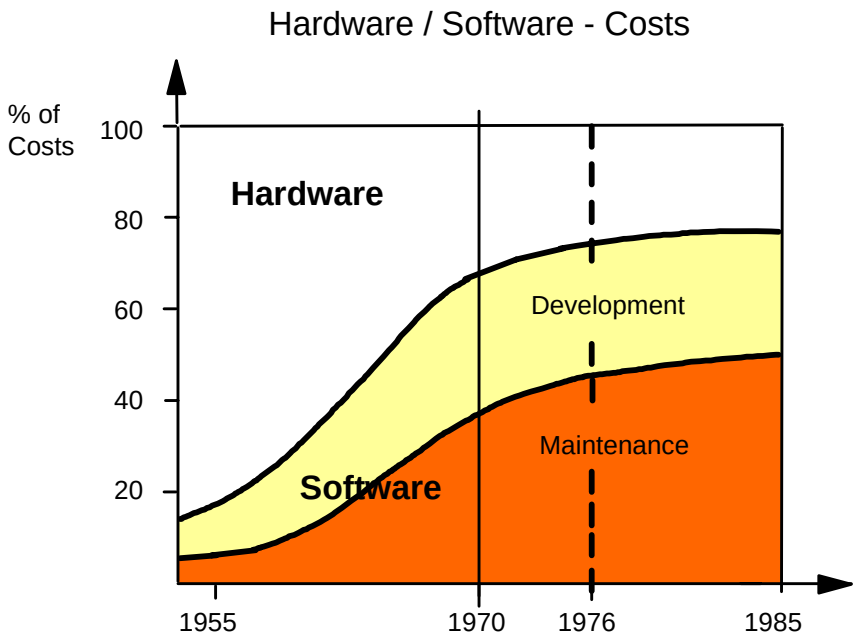
1. The Context: Software Engineering Basics

Aims

- Motivation
- Definition: Architecture Modelling
- Interaction with other working areas

Contents of Chapter 1

Software Crisis (due to the Maintenance Problem)



Software costs and maintenance problem

- No absolute cost figure
  Germany: about 150 Billion $?
- Trend?
  positive: decreasing SW costs
  negative: new tasks/domains
- Is graph still valid?

„Software Engineering“

- Software Engineering Definitions

  (Term SE coined at NATO Conferences Garmisch, Rome 69 and 70)

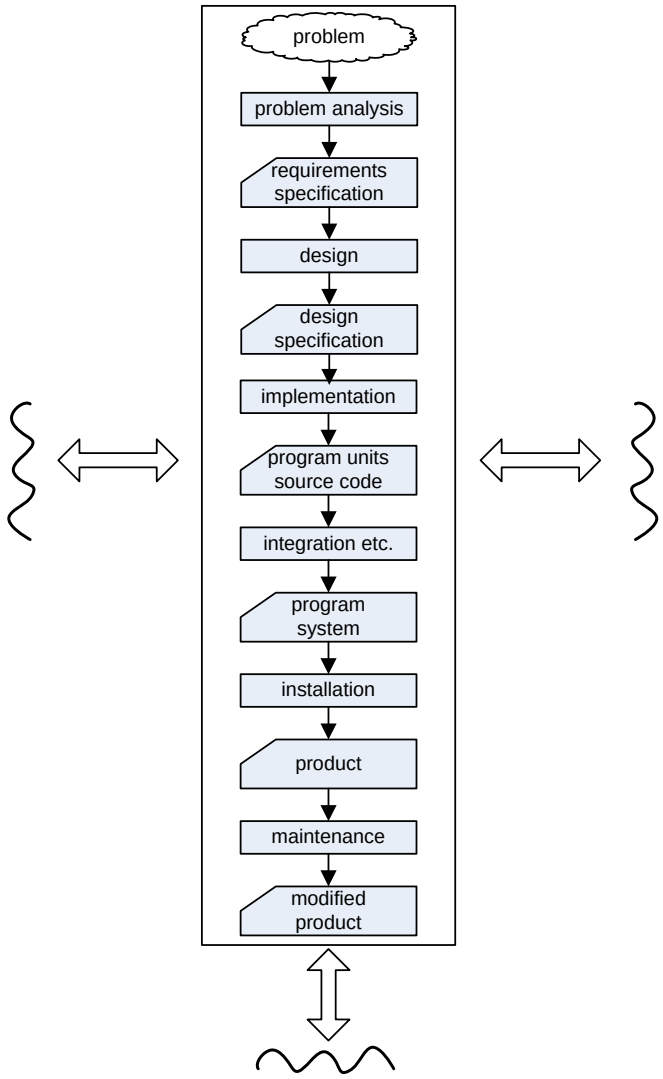
  - » F.L. Bauer: “SE means to produce software economically, which runs efficiently and reliably on real computers”
  - » German Computer Science Society: “SE is the discipline of Computer Science which deals with supply and systematic use of methods and tools for the development or use of application software”

- Characterization
  - » Development of large software is qualitatively different from small software
  - » Problems only
    - If many developers cooperate
    - Different versions of program system exist
  - » Cooperation in the project team
    - Planning and distribution of labor
    - Tasks / solutions, interaction / integration
    - Quality assurance
    - Documentation

  } Complexity: modeling problem on different levels

Life Cycle Models (Coarse-grained)

