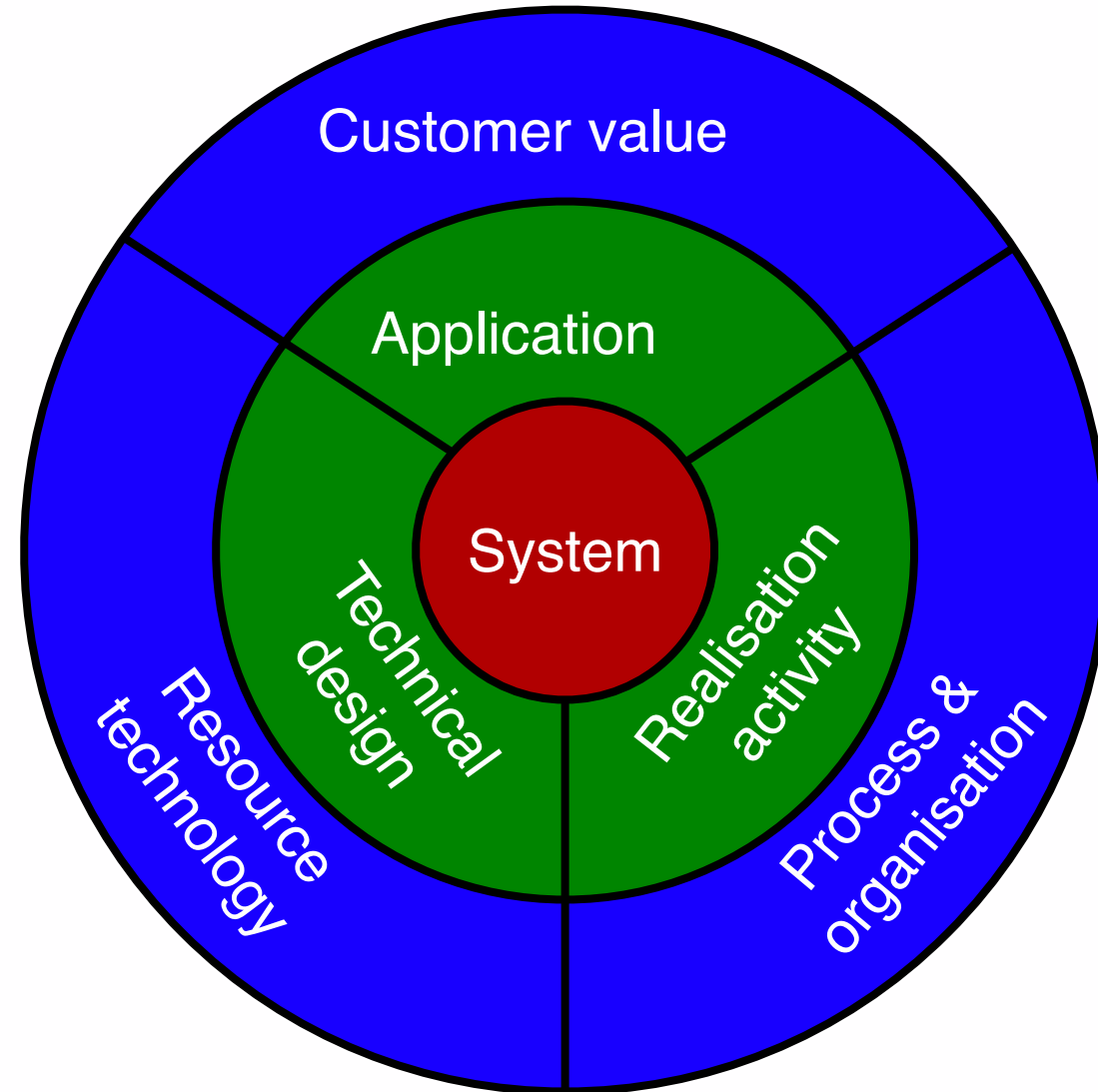


Lecture 2

Design of Software Architectures

Lecture 1 Recap: Software architecture

- The **software architecture of a system** is
 - a collection of **statements**
 - that gives direction to
 - the **application**
 - the **design**
 - the **realisation**
 - of the software in its **environment**
- **Statements** are a model of:
 - **structure** or **guidelines**



Lecture 1 Recap: Architecture Description (AD)

Suggested Structure

- Introduction
- System scope
- Environment
 - Stakeholders and their concerns
 - Related systems
 - Related processes, organisations, stakeholders
 - Main QA scenarios and use cases
 - Trends and developments
- Design
 - Dominant decomposition/style
 - Design decisions: Justification of choices related to environment
 - View 1..N
 - Per view: Used design patterns (show examples; if you cannot, generalise)
 - Consistency: relation between views, known gaps/inconsistencies.
- Appendix
 - Explanation of viewpoints, references to standards

Lecture 2: Overview

- A Good Architecture is...
- Stakeholders/Concerns/Views.

Good AD...

- is clear & concise (diagrams, charts, visual aids, etc.)
- states purpose and goals
- describes major components
 - their interactions, flow of data, control
- mentions architectural styles, patterns, paradigms
- describes interfaces, APIs
- describes data storage and management
- discusses deployment and scalability
- addresses security and privacy
- addresses performance considerations, error handling and resilience
- describes dependencies and third-party integrations
- is **correct, consistent** and **communicated**, etc.

Good architecture is correct

- The architecture is based on
 - **validated statements** about the environment
 - (the stakeholder concerns in particular)
- **Concerns are prioritised**
- Architecture **balances** the concerns
- Achieved by:
 - Adequate environment analysis
 - Conscious balance of interests
 - Validation of environment statements

-> “The system fits its environment”

Good architecture is consistent

- The architecture forms a whole
- Architecture **statements do not conflict**
- The system can become reality with the architecture
- Achieved by:
 - Verification on contradictions
 - Demonstrable PoCs
 - Conscious management

-> “The system is well engineered”

Good architecture is communicated

- Stakeholders
 - **know** their relation with the architecture
 - **know** what to do with it
 - **understand** how their concerns are (not) covered
- Starts with a communicable architecture view/description
- Achieved by:
 - Actionability by stakeholders
 - Sufficient anchoring
 - Conscious performance

