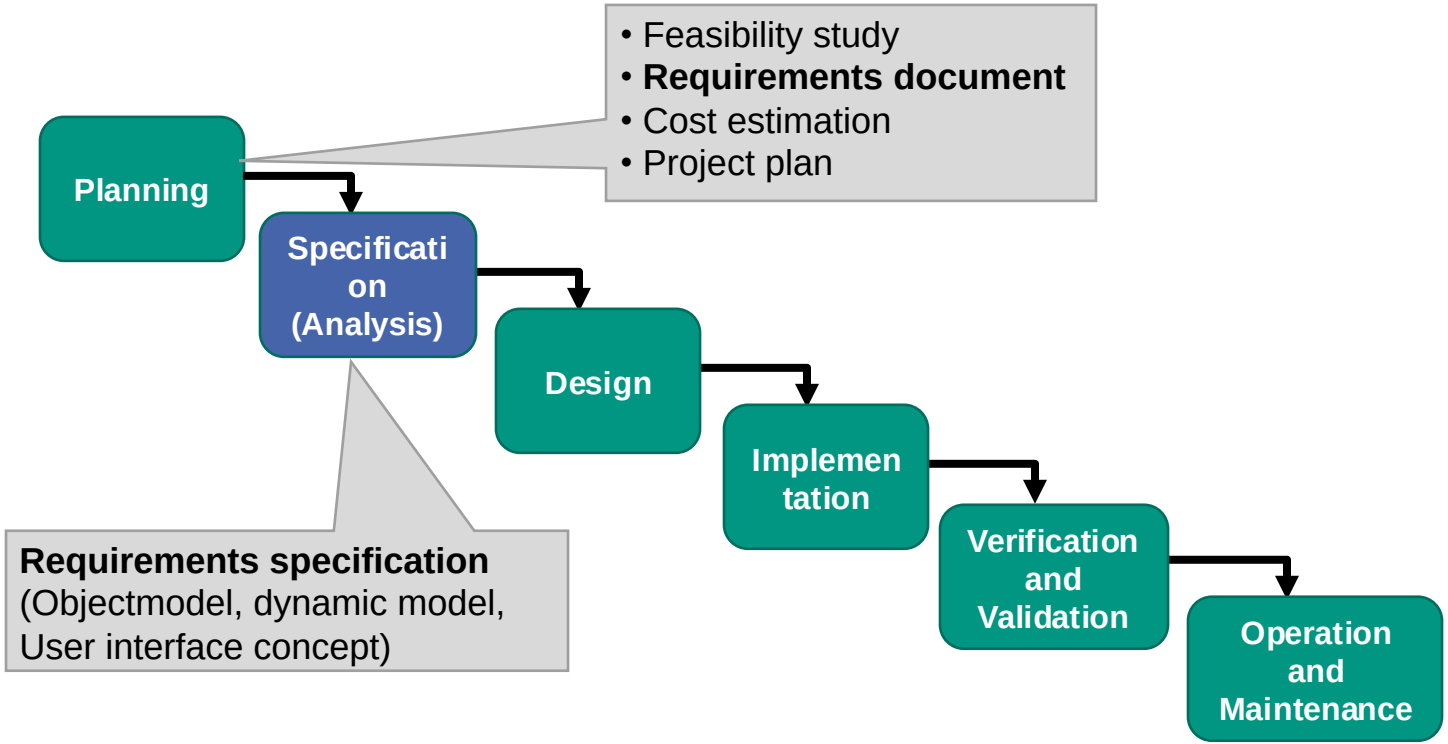
Introduction to Software Engineering

The Specification Phase

Prof. Walter F. Tichy

ქუთაისის
საერთაშორისო
უნივერსიტეტი

Where are we?


• Feasibility study
• **Requirements document**
• Cost estimation
• Project plan
Planning
Specificati on (Analysis)
Design
Implemen tation
Verification and Validation
Operation and Maintenance
Requirements specification
(Objectmodel, dynamic model, User interface concept)

Learning goals

- Define the specification phase and its role in the development process
- Produce object models and dynamic models (in UML)
- Write a requirements specification using the template provided

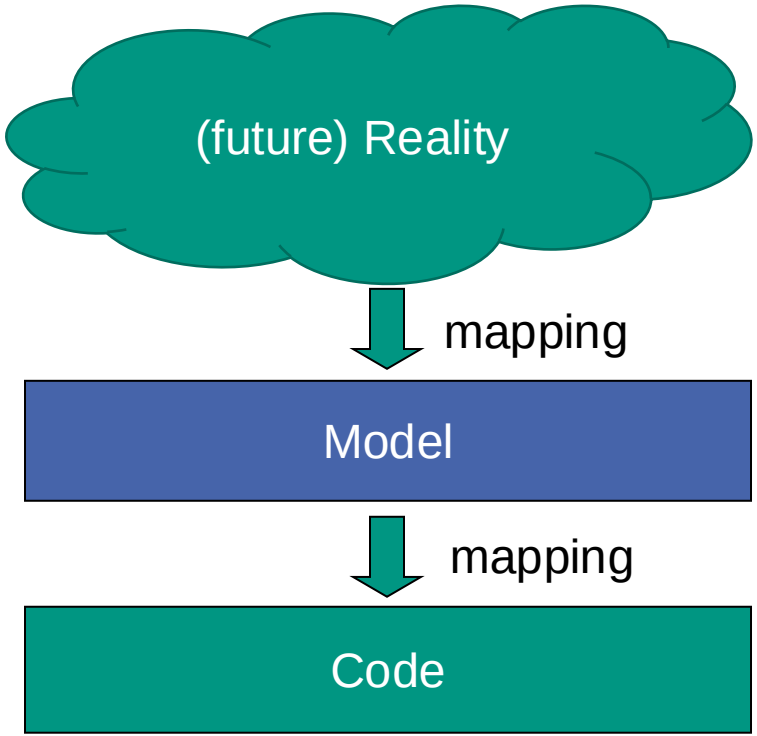
Specification Phase (aka Analysis Phase)

