stores, manipulates, manages and distributes information**
- Concerns
  - **Information structure and content**
  - **Information purpose and usage**
  - **Information ownership**
  - **Volatility of information, timeliness, latency**
  - **Information flow, consistency and quality**
  - **Age, archiving, retention**
- Models and views
  - **Structure models (ERD, …), flow models (BPMN, Petri nets, …)**
  - **Information life cycle models, ownership models, …**
- Problems and pitfalls
  - **Key-matching deficiencies, interface complexity,**
  - **Overloaded central DB, inconsistent distributed DB**
  - **Multiple concurrent updaters …**

Utrecht University

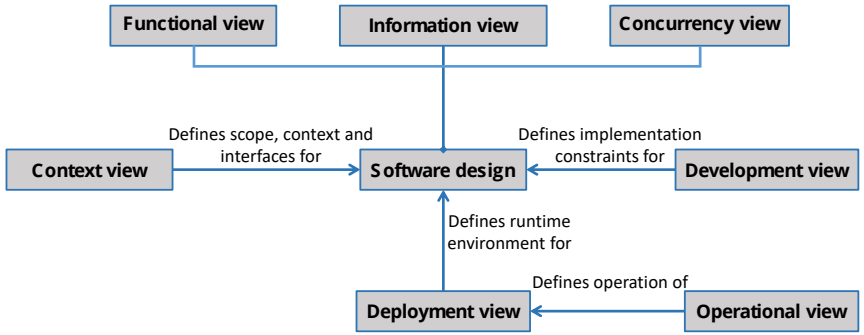
Deployment view



- Deployment view:
  - **Describes the environment into which the system will be deployed, and the dependencies that the system has on elements of it**
- Concerns
  - **Runtime platform,**
  - **Specification of hardware or hosting, network requirements**
  - **Physical constraints**
- Models and views
  - **Runtime platform models,**
  - **Network models**
  - **Technology-dependent models**
- Problems and pitfalls
  - **Unproven technology,**
  - **Unsuitable / missing Service Level agreements**
  - **Ignoring inter-site complexities**
  - **Disaster recovery environment**

Utrecht University

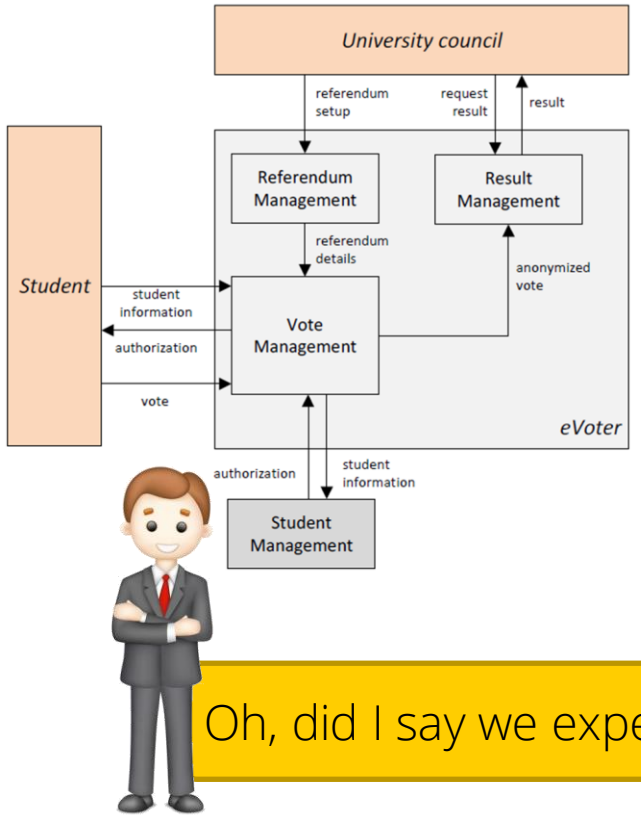
Operational view



- Operational view:
  - **Describes how the system will be operated, administered, and supported when running in its production environment**
- Concerns
  - **Installation and upgrade, functional & data migrations**
  - **Operational monitoring and control, alerting,**
  - **Configuration management, backup & restore**
- Models and views
  - **Installation models, migration models,**
  - **Configuration models, …**

Utrecht University

Remember our Electronic voting system?