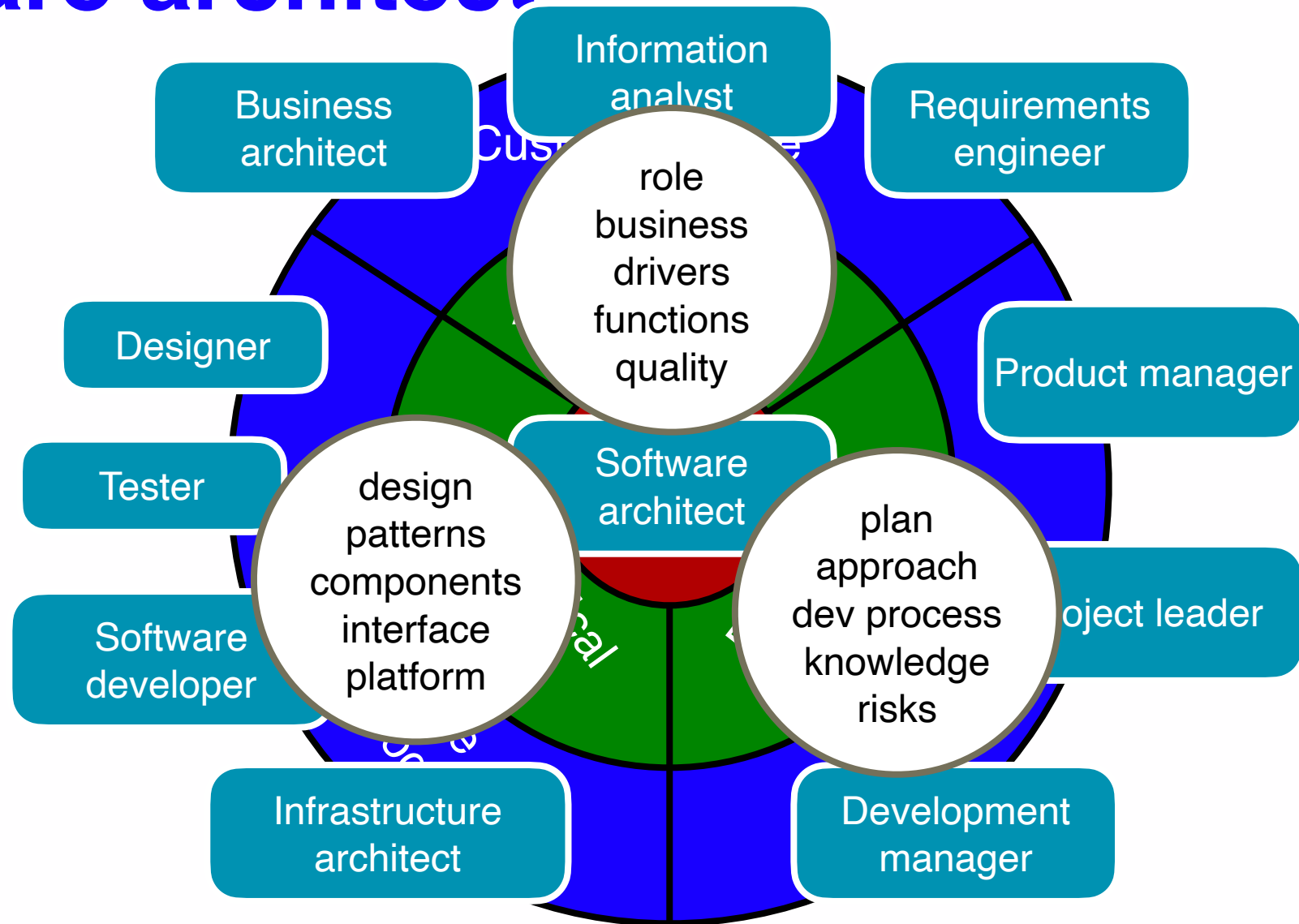
-> “Everyone knows what they should know”

General Advice

- **Consistency** is key
- Remember relations from ISO 42010
 - stakeholder —has-> concern —is-about-> quality attribute
 - stakeholder —plays-role-in-> process —involves-> view
 - system —interacts-with-> environment element
- **Quality** >> **functionality**
- Intermediate documents can go into the appendix
- Use the lecturer as a **coach** or a **stakeholder**



Software architect



Make it good

Good architecture is correct

- The architecture is based on
 - validated statements about the environment
 - (the stakeholder concerns in particular)
- Concerns are prioritised
- Architecture balances the concerns
- Achieved by:
 - Adequate environment analysis
 - Conscious balance of interests
 - Validation of environment statements

→ "The system fits its environment"



Good architecture is consistent

- The architecture forms a whole
- Architecture statements do not conflict
- The system can become reality with the architecture
- Achieved by:
 - Verification on contradictions
 - Demonstrable PoCs
 - Conscious management

→ "The system is well engineered"



Good architecture is communicated

- Stakeholders know their relation with the architecture
- Stakeholders know what to do with it
- Stakeholders understand how their concerns are (not) covered
- A communicated architecture starts with a communicable architecture view/description.
- Achieved by:
 - Actionability by stakeholders
 - Sufficient anchoring
 - Conscious performance

→ "Everyone knows what they should know"



- Concretise the abstract advice
- Formulate guidelines for your project
 - correctness
 - consistency
 - communication

Lecture 2: Overview

- A Good Architecture is...
- Stakeholders/Concerns/Views.

Role of the software architecture



to provide
solution direction
for
most important properties
that are
the most difficult to realise



