


# System Vision, Context and RE Framework

# Course Topics

- ~~Why Requirements Engineering?~~
- ~~Introduction to Requirements~~
- ~~RE in Software Development Life Cycles~~
- System Vision, Context, and RE Framework
- Fundamentals of Goal Orientation
- Fundamentals of Scenarios
- Requirements Discovery
- User Stories and Agile Estimation
- Features Prioritization
- Requirements Negotiation
- Requirements Validation
- Fundamentals of Requirements Management


# Outline



- Vision Statement
  - Advanced Data Sheet
  - System context, System boundary, context boundary during requirements engineering
  - RE framework
- 


# Vision



- The Vision is an outcome of the company's business and portfolio investment strategy.
  - Vision states a goal, not how to achieve it
  - Vision is a guidance throughout the development process
  - **The system vision is a joint vision of the system agreed upon by all active stakeholders**
  - Vision characteristics:
    - Big picture
    - Abstract
  - Vision purpose:
    - Agreement on what this project is about
    - Easy communication with stakeholders
- 

# Example of Vision Statement template




- **For** [target customer]
  - **Who** [statement of the need or opportunity]
  - **The** [product name]
  - **Is** [a product category]
  - **That** [key benefit, compelling reason to buy or use]
  - **Unlike** [primary competitive alternative, current system, or current business process],
  - **Our product** [statement of primary differentiation and advantages of new product]
- 

# Vision Statement Example




**For** scientists **who** need to request containers of chemicals, the Chemical Tracking System **is** an information system **that** will provide a single point of access to the chemical stockroom and vendors. The system will store the location of every chemical container within the company, the quantity of material remaining in it and the complete history of each container's location and usage. This system will save the company 25% on chemical costs in the first year of use by allowing the company to fully exploit chemicals that are already available within the company, dispose of fewer partially used or expired containers and use a single standard chemical purchasing process. **Unlike** the current manual ordering processes, **our product** will generate all reports required to comply with government regulations that require the reporting of chemical usage, storage, and disposal.