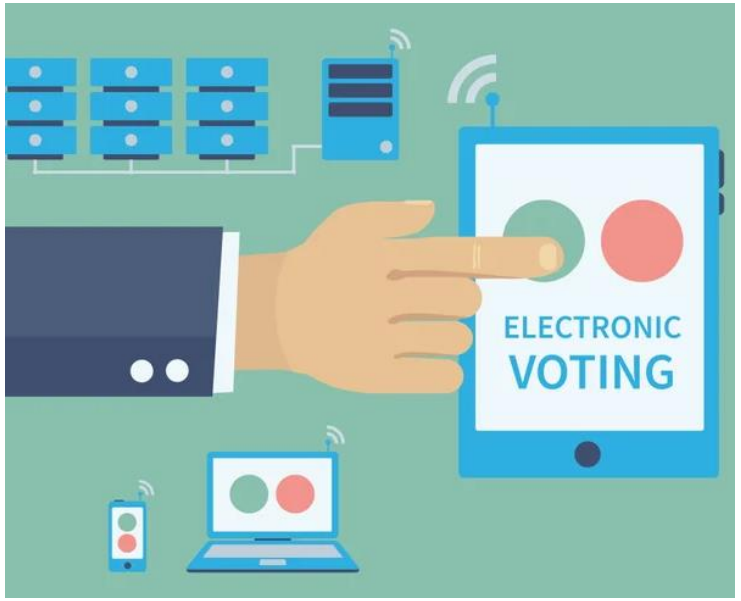


User stories:

- As the organiser, we want to hold referenda, so that we know the opinion of our participant
- As a participant I want to vote anonymously, so that my privacy is respected

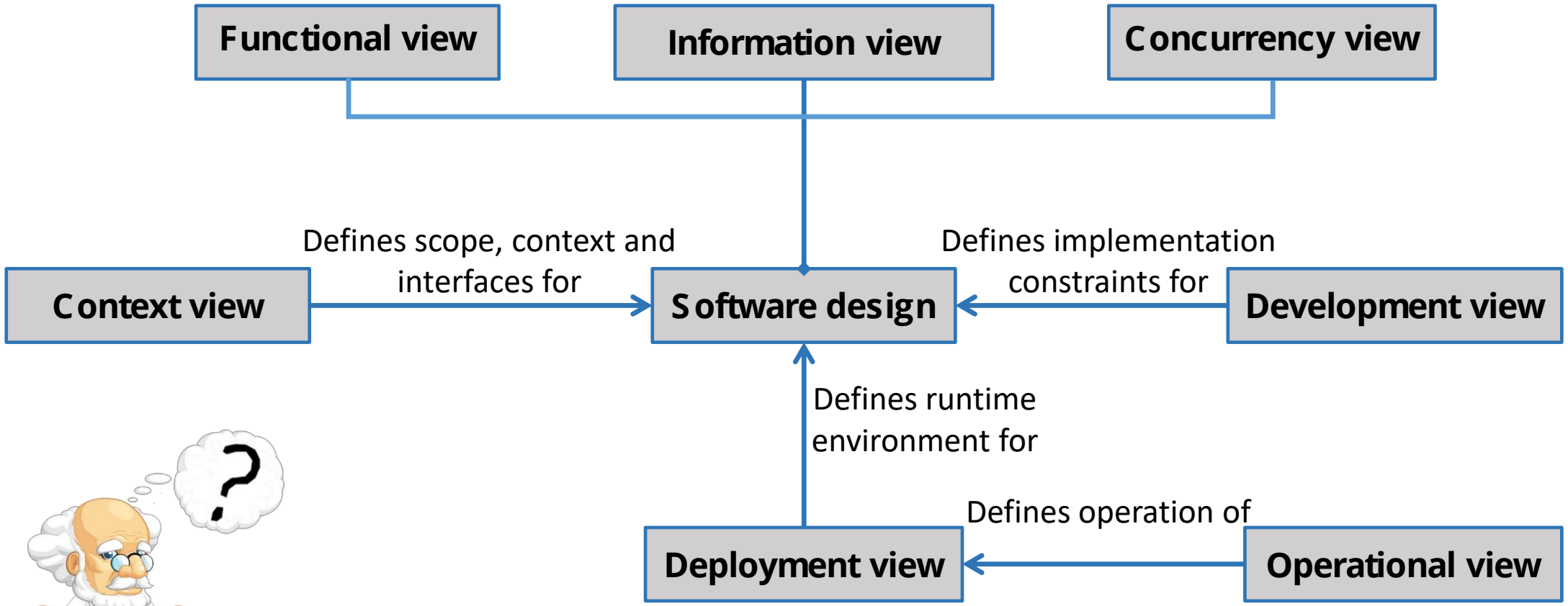
Oh, did I say we expect 150M voters in 12 hours?