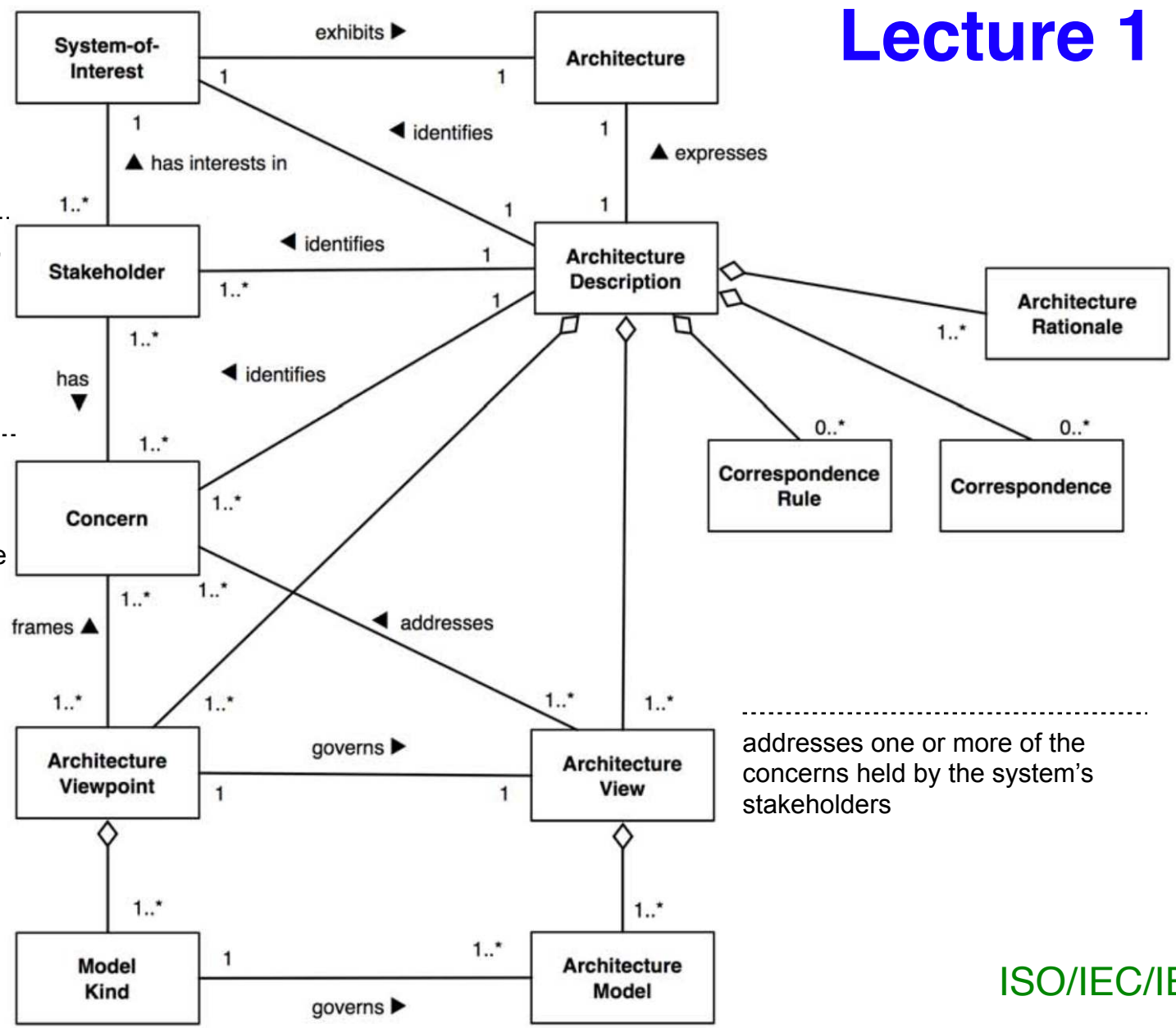
Lecture 1 Recap: AD

individual, team, organization, or classes thereof, having an interest in a system

interest in a system relevant to one or more of its stakeholders (functionality, features, limitations, resource utilization, reliability, inter-process communication)

the viewpoint establishes the conventions for constructing, interpreting and analyzing the view

addresses one or more of the concerns held by the system's stakeholders

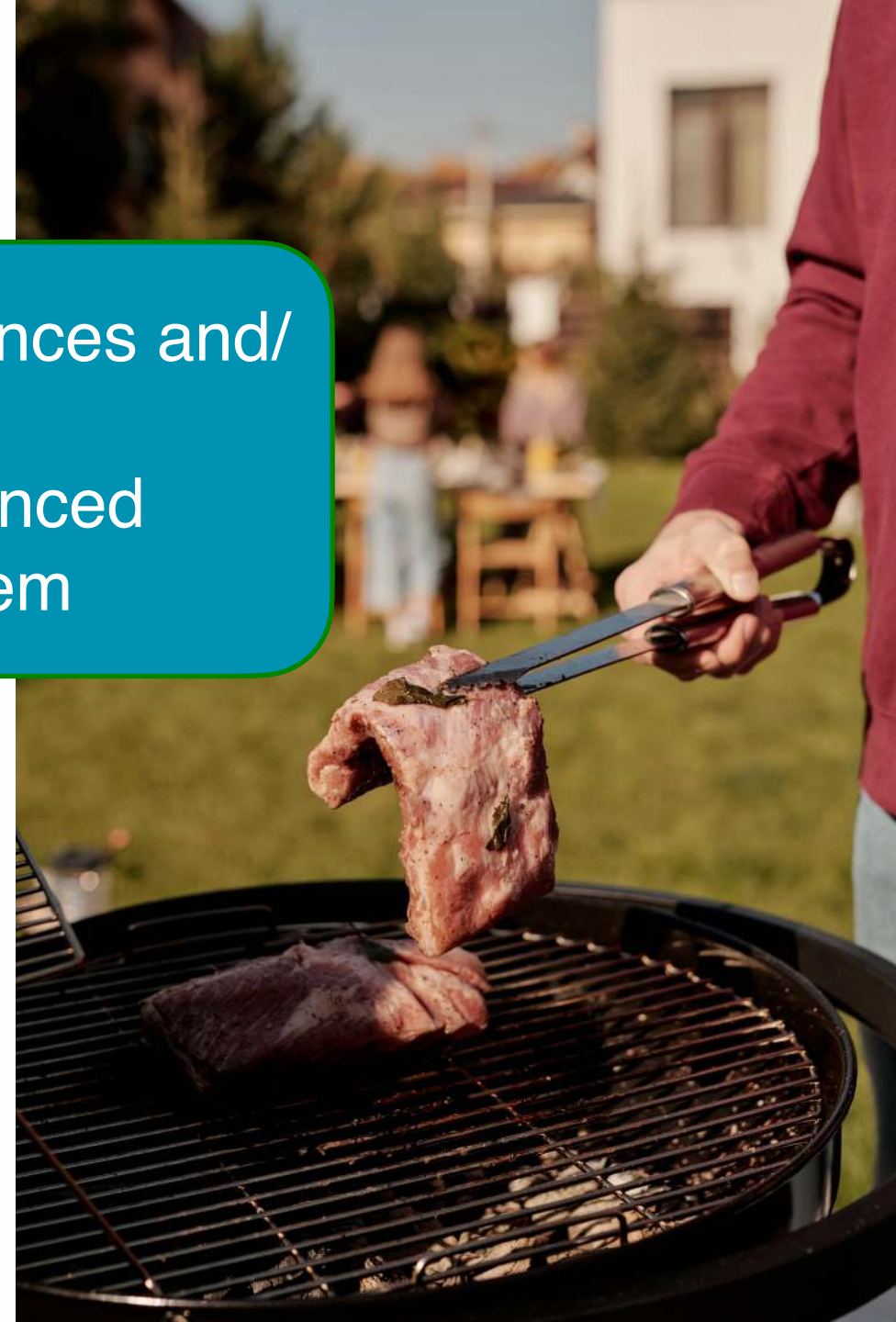


ISO/IEC/IEEE 42010:2011(E), §4.2.2

Relevant stakeholders

- **users** of the system
- **operators** of the system
- **acquirers** of the system
- **owners** of the system
- **suppliers** of the system
- **developers** of the system
- **builders** of the system
- **maintainers** of the system