- The specification phase produces the **requirements specification**.
- The requirements specification defines or „models“ the system to be developed (or the change to an existing system) completely and exactly, so that developers can implement the system (or the change) without having to ask for more information or to guess what is to be done.
- The requirements specification defines the „what“ but not the „how“ of what to implement (e.g., algorithms and data structures are not provided)
- The requirements specification is a refinement of the requirements document (the template is nearly the same).
- While the requirements document is for communication with the user, the requirements specification is for communication with the developers.

Modeling

- The requirements spec provides a model of the system to be implemented.
- Types of models:
  - Functional model (from the requirements document)
    - scenarios, use case diagrams
  - Object model
    - class and object diagrams
  - Dynamic model
    - sequence diagrams
    - state charts
    - activity diagrams

Where does the model go?



Model and Reality

- Reality R
  - Real objects, persons, concepts, etc.,
  - Processes that take a certain time,
  - Relations between objects, persons, concepts, processes, etc.,
- Model M: Abstractions of existing or imaginery
  - objects, persons, concepts, etc.,
  - processes,
  - Relations among them

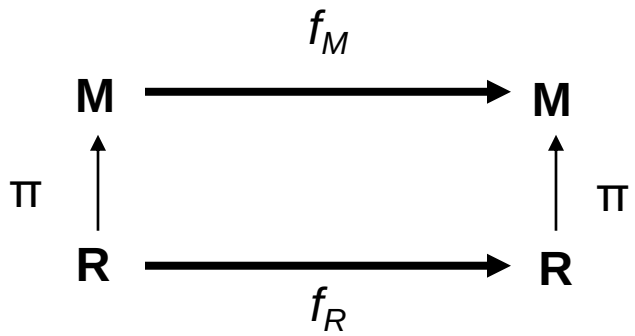
What is the Purpose of a Model?

- In general, we use models…
  - To abstract from the details of reality, which allows us to reason about the complex reality by using simple, defined deduction steps in the model.
  - To obtain insights about the past or present
  - To make predictions about the future.

- We use software models
  - To obtain an understanding about the future reality (the software)
  - In other words, we use models to see what we would get, if we would implement the model.

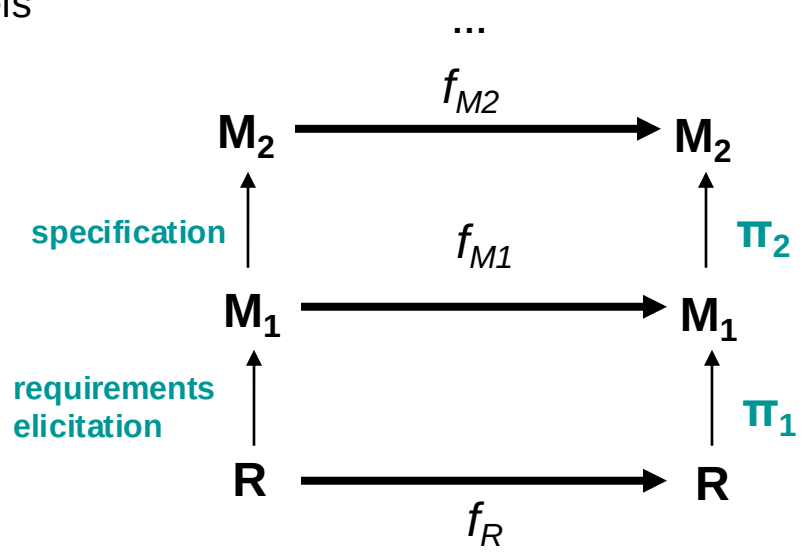
What makes a good model?

- Relations valid in reality R are must also be valid in model M.
  - π : mapping of reality R to a model M (abstraction)
  - f_M: relations between abstractions in M
  - f_R: equivalent relations between real elements in R
- In a good model, the following diagram is commutative:



Metamodeling: models of models...

- Modeling can have several abstraction steps
- For instance, a use case is more abstract (has less detail) than a scenario
  - Software development can
    can be viewed as transforming models



“Realities” of the software engineer

- Software engineers can model different realities:
  - an existing system (physical, technical, social, or existing software) and build it.
    - An existing software system is an important case: Model the software before attempting to adapt or change it. An older system is often called „legacy system“
  - An idea or future system without a real counterpart.
    - A visionary system or customer requirement

Types of Models