# Vision Statement: DriveNow Car Sharing System

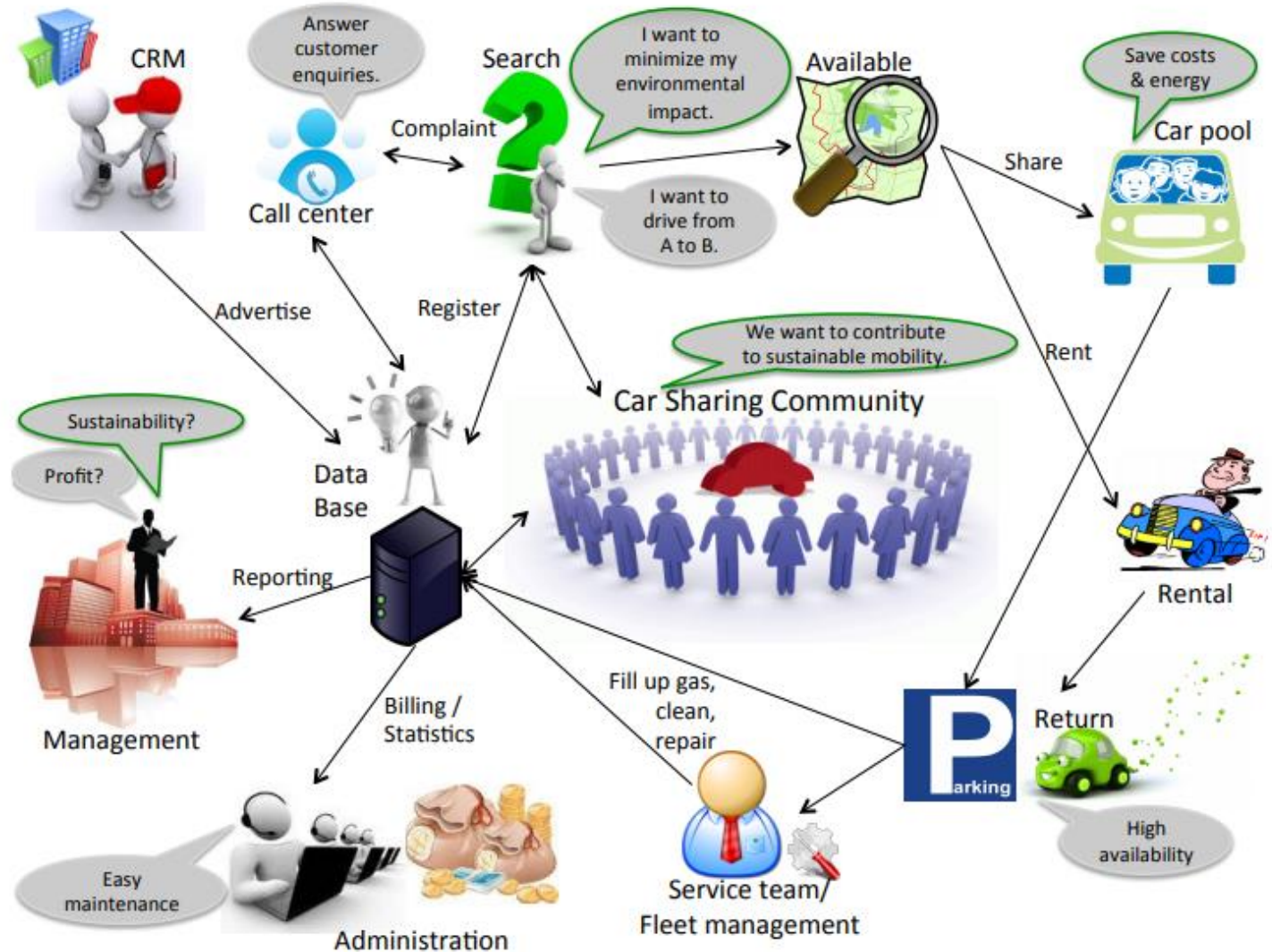


- DriveNow is a **smart system**, developed by BMW, for **individual mobility** and **cost efficient travelling**. It brings **new flexibility** for **city and suburban people**. It can help to **modernize the overall carpool** by enabling drivers to dismiss their old car in favor for a more **economical sharing model**, and helps therefore to **promote emission reduction**.
  - DriveNow offers the possibility to reach new customers in different ways: On the one hand it provides access to **individual mobility for price-oriented customers**. On the other hand it reaches new customers who **need a car only occasionally**.
- 



# DriveNow Car Sharing System Rich Picture

**CRM: Customer Relationship Management**

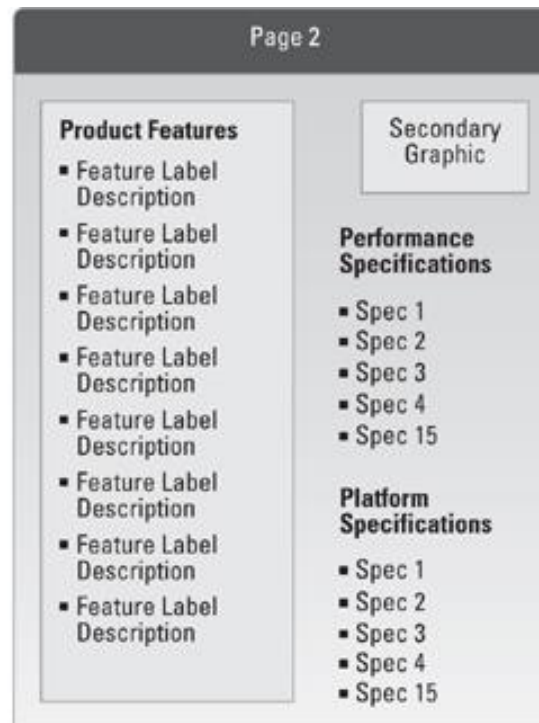
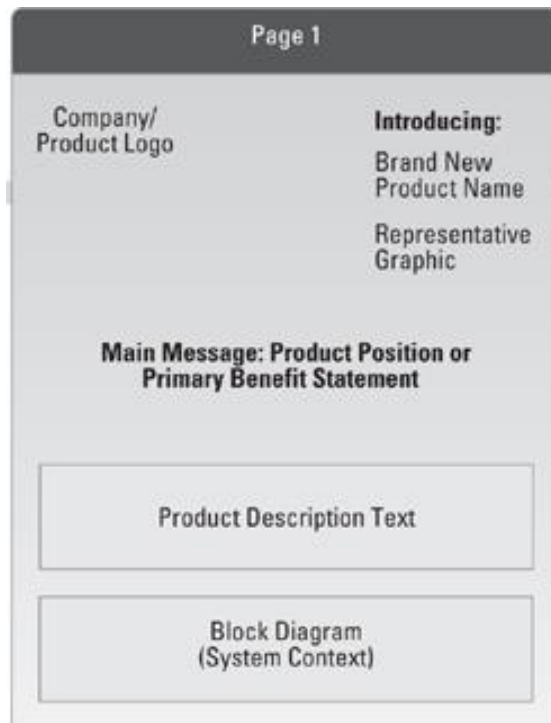


<http://sustainabilitydesign.org/2015/08/28/car-sharing-system/>



# The Advanced Data Sheet Template

- Defines, at a high level, what the product does and for whom it is intended to do it.