Anyone who influences and/
or
is being influenced
by the system



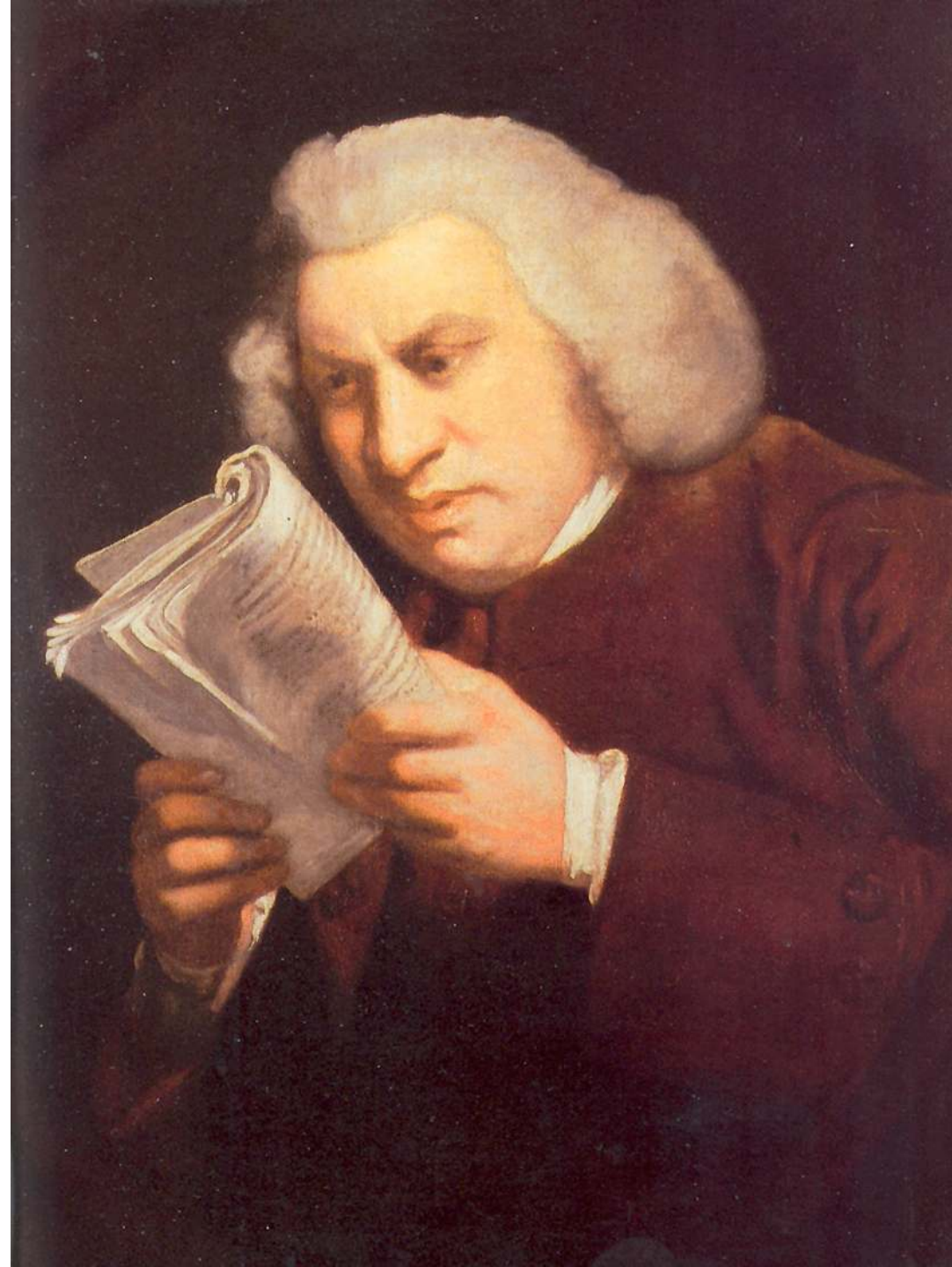
Relevant concerns

- **purposes** of the system
- **suitability of the architecture** for achieving the system's purposes
- **feasibility** of constructing and deploying the system
- potential **risks** and **impacts** of the system to its stakeholders throughout its life cycle
- **maintainability** and **evolvability** of the system



View

- **Architecture View** is
 - a work product
 - expressing the architecture
 - from the perspective of specific concerns
- Views are **not limited to expressing the system**
 - environment
 - relation between the system and environment



Stakeholders/Concerns/Views?



- owners (S)
- safety (C)
- harm (C)
- safety features and mechanisms (V)
- manufacturers (S)
- fuel providers (S)
- reliability (C)
- quality control measures (V)
- fuel (C)
- insurance companies (S)
- government regulators (S)
- emissions (C)
- emission control systems (V)
- cost breakdowns (V)
- impact on environmental sustainability (V)