Can your system handle this? How do you check this?



Utrecht University

Viewpoints as an aid to design the system



How to address properties of the system?

Utrecht University

How do you know your software architecture is “correct”?

Utrecht University

How do you know your software architecture ~~is “correct”~~?

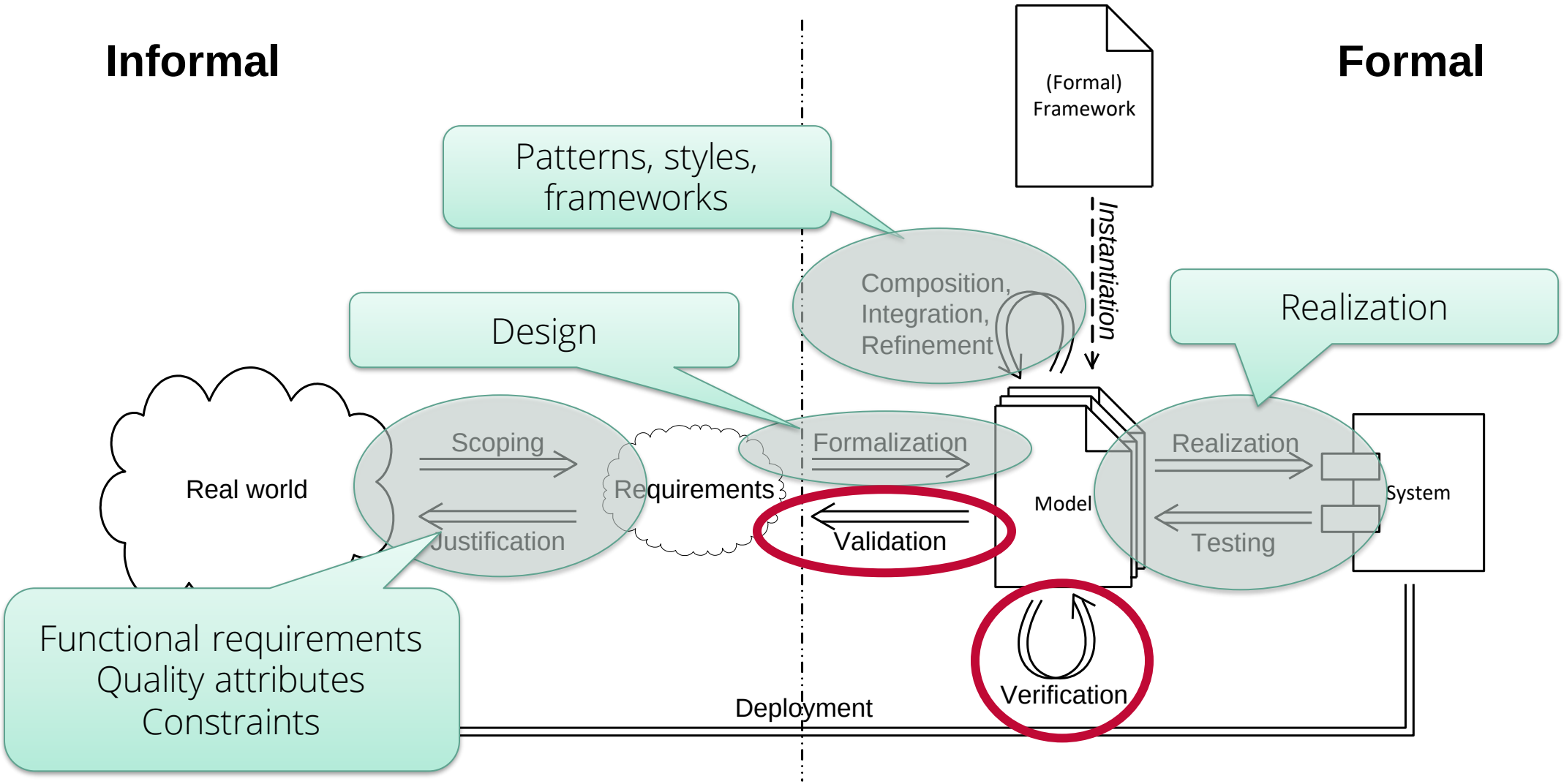
Utrecht University

How do you know your software architecture

Sufficiently *satisfies all requirements?*