# Goal of Requirements Engineering: Establishing a Vision in Context



- Each software-intensive system is embedded within a "**system context**" that contains **the requirement sources** and strongly influences the definition of the system requirements.
- The vision and the system context are thus the two essential inputs for the RE process
- The main goal of RE is to:  
**"establish a vision within an existing context"**

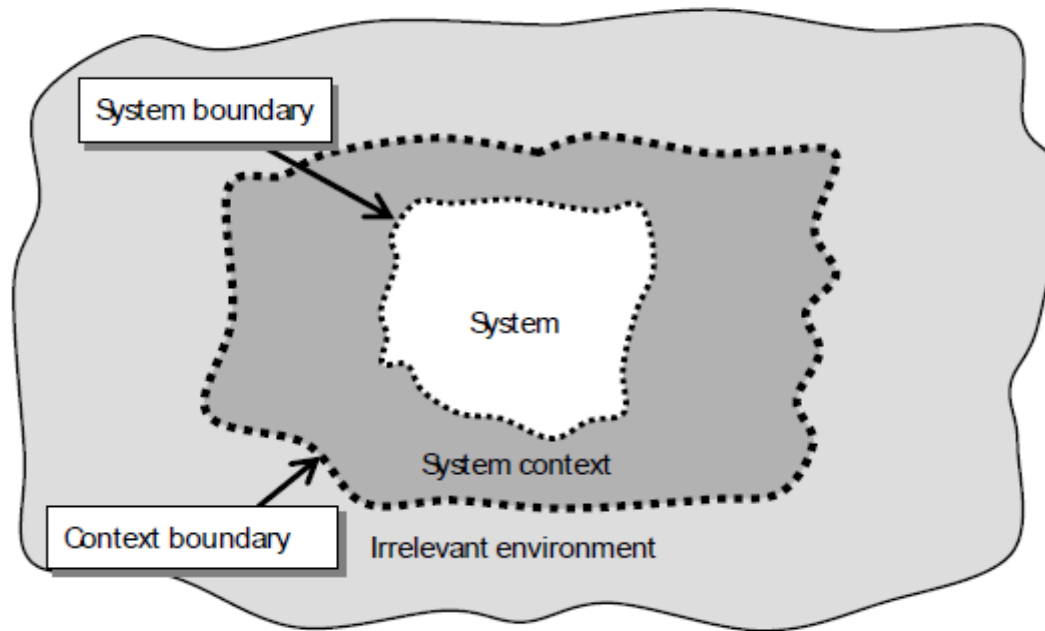
# The Term 'Context'



- Software-intensive systems are always embedded in **environment**
    - Technologies, business processes, existing software components, people, laws, etc.
  
  - **System environment significantly influences requirements for the system**
    - A wrong assumption about the behavior of an external system will most likely lead to an incorrect definition of the requirement specifying interaction with the external system.
- 


# System Context, System boundary, Context Boundary

Stakeholders separate the aspects that **belong to the system** from the aspects that are part of **the system context** or the **irrelevant environment**



# System Context, System boundary, Context Boundary



- The **system context** is the part of the system environment that is relevant for the definition as well as the understanding of the requirements of a system to be developed.
  - The **system boundary** separates the system to be developed from its environment, i.e., it separates the part of the reality that can be modified or altered by the development process from aspects of the environment that cannot be changed or modified by the development process.
  - The **context boundary** separates the relevant part of the environment of a system to be developed from the irrelevant part, i.e., the part that does not have to be considered during requirements engineering.
- 

# Relationship between context aspects and the system

The bank customer uses the ATM to withdraw money from his account.