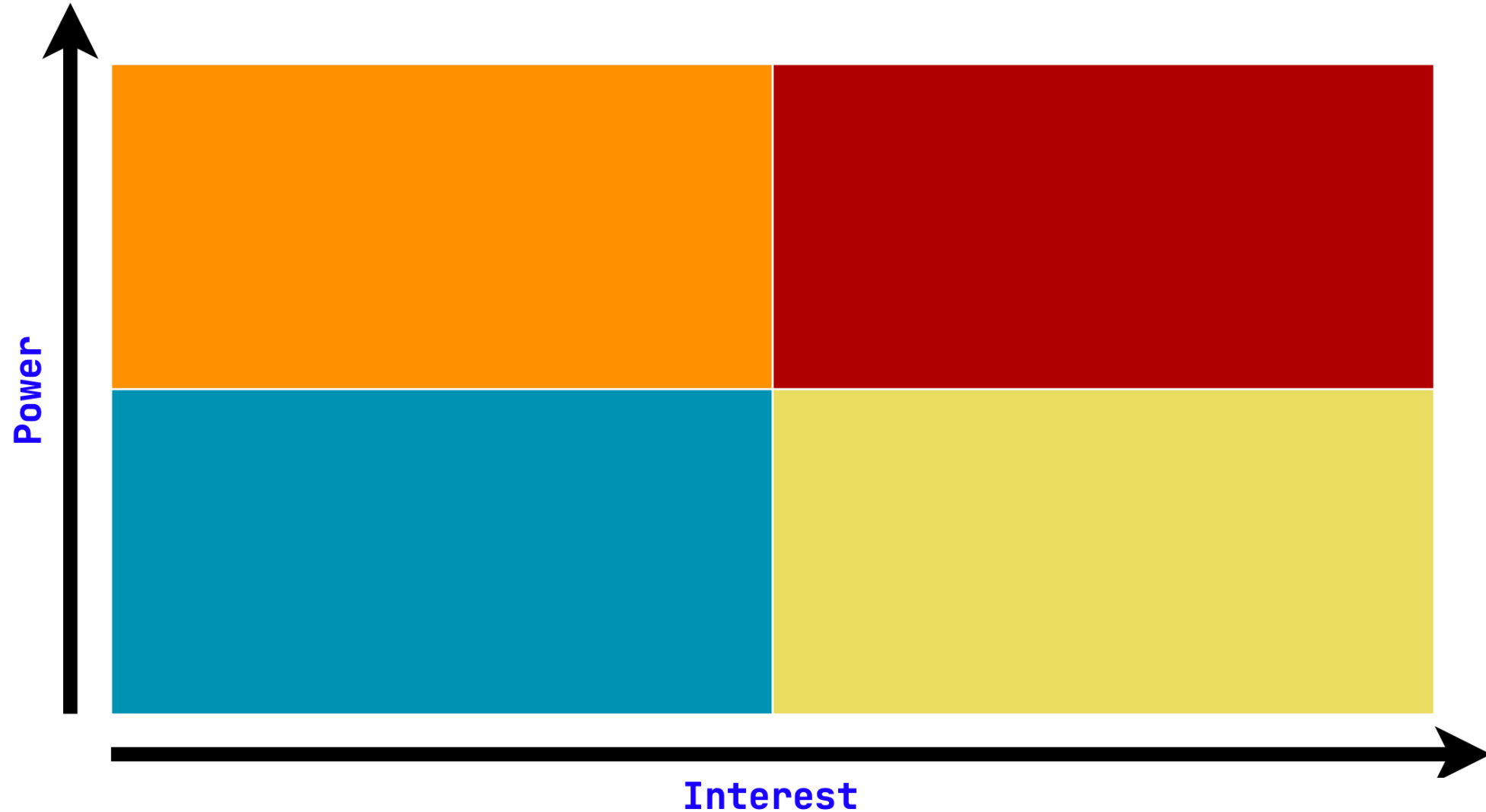
What do you do with stakeholders?



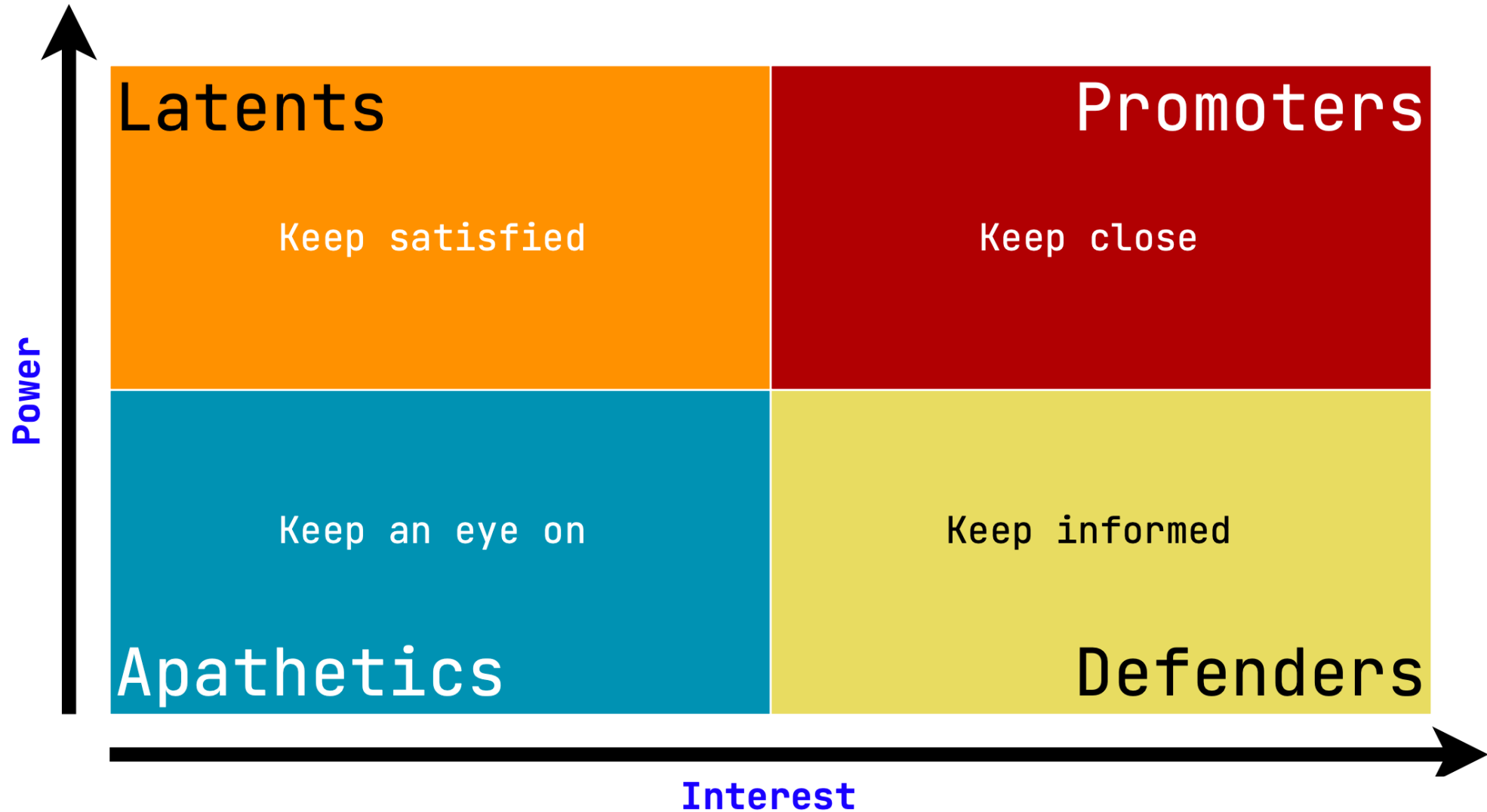
How to find stakeholders?

- **Direct** actions with the system
 - develop, buy, use, ...
- **Indirect** actions
 - finance the development, motivate buyers, train users, ...
- Actions on a **common** environment element
 - budget, market, resources, society, ...
- Competitors, government, other projects, ...

