Validation of
- Context of consideration
- Execution of RE activities
- Created requirements artefacts
Validation Goal

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

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

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

- Content
- Agreement
- Documentation
Requirements Lifecycle

Source: Adapted from Pohl, CAISE 1993
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 all known conflicts; if they have been resolved
  - Check whether there are conflicts that have not yet been identified
Validation Goal

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

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

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

Subject 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 on incorrect context information
    • E.g., wrong assumptions

• **Insufficiently considered context information**
  – Relevant context not adequately documented

• **Incomplete requirements sources**
  – Not all relevant requirements sources were considered
Validation Goal

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

• Has the execution of the activities been documented in the prescribed way?

• Have all activities that are required according to the process definition been performed

• Have all inputs defined in each activity description been considered for the respected activity
Validation of Execution of Activities

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

• Have all **outputs** defined in each activity description been created?

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

- Requirements specification
- Goal models
- Scenario models
- Static models
- Behavioral models
- Functional 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 ...

... illustrate satisfaction...

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

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

Elicitation, refinement and validation of solution-oriented requirements

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

- **Goal Models**
  - Capture strategic goals of stakeholders
  - Good for exploring ‘how’ and ‘why’ questions with stakeholders
  - Good for analysing trade-offs, especially over design choices

- **Strategic Dependency Models (i*)**
  - Capture relationships between actors in an organisational setting
  - Helps to relate goal models to organisational setting
  - Good for understanding how the organisation will be changed
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 template
Class diagrams

Class Diagrams
• capture the structure of the information used by the system
• good for analysing the relationships between data items used by the system
• good for helping you 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?
**Statecharts**
- capture all possible responses of an object to all use cases in which it is involved
- good for modeling the dynamic behavior of a class of objects
- good for analyzing event ordering, reachability, deadlock, etc.

**Cross-checks:**
- Does each statechart diagram capture (the states of) a single class?
  - Is that class in the class diagram?
- Does each transition have a trigger 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?
  - Are all 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 dialog structure for a user interface or a business process
- good for identifying which objects (classes) participate in each use case
- helps you check that you identified all the necessary classes and operations

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 sent message?
  - Is there a method call in the receiving class for each received message?
Validation of
- Context of consideration
- Execution of RE activities
- Created requirements artefacts