- Context aspect having a direct interaction relationship with the system

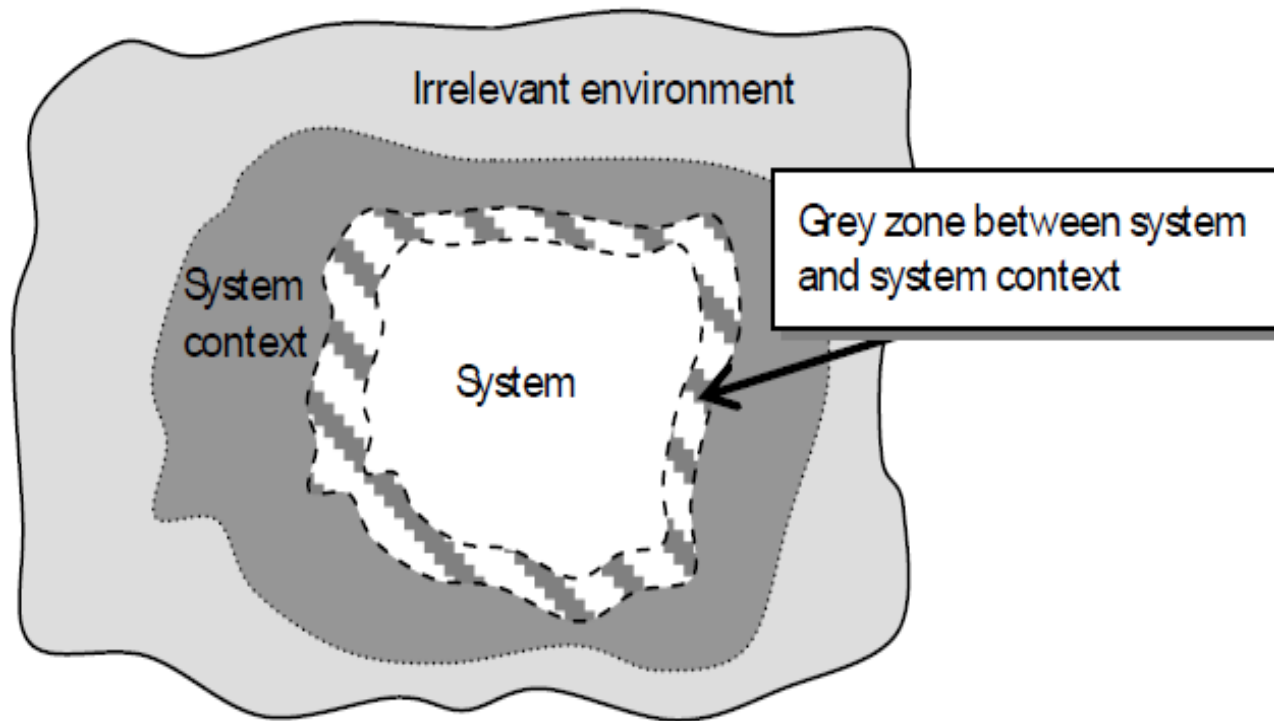
When defining the requirements of the ATM, the specifics of the different customers of a bank must be taken into account (for instance, if the ATM is used by international customers, the user interface must support different languages).

- Context aspect having no interaction relationship with the system but still influencing the requirements of the systems.

A law requires that sensitive data items entered into a banking system and used within the system are encrypted using a certain encryption standard.

## Grey Zone between the System and the Context

During requirements engineering, the system boundary as well as the interfaces are **typically unstable**, i.e. **they change quite frequently**.





# Adjusting the system boundary and gray zone


During the development of a navigation system, the stakeholders define a scenario 'navigate to a point of interest'. In this scenario, it is the responsibility of the driver to look up and enter the address of the point of interest to navigation system.

- This decision requires an adjustment of the system boundary since looking up the address of a point of interest is now within the responsibility of the system.