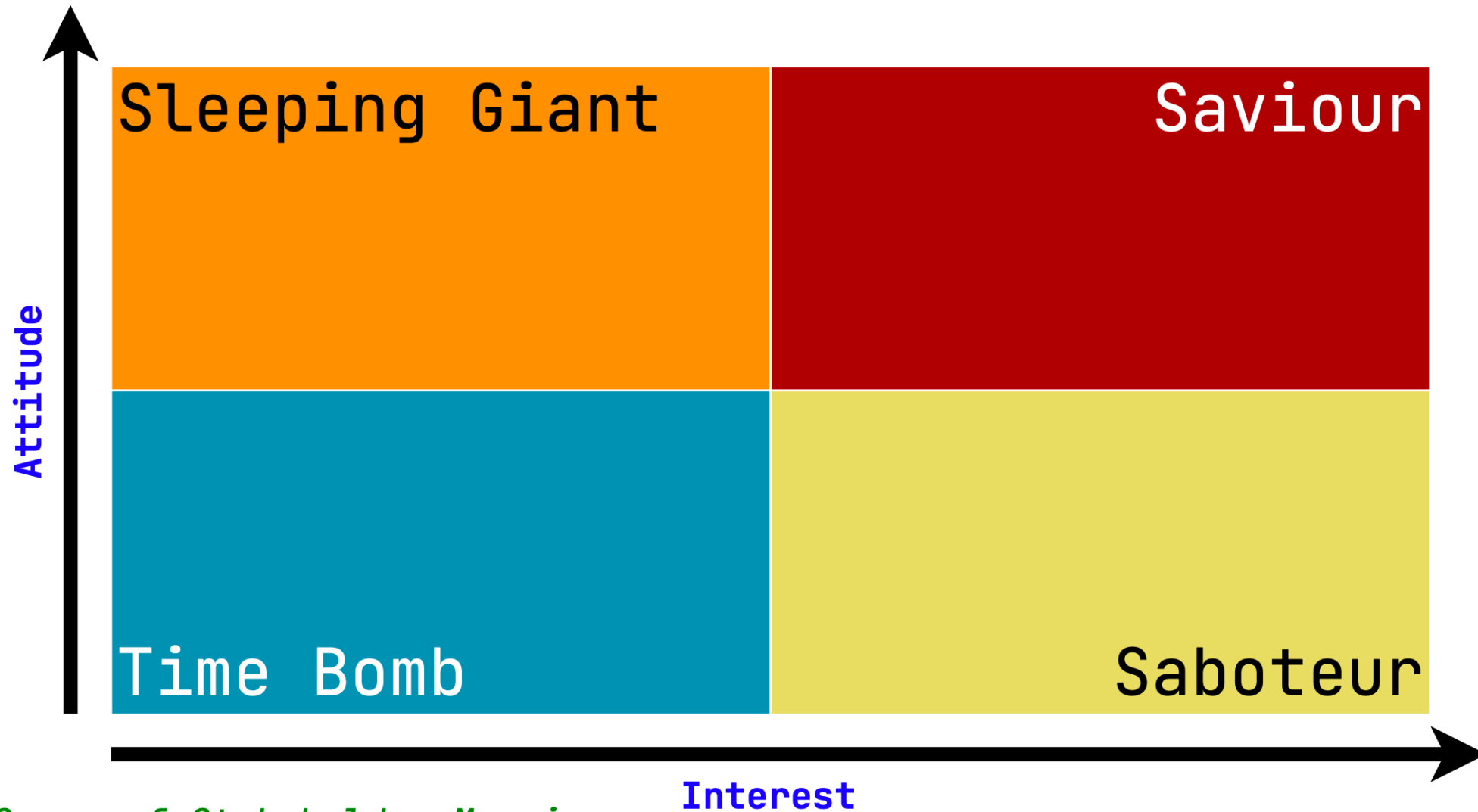
Power-interest matrix



Power-interest matrix

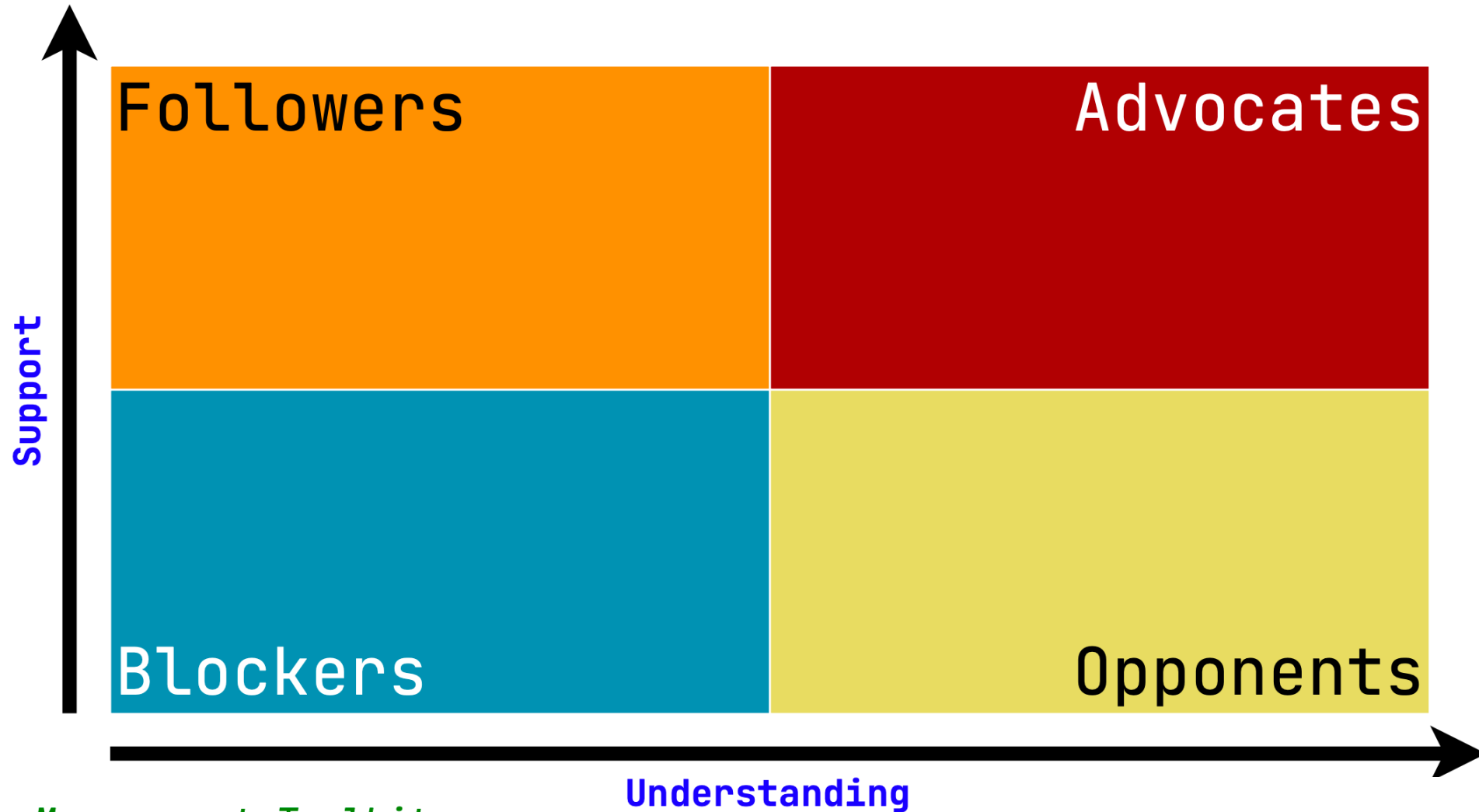


Attitude-interest matrix



Making Sense of Stakeholder Mapping

Support-understanding matrix



Communication plan

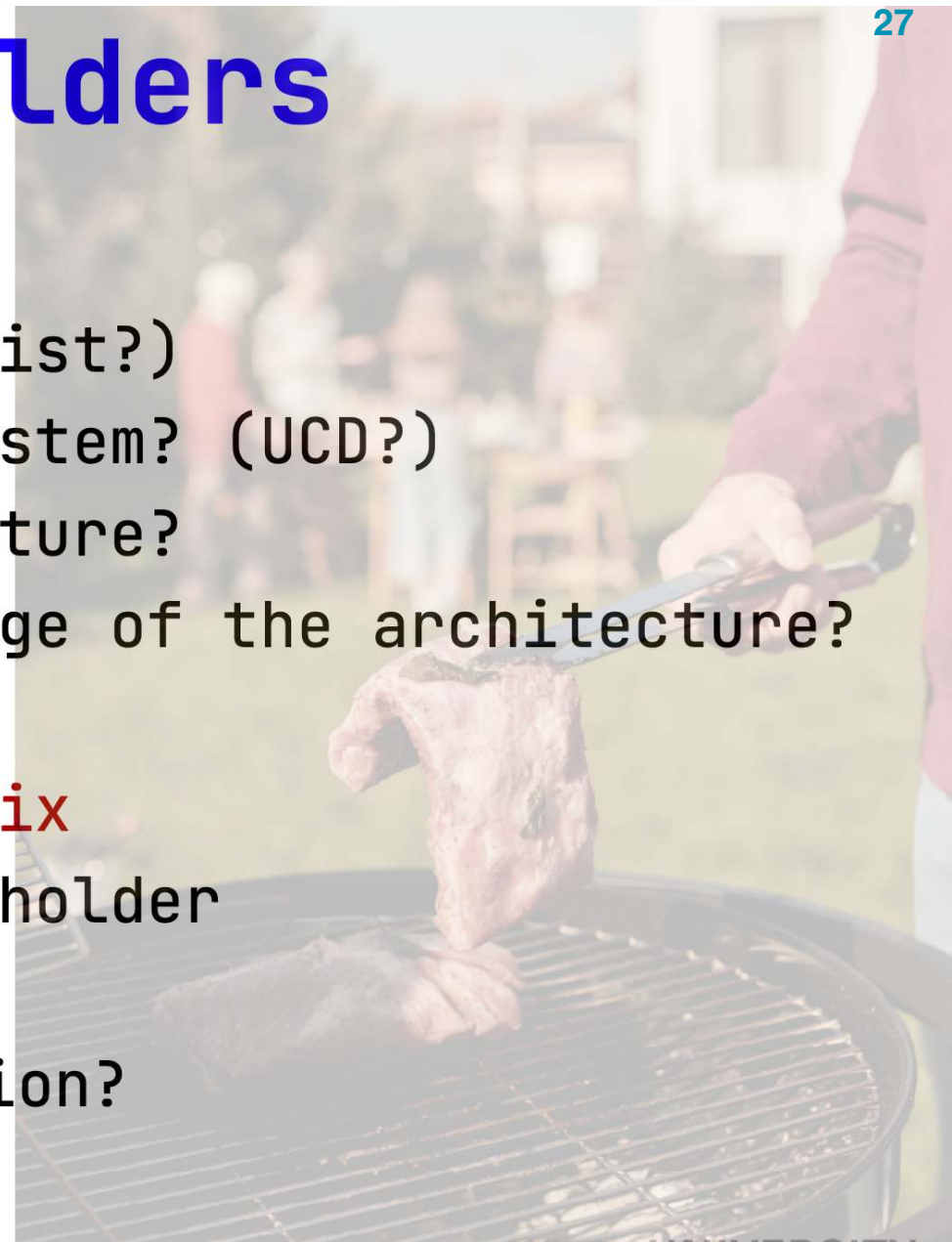
- Which activities/decisions **involve** an architecture?
 - which **architecture information** is needed?
- Communication can be **implicit** or **indirect**
- **Decisions** need to be communicated ASAP
- Need for info **=** concern?
- Choose a **plan** suitable
 - for the stakeholder
 - for the concern

Communication matrix

Who	Stakeholder (group/individual)
Why	What do you want to achieve (inform/think/decide)
About what	Topic of the communication (proposal/decision/...) Which view(point)?
When	At which moment (once/periodically/milestones)
With what	Format (presentation/workshop/meeting/email/call)
By whom	Who is doing it (manager/project leader/architect)

Identify your stakeholders

- What is their **mission**? (Why do they exist?)
- What is their **relationship** with the system? (UCD?)
- What is their **relation** to the architecture?
- How are they **involved** in the dev't/usage of the architecture?
- How do you **communicate** with them?
- Position your stakeholders in the **matrix**
- Decide how you will **involve** each stakeholder
 - (**No** involvement is also a decision)
 - **Frequency** and **type** of of communication?
 - Around which **view(points)**?



Concerns (drivers)

- Concern is
 - something a stakeholder *wants to achieve*
 - or *has interest* in
- Concerns are statements about one or more aspects
- **Every** stakeholder has concerns
 - otherwise, it is not a stakeholder
- Stakeholders take decisions which influence concerns
 - decisions are statements

Concerns vs system properties

- Concern

- interest, requirement, expectation, worry
- typically high-level
- guide the identification of system properties

- System property

- architectural quality attribute / non-functional requirement
- specific attribute or characteristic that the software system must possess to meet stakeholder concerns
- typically quantifiable and can be measured using metrics, tests, or benchmarks