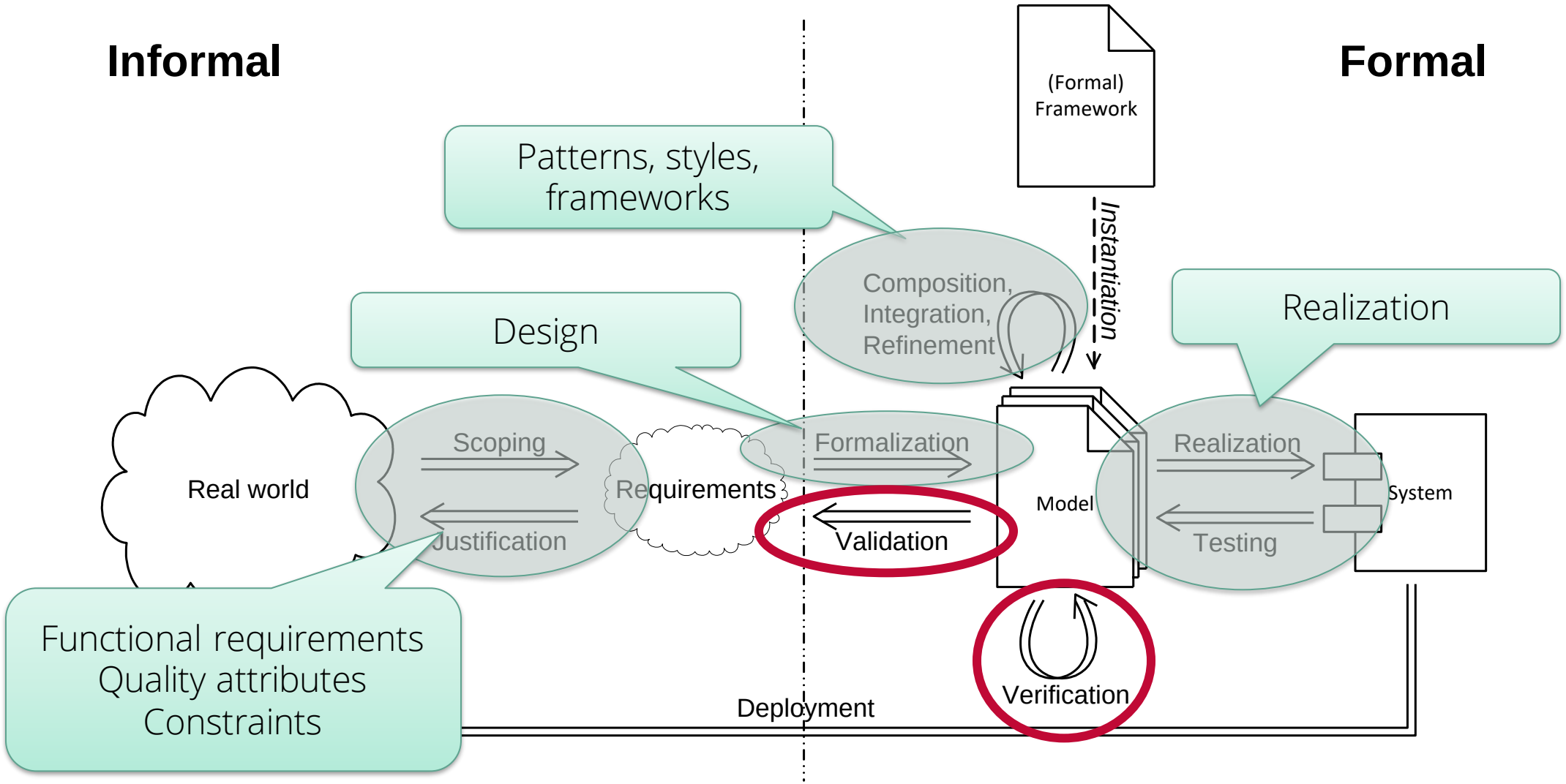
Utrecht University

Software Architecture: a short summary



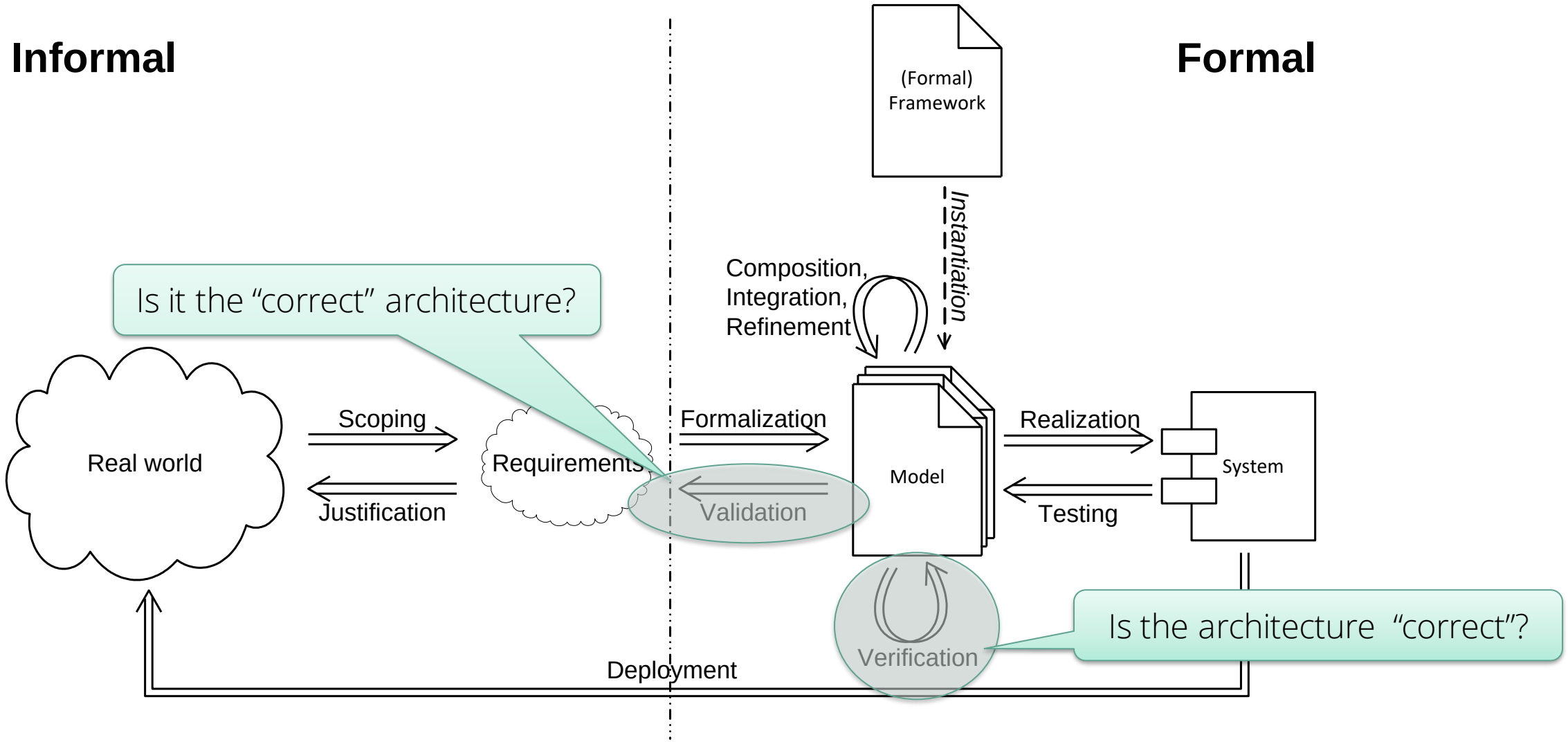
Utrecht University

So, we have an architecture, how “correct” is it?



Utrecht University

So, we have an architecture, how “correct” is it?



Utrecht University

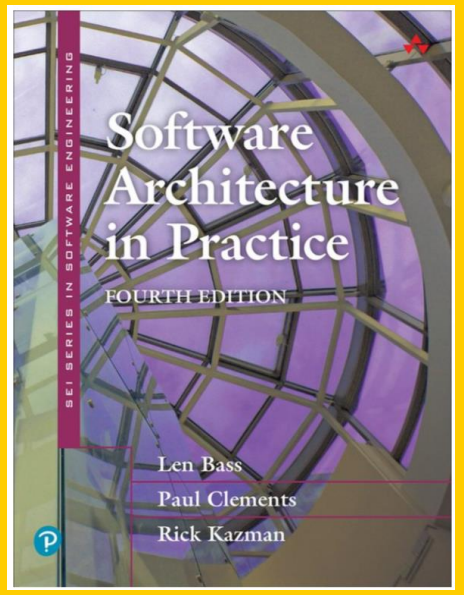
Basic assumption in modeling: Property P holds on model? Then also on the system

- Verification:
  Check whether the model is "correct"
- Validation
  Check whether it models the correct thing

	System has property	System does not have property
Model has property	Correct	False positive
Model does not have property	False negative	Correct

Utrecht University

Quality attributes & perspectives



A combination of two books…

Utrecht University

Software Architecture & requirements

System requirements come in three flavors:

- Functional requirements:
  what the system ***must do***: how it must behave or react to run-time stimuli
- Quality attributes:
  Annotate (qualify) functional requirements.
- Constraints:
  Design decisions with zero-degrees of freedom

Utrecht University

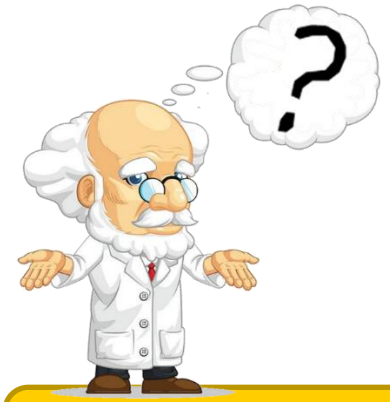
An example!

- When the user presses the green button, the Options dialog should appear

A performance QA: the dialog appears within 500ms

An availability QA: It may only fail in 1 out of 1000 times.

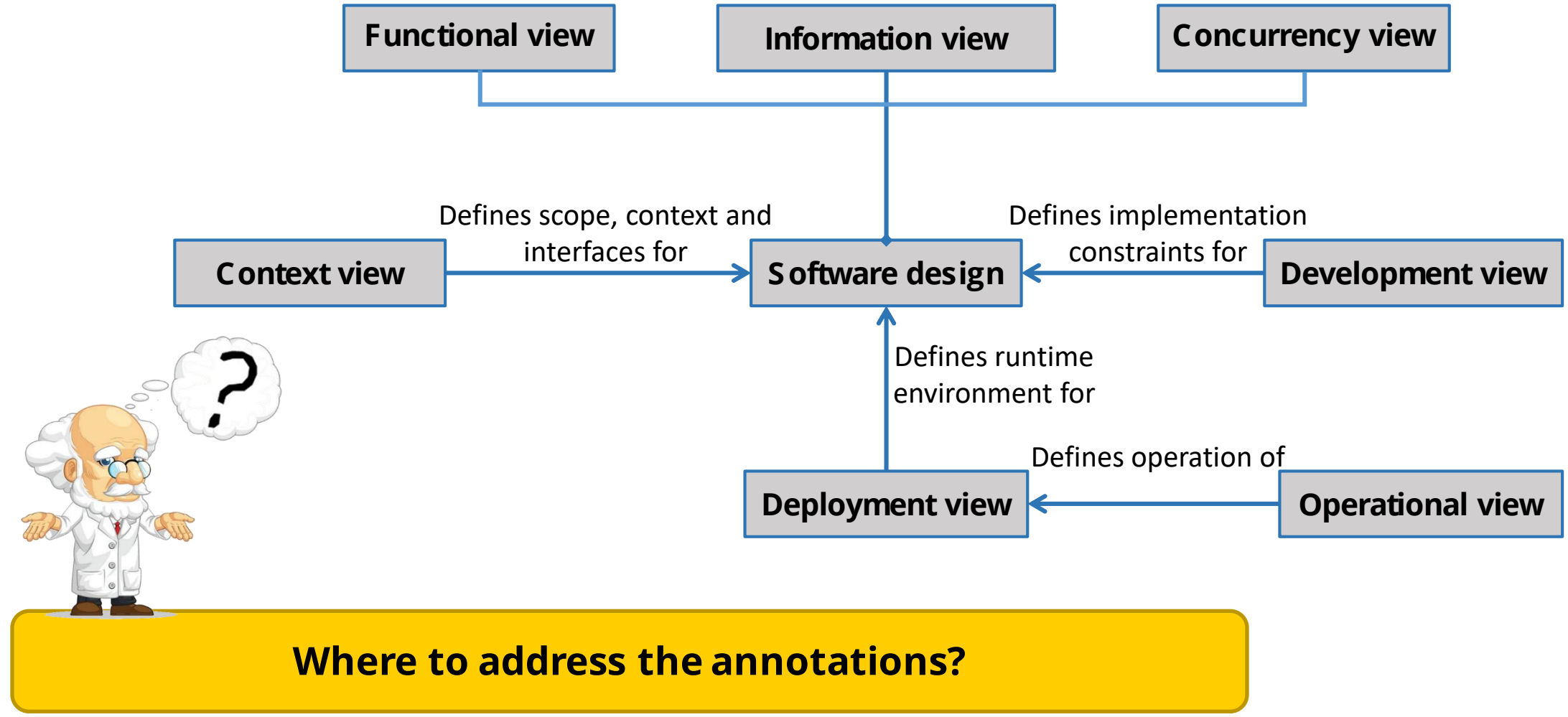
A Usability QA: The green button should be easy to be found



How are these addressed in requirements engineering?