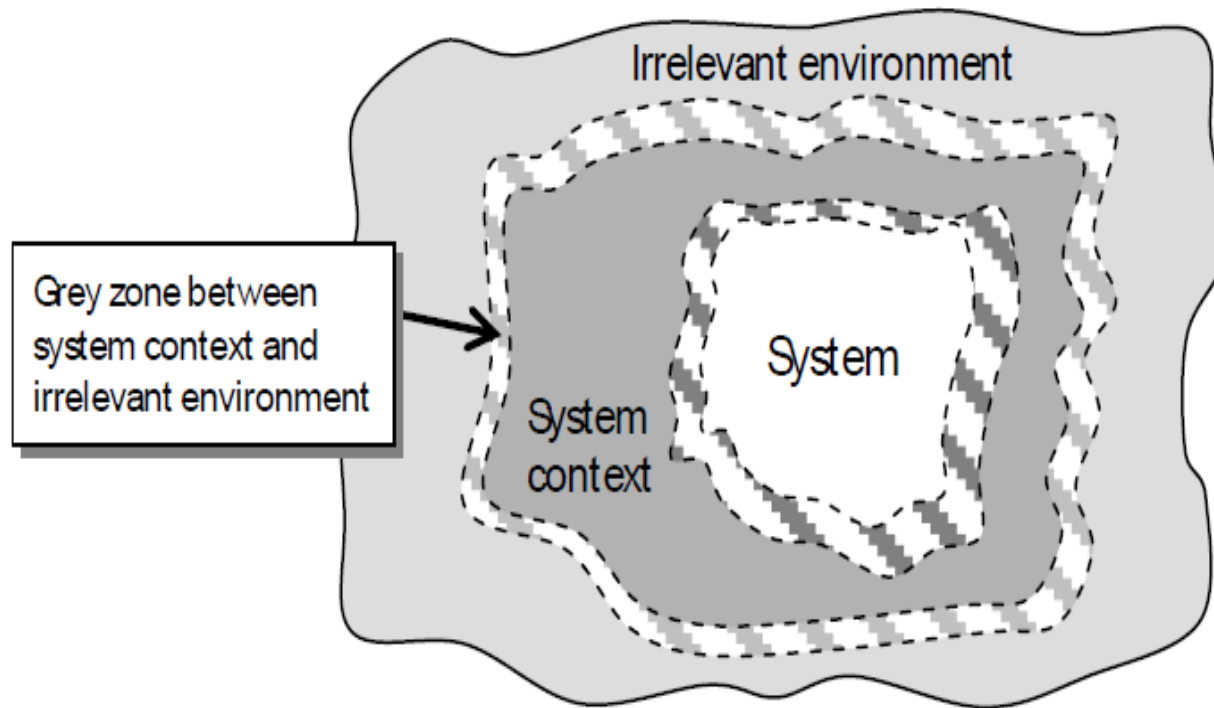
Later on, the stakeholders decide that the system shall, at the request of the driver, display the 15 points of interest of a certain category (e.g., fuel station) nearby and allow the driver to select the desired destination.

# Hints about System Boundary



- Determine explicitly which aspects belong to the system.
  - Determine which aspects are outside the system boundary.
  - When defining the system boundary **involve all relevant stakeholders**.
  - Try to reach an agreement about the system boundary. In case you cannot decide whether some object belongs to the system or not, put it in the **grey zone**.
  - **Check periodically** whether the defined system boundary is still valid. Pay attention to required extensions or reductions of the system boundary.
  - If the system boundary needs to be adjusted, verify whether the adjustment impacts the already defined requirements.
- 

# Gray Zone between the System Context and the Irrelevant Environment




# Gray Zone between the System Context and the Irrelevant Environment




A new GPS enabled mobile phone has initially been designed with the goal of protecting user privacy according to the current laws of the European Union. Special focus was placed on the system's discretion with user data – no sensitive data about the user shall leak out to service providers.

However, during requirements engineering, it becomes obvious that the system does not transmit any user data to the service provider or any other system/actor. Consequently, the European privacy laws no longer have to be considered and are thus defined as part of the irrelevant environment.



# Hints about defining the context boundary



- If you are unsure whether some context aspect impacts the system requirements or not, assign it to the gray zone.
  - When you come to the conclusion that one context aspect is irrelevant for the system, **document this aspect** as a part of the **irrelevant environment**. This allows you to re-check the relevance of this aspect, for instance, when new system functionality is added.
  - When defining **new requirements** (e.g., functions) check whether context aspects (e.g., a law) classified as irrelevant so far become relevant due to the new requirements.
  - Use **goals and scenarios** to check whether specific aspects of the environment are relevant for the system or not. If an aspect is relevant, it should affect at least one goal or one scenario.
- 

# Need to Document Context Aspects

- Requirements depend on the context.
- Example of a documented requirement with vague (almost no) context information.

**R: The planned means of transportation shall offer travellers a fast journey to their destination.**

- Depending on the context information, the requirement has different interpretation.
  - For example, not clear whether the transportation shall be performed by air, sea, etc.

