NÕUETE VALIDEERIMINE
(REQUIREMENTS VALIDATION)
Validation

Check whether the **outputs** of RE activities fulfill defined quality criteria

Check whether the **inputs** of RE activities fulfill defined quality criteria

Check whether the **execution of RE activities** adheres to process definitions and activity guidelines
Validation

Check whether the outputs of RE activities fulfill defined quality criteria

Check whether the execution of RE activities adheres to process definitions and activity guidelines

Content
Agreement
Documentation
Requirements Lifecycle

Source: Adapted from Pohl, CAISE 1993

Diagram showing the relationship between different dimensions of requirements, including specification (content), agreement, and representation (documentation), with axes for completeness, fairness, and vagueness, and levels of formality (informal, semi-formal, formal).
Validating
Created Requirements Artefacts

Validation with regard to

• **Content dimension:**
  – Check whether all relevant requirements are known and understood to the required level of detail

• **Documentation dimension:**
  – Check whether the requirements are documented according to the defined documentation and specification rules

• **Agreement dimension:**
  – Check whether the stakeholders have reached agreement about the documented requirements
  – Check for all known conflicts; have they been resolved?
  – Check whether there are conflicts that have not yet been identified
Validation

Check whether the **outputs** of RE activities fulfill defined quality criteria

Check whether the **execution of RE activities** adheres to process definitions and activity guidelines

Check whether the **inputs** of RE activities fulfill defined quality criteria

- Stakeholder facet
- Usage facet
- IT system facet
- Development facet
Context defects

- **Missing context information:**
  - Important requirements have not been identified

- **Incorrect context information:**
  - Requirements are defined based on incorrect context information:
    - E.g., wrong assumptions

- **Insufficiently considered context information:**
  - Relevant context has not been adequately documented

- **Incomplete sources of requirements:**
  - Not all relevant sources of requirements have been considered
Validation

- Check whether the **outputs** of RE activities fulfill defined quality criteria
- Check whether the **inputs** of RE activities fulfill defined quality criteria
- Check whether the **execution of RE activities** adheres to process definitions and activity guidelines
Validation of Execution of RE Activities

- Has the execution of the RE activities been documented in the prescribed way?

- Have all RE activities been performed that are required according to the process definition?

- Have all inputs defined for each RE activity been considered for the respective activity?
Validation of Execution of RE Activities

• Does the execution of each RE activity correspond to the rules and guidelines defined in the description of the respective activity?

• Have all outputs defined in the description of each RE activity been created by the execution of the respective activity?

• Have all stakeholders who are relevant for performing the respective RE activity been involved in the execution of the activity?
Model analysis

• Requirements specification
• Goal models
• Scenario models
• Information models
• Behavioral models
• Interaction models
Requirements Specification

1 Introduction
   - Purpose
   - Scope
   - Definitions, acronyms, abbreviations
   - Reference documents
   - Overview

2 Overall Description
   - Product perspective
   - Product functions
   - User characteristics
   - Constraints
   - Assumptions and Dependencies

3 Specific Requirements

Appendices

Index
Goal-Scenario coupling

... initiate and influence the definition of ...

... classify ...

Goals

Scenarios

... illustrate satisfaction ...

... lead to the identification of new ...

... lead to revision of ...
Key Relationships

- Goals
- Scenarios

Elicitation, refinement and validation of solution-oriented requirements

Refinement of existing and elicitation of new goals and scenarios
We have looked at the following non-UML diagrams

- Goal Models
- Organization Models
- Domain (Knowledge) Models

These models should be validated for consistency between each other!
Use cases

Use Cases:
- Capture the view of the system from the view of its users
- Good starting point for specification of functionality
- Good visual overview of the main functional requirements

Cross-checks:
- Does each use case have a user?
  - Does each user have at least one use case?
- Is each use case documented?
  - Using sequence diagrams or use case templates
Class diagrams

**Class Diagrams:**
- Capture the structure of the information to be handled by the system
- Good for analysing the relationships between the data items handled by the system
- Good for helping to identify a modular structure for the system

**Cross-checks:**
- Does the class diagram capture all the classes mentioned in
  - other diagrams?
  - specification?
  - glossary?
- Does every class have methods to get/set its attributes?
**State diagrams:**
- Capture all possible responses of an object to all use cases in which it is involved
- Good for modelling the dynamic behaviour of a class of objects
- Good for analysing the ordering of events, reachability, deadlocks, etc.

**Cross-checks:**
- Does each state diagram capture (the states of) a single class?
  - Is that class in the class diagram?
- Does each transition have a triggering event?
  - Is it clear which object initiates each event?
  - Is each event listed as an operation for that object’s class in the class diagram?
- Does each state represent a distinct combination of attribute values?
  - Is it clear which combination of attribute values constitutes the state?
  - Are all of those attributes shown on the class diagram?
- Are there method calls in the class diagram for each transition?
  - …a method call that will update attribute values for the new state?
  - …method calls that will test any conditions on the transition?
  - …method calls that will carry out any actions on the transition?
Sequence Diagrams:
• Capture an individual scenario (one path through a use case)
• Good for modelling a dialog structure for a user interface or a business process
• Good for identifying which objects (classes) participate in each use case
• Helps to check that the necessary classes and operations have been considered

Cross-checks:
• Is each class in the class diagram?
• Can each message be sent?
  – Is there an association connecting sender and receiver classes on the class diagram?
  – Is there a method call in the sending class for each message sent?
  – Is there a method call in the receiving class for each message received?