Utrecht University

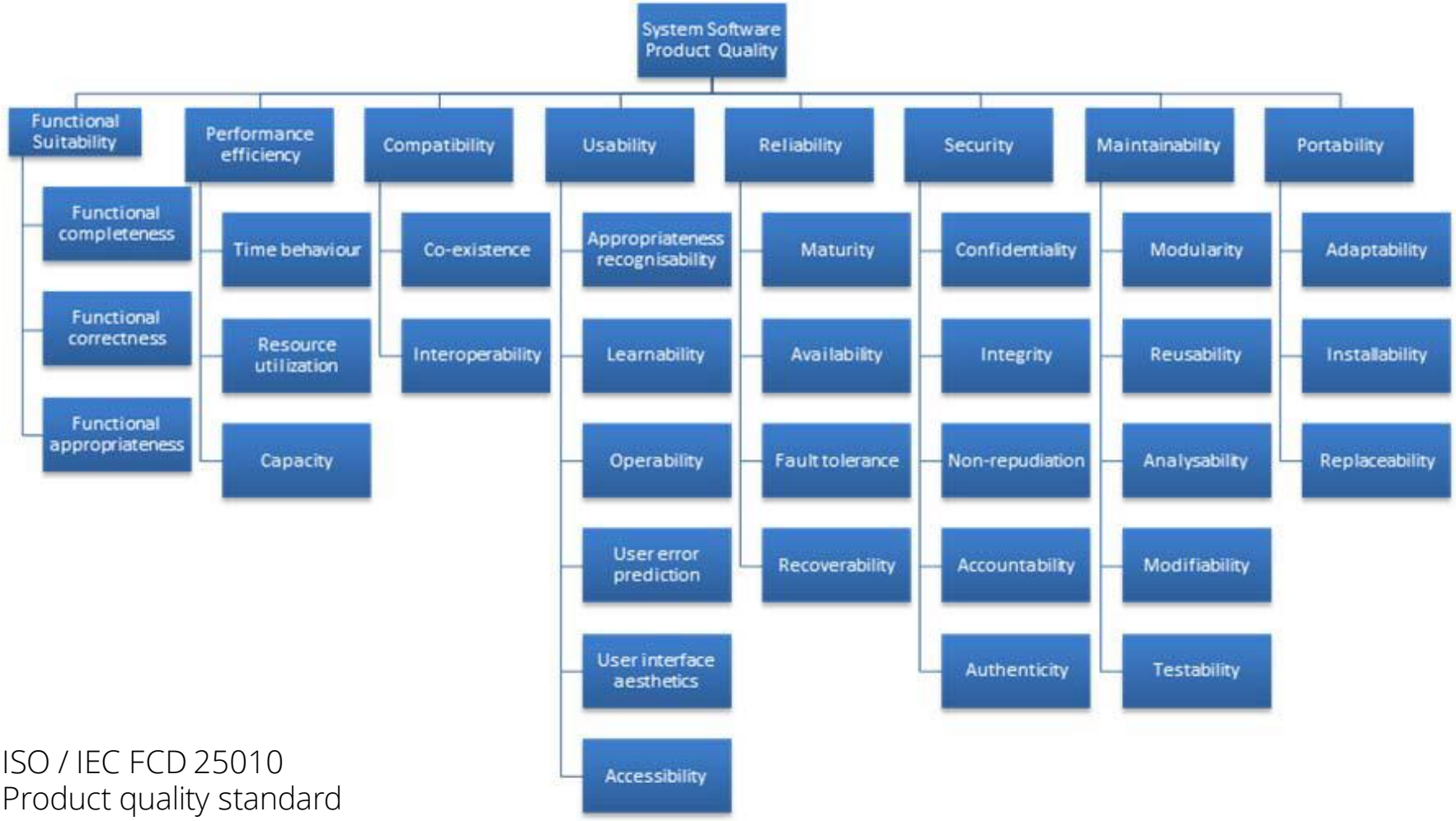
Quality attributes and viewpoints



Where to address the annotations?

Utrecht University

Many different qualities to look into...