# Need to Document Context Aspects

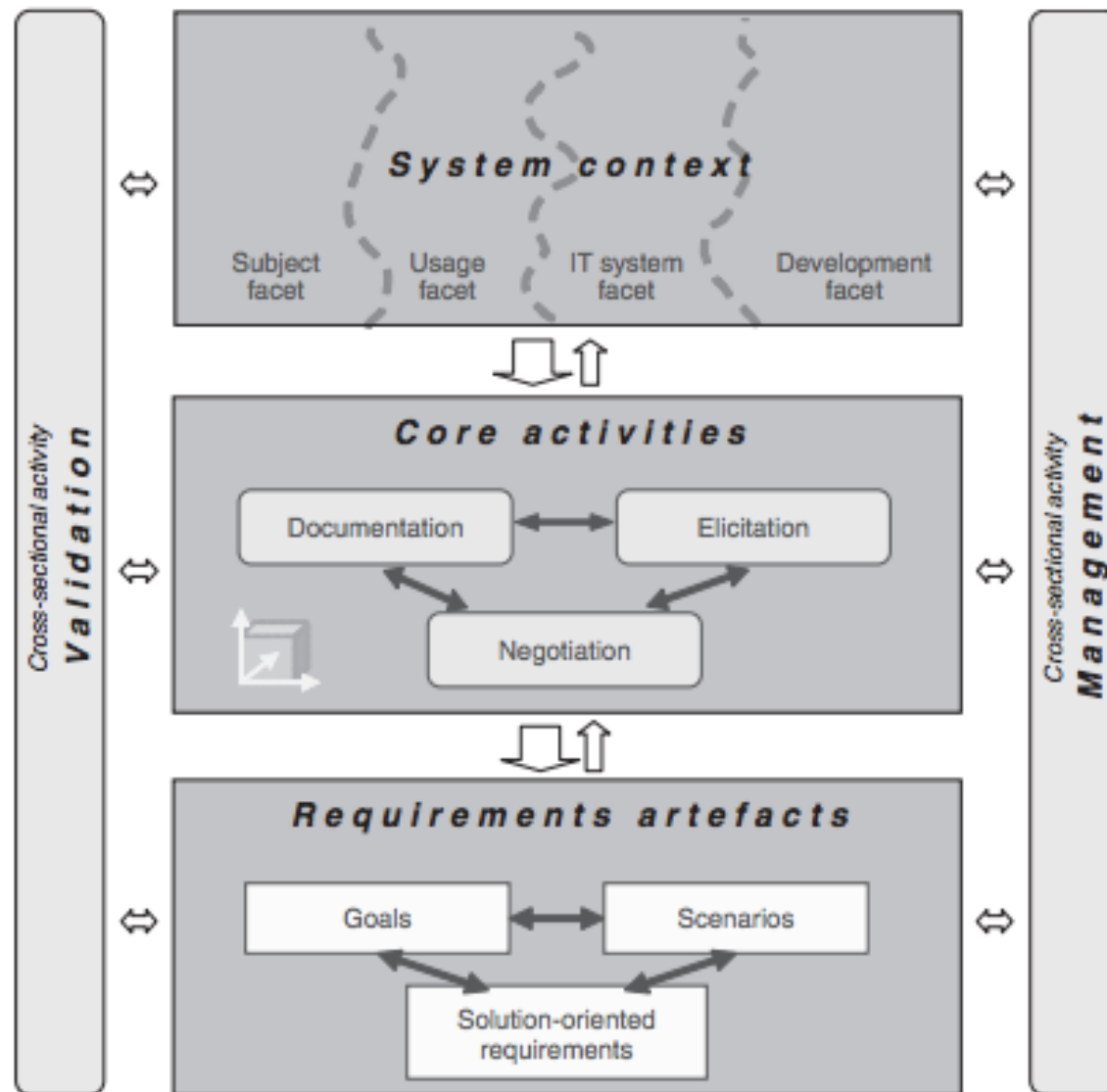
- The context information will reduce the potential interpretations of the requirements significantly.

The system shall transport people from an island to the mainland, and there is no airstrip on the island.

- Note, the island and the mainland **cannot be changed by the system development process and thus they belong to the system context.**