Concerns vs system properties

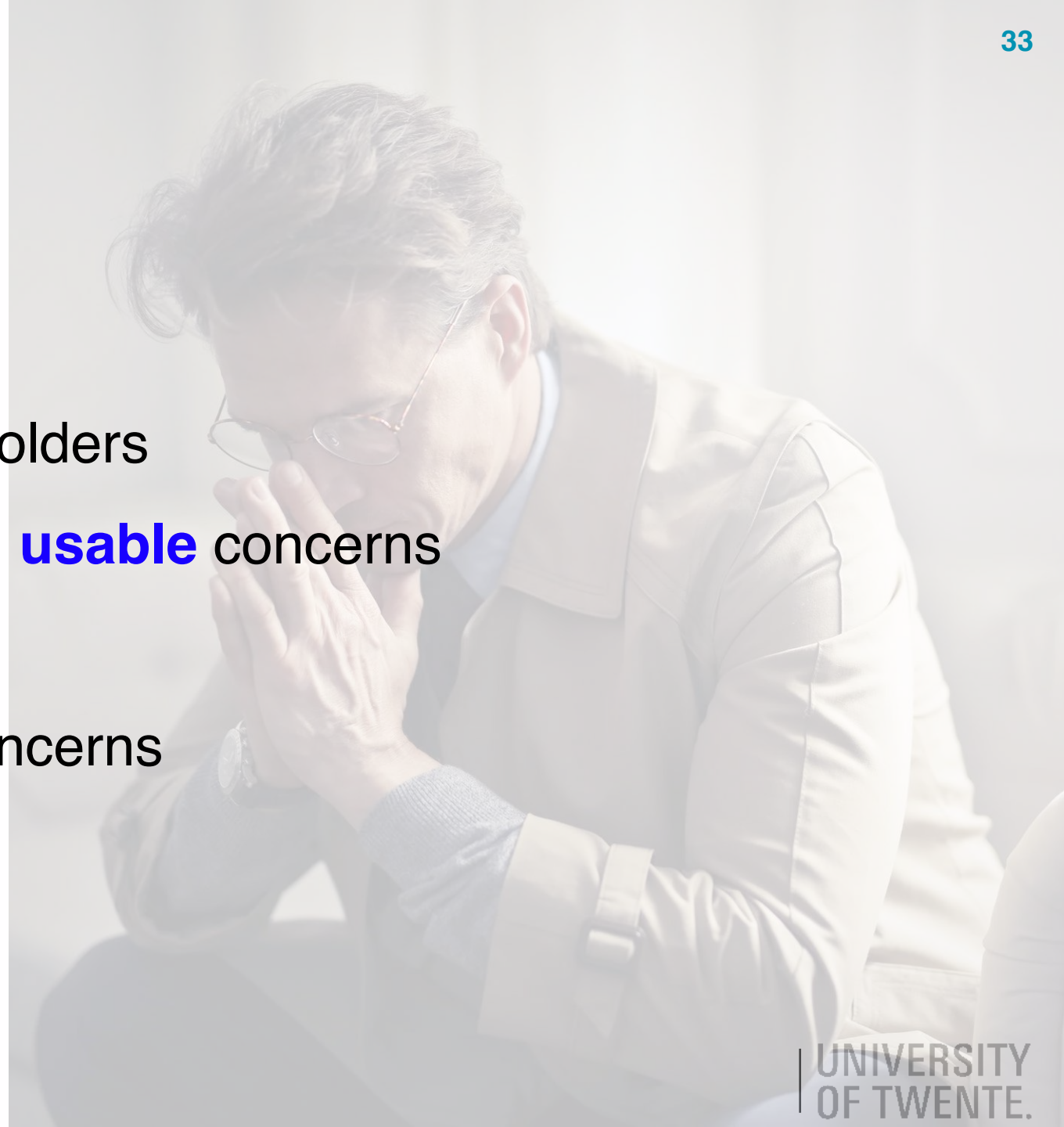


How to find usable concerns?

- Formulate concerns from stakeholder perspective (no **generic** terms!)
 - **goals** to achieve
 - **tasks/activities** to perform
 - information **needs** to satisfy
- Must be usable for trade-offs with other concerns:
 - be clear on **importance** for the stakeholder
 - **prioritise** per stakeholder, then overall
 - know which concerns are **non-negotiable**
- Concerns are the reason behind a **requirement**
- Cast doubt on concerns defined in system terms:
 - find out what the **real** concern is

Usable concerns

- Define stakeholder concerns
 - for **10+** most important stakeholders
- Apply the guidelines for acquiring **usable** concerns
- Brainstorm ideas
 - how to address/solve **top 5** concerns



Homework

H

Make it good

11

Good architecture is correct

- The architecture is based on
 - validated stakeholder concerns in particular
 - (the stakeholder concerns in particular)
- concerns are prioritised
- architecture balances the concerns
- Achieved by:
 - Adequate environment analysis
 - Consistent balance of interests
 - Validation of architectural statements

Good architecture is consistent

- The architecture forms a whole
- Architecture statements do not conflict
- The system can become reality with the architecture
- Achieved by:
 - Verification on contradictions
 - Brainstorming PoIs
 - Conscious management

Good architecture is communicated

- Stakeholders know their relation with the architecture
- Stakeholders know what to do with it
- Stakeholders understand how their concerns and (not) covered architectural goals/descriptions
- Architecture starts with a communicable architectural goal/description
- Achieved by:
 - Accessibility by stakeholders
 - Sufficient anchoring
 - Consistent performance

- Concretise the abstract advice
- Formulate guidelines for your project
 - correctness
 - consistency
 - communication

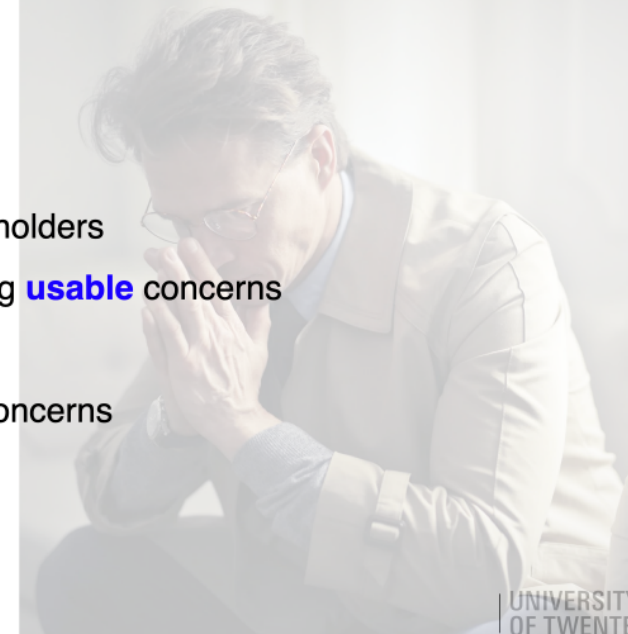
UNIVERSITY
OF TWENTE

H

Usable concerns

33

- Define stakeholder concerns
 - for **10+** most important stakeholders
- Apply the guidelines for acquiring **usable** concerns
- Brainstorm ideas
 - how to address/solve **top 5** concerns

UNIVERSITY
OF TWENTE