ISO / IEC FCD 25010
Product quality standard

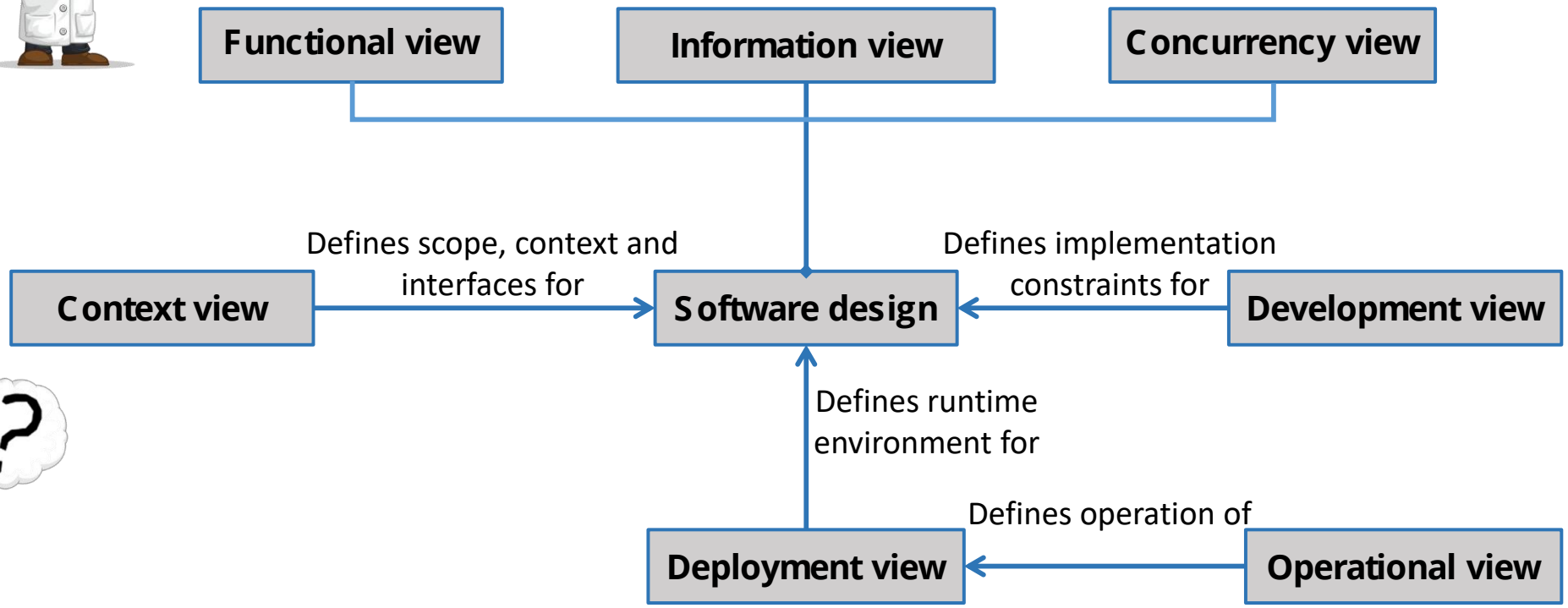
Utrecht University

Quality attributes and viewpoints

Idea: describe for each view how it is influenced

What are the concerns of stakeholders?

How can you address these concerns?



Utrecht University

Viewpoints and Quality Attributes

- Put the important QAs on the columns
- For each cell:

  What is the importance of the QA on the view?
  How influence the view and QA each other?

	Availability	Performance	…
Context			
Functional			
Information			
Concurrency			
Development			
Deployment			
Operational			

Utrecht University

Viewpoints and Quality Attributes

- Put the important QAs on the columns
- For each cell:

  What is the importance of the QA on the view?
  How influence the view and QA each other?

	Availability	Performance	…
Context			
Functional			
Information			
Concurrency			
Development			
Deployment			
Operational			

Next week, Monday : Quality Attributes: II: How to analyse this systematically?

Utrecht University

Agenda for today

- 13:15 – 13:45: Functional viewpoint
- 13:45 – 14:45: You: Functional architecture for TrIP
- 15:00 – 15:45: Quality attributes & tactics
- 15:45 – 16:45: Defining QAs in your assignment
- 16:45 – 17:00: Wrap-up

Utrecht University

For Next time

- Next Lecture
  - **Patterns and styles**
    - **➔ During lecture we are going to apply patterns**
  - **Architectural decisions**
- Study the following paper:
  - **A. Jansen, J. Bosch (2005). Software Architecture as a Set of Architectural Design Decisions.**
- Questions:
  - **What problems do the authors observe?**
  - **What solutions do they propose?**

Utrecht University

DISCLAIMER

The information in this presentation has been compiled with the utmost care, but no rights can be derived from its contents.