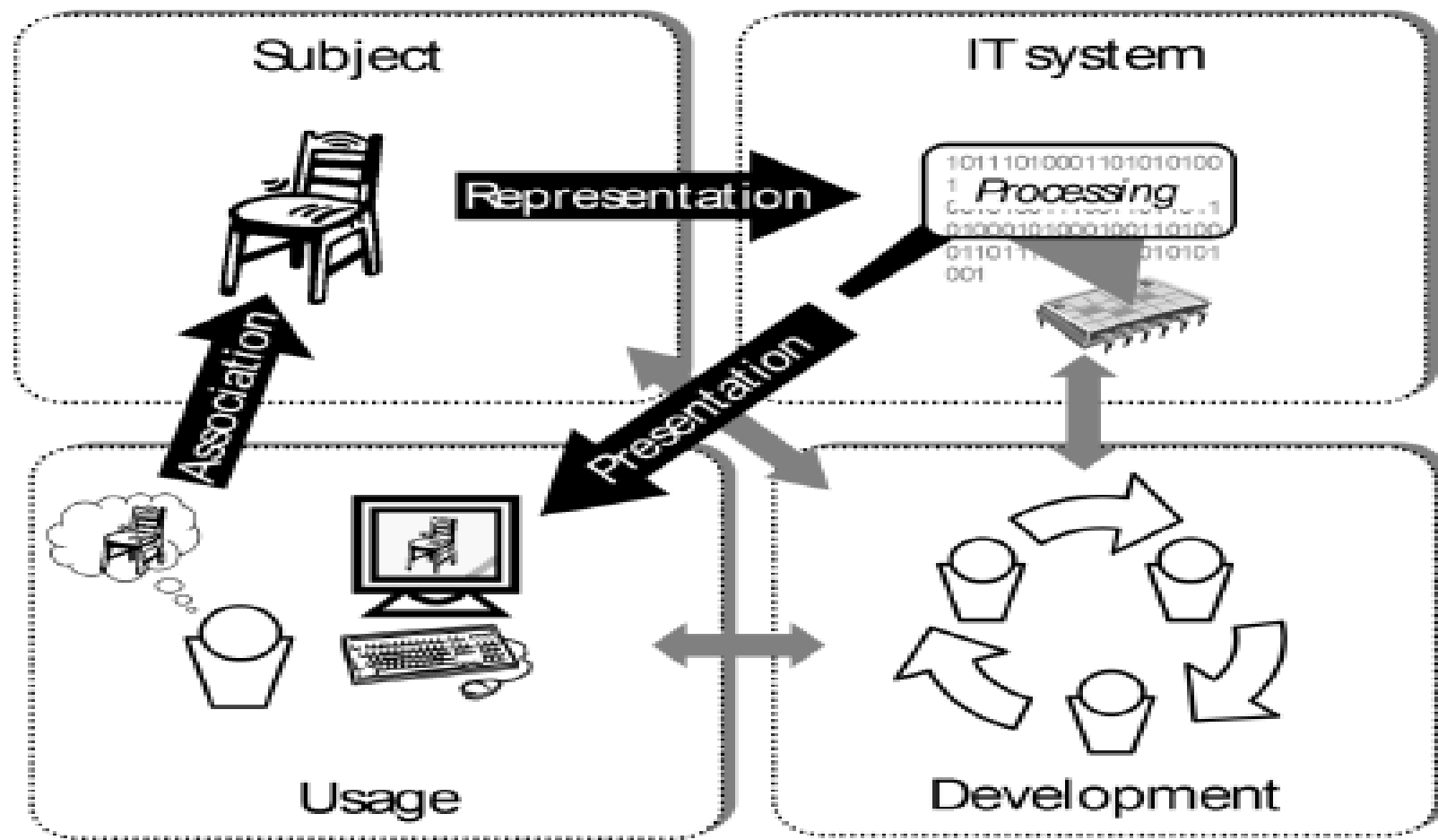
# The Requirement Engineering Framework



# System Context: Four Facets

- **Subject facet:** Objects and events relevant for the system.
  - Example: Elements the system must store or process information about.
- **Usage facet:** Aspects concerning the usage by people or other systems.
  - Example: Different user groups with specific characteristics, laws and standards restricting or influencing the system usage.
- **IT system facet:** Objects and elements of the IT system environment of the system
  - Example: Existing hardware and software components to be used
- **Development facet:** Aspects concerning the development process of the system
  - Example: process guidelines, development tools

# Logical relationships between the four context facets




# In-class Activity



Consider the Online Student Registration System.

Define the following:

- Subject facet
  - Usage facet
  - IT system facet
  - Development facet
- 

# Three RE Core Activities



- Three core activities:
  - Elicitation (content dimension)
  - Negotiation (agreement dimension)
  - Documentation (documentation dimension)

# Three RE Core Activities




## ■ Elicitation

- Requirements are elicited from stakeholders and other sources
- New and innovative requirements are collaboratively developed.
- Goal is to improve the understanding of the requirements

## ■ Negotiation

- All conflicts between the viewpoints of the different stakeholders have to be detected and made explicit.
- The identified conflicts should be resolved.

## ■ Documentation

- Documentation and specification of the elicited requirements according to the defined documentation and specification rules.
- 


# Two Cross Sectional Activities



## ■ Validation

- Validation of the requirements artefacts, core activities, and consideration of the system context


## ■ Management

- Management of artefacts and activities
  - Observation of the system context
  - Activities are interrelated, for example performing one activity may require the execution of additional activities
- 




# Cross Sectional Activity: Validation



- Validation of requirements artefacts
    - Detecting defects in the requirements.
  - Validation of the core activities
    - Checking the compliance between the activities performed and the process and/or activity specifications, e.g., one should validate whether the steps defined for an activity and the defined follow-up activities have been performed.
  - Validation of the consideration of the system context
    - Validating whether the system context has been considered in the intended way during requirements engineering.
    - All relevant stakeholders have been involved in the process at the right time.
- 

# Cross Sectional Activity: Management



- Management of the requirements artefacts
    - **Prioritization** of requirements
    - Persistent recording of requirements.
  - Management of the activities
    - Planning and control of requirements engineering activities
    - Ensure an efficient and effective requirements engineering process.
  - Observation of the system context
    - For example, it might require the execution of an elicitation activity and a documentation activity in order to document the new requirements caused by a change.
- 

## Interrelations between the Five Activities: 3 core and 2 cross sectional


One activity typically causes the execution of additional RE activities.

**Example1:** Elicitation of additional requirements:

During an interview (i.e., **elicitation activity**), new requirements are identified. However, **the documentation** of the new requirements in the interview minutes is **not in compliance with the project-specific documentation rules**. Thus **an additional task** is created, namely the **documentation of the new requirements** so as to be in compliance with the defined rules.

In addition, the new requirements should be **agreed** between the stakeholders involved. Thus, **a new validation activity** is performed to check whether the stakeholders agree with the new requirement. During the validation of the agreement, **conflicts** about the requirement between the involved stakeholders might be identified. If so, these **conflicts need to be resolved** and the outcome of the **conflict resolution must be documented and comply with the documentation rules**, etc.

## Interrelations between the Five Activities: 3 core and 2 cross sectional



### ■ Example 2: Detection of a missing requirement

While **reviewing** a set of requirement artefacts (i.e., **validation activity**), the stakeholders detect that an **important requirement has been omitted**. The stakeholders briefly sketch the omitted requirement. Obviously, the new requirement is not yet documented in compliance with the defined documentation rules. Moreover, **the documentation of the new (previously omitted) requirement does not contain all the required information, and not all the stakeholders have yet agreed to the new requirement**. The identification of a new requirement during the validation activity thus might lead to the execution of **additional elicitation, documentation, and negotiation activities**.

## Interrelations between the Five Activities: 3 core and 2 cross sectional



- Example 3: **Removal of a requirement from the specification**

**Negotiations** between customers and system users result in the **removal of a requirement** from the specification (as a resolution of a conflict for example).

The elimination of this requirement requires an **evaluation of whether other requirements artefacts are affected by this change**. The related requirements artefacts thus have to be **analyzed and the documented artefacts have to be adjusted accordingly**, if required.

# Requirements Artefacts: Goals

- A goal is an Intention with regard to the objectives, properties or use of the system
- A goal should be solution free
  - Stakeholders typically have many different alternatives for satisfying a goal

**Goal1** : The system shall guide the driver to a desired destination automatically.

**Goal2** : The response times of the system shall be 20% lower compared with the predecessor system

## Requirements Artefacts: Scenarios

- A scenario describes a concrete example of satisfying or failing to satisfy a goal (or a set of goals). A scenario typically defines a **sequence of interaction steps** executed to **satisfy the goal** and relates these interaction steps to the system context.

### Scenario Example:

Carl drives his car on the motorway at a speed of 50 mph. Peter, the driver of the car ahead of Carl, steps on the brake pedal firmly. After recognizing that the car in front is braking, Carl pushes on the brake pedal as well. The on-board computer of Carl's car detects that the safety distance to Peter's car is no longer maintained and issues a warning to the driver.



## Requirements Artefacts: Solution-oriented requirements

- Define the data, functions, behavior, quality, and constraints
- Often imply a conceptual solution
- Data models, for instances, defines entities, attributes, and relationships between entities
- A behavioral model defines the states of the system and externally visible behavior
- **The three requirements artefacts types (Goals, Scenarios and solution-oriented) are used complementarily**