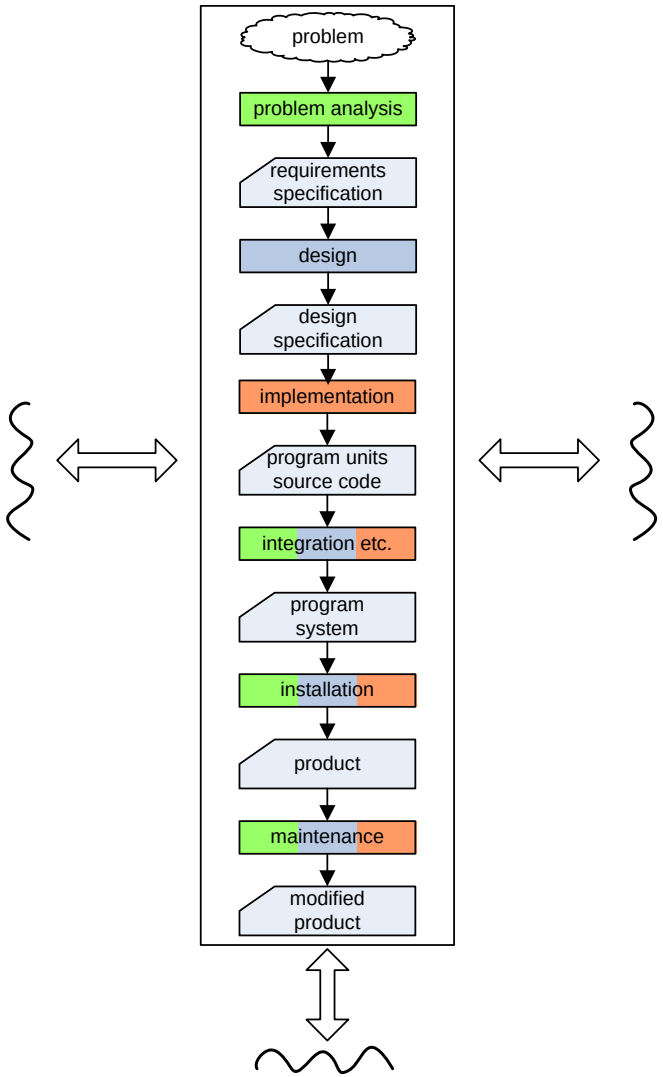
- Specific form: phase model (waterfall model)
  - » Activities: phases
  - » Results: documents
- Reason for phases:
  - » Complexity reduction
  - » Quality assurance
- Definition: „document“
  - » Result of a subtask in development process



Activity Areas

- Phase model: time relation („is result of“, „is necessary input for“)
- Activity Areas
  - » Modelling on the same level
  - » Collecting similar activities occurring at different points of time scale
  - » Prerequisite for discussing the relations between similar and different tasks
  - » No distinction between construction and modification
- Three working areas
  - » Defining / modifying outside behaviour / requirements: requirements engineering
  - » Defining / modifying essential structure of system according to requirements: architecture modelling (design, prog. in the large)
  - » Defining / modifying detailed realization: implementation (programming in the small)

Where do Corresponding Activities Occur?

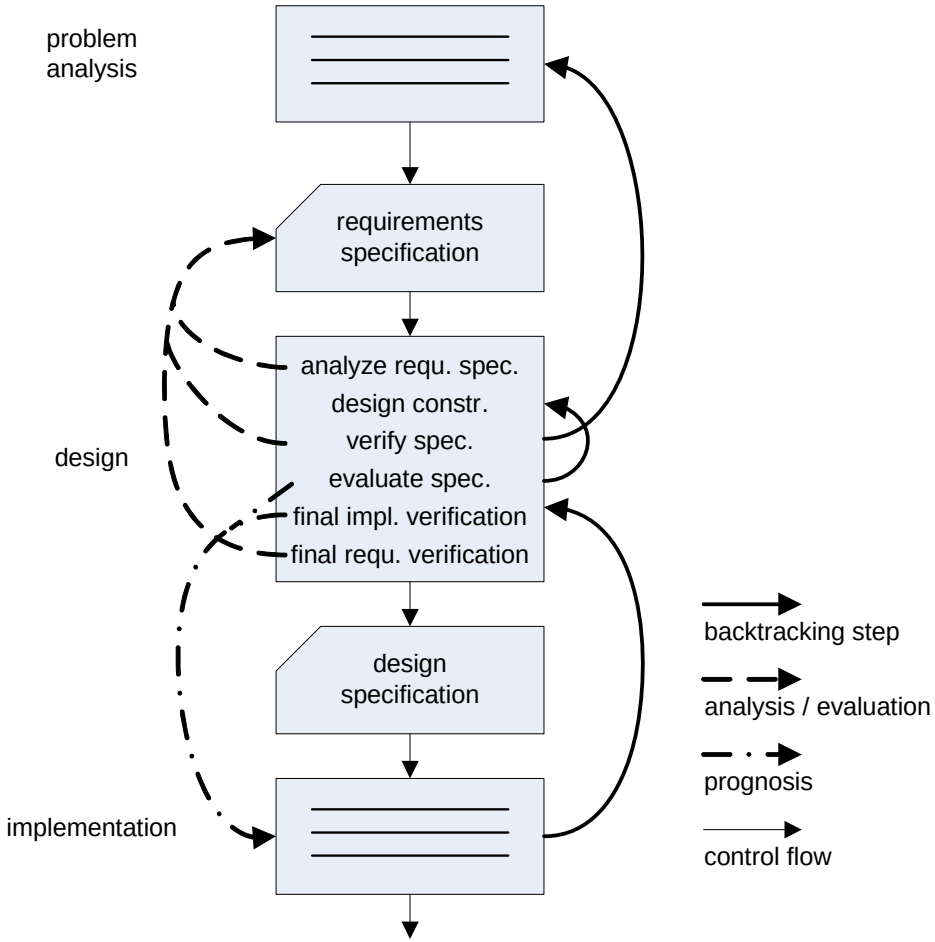


- Requirements Engineering
- Programming in the large
- Programming in the small

Looking on Design (or other Phase) More Closely

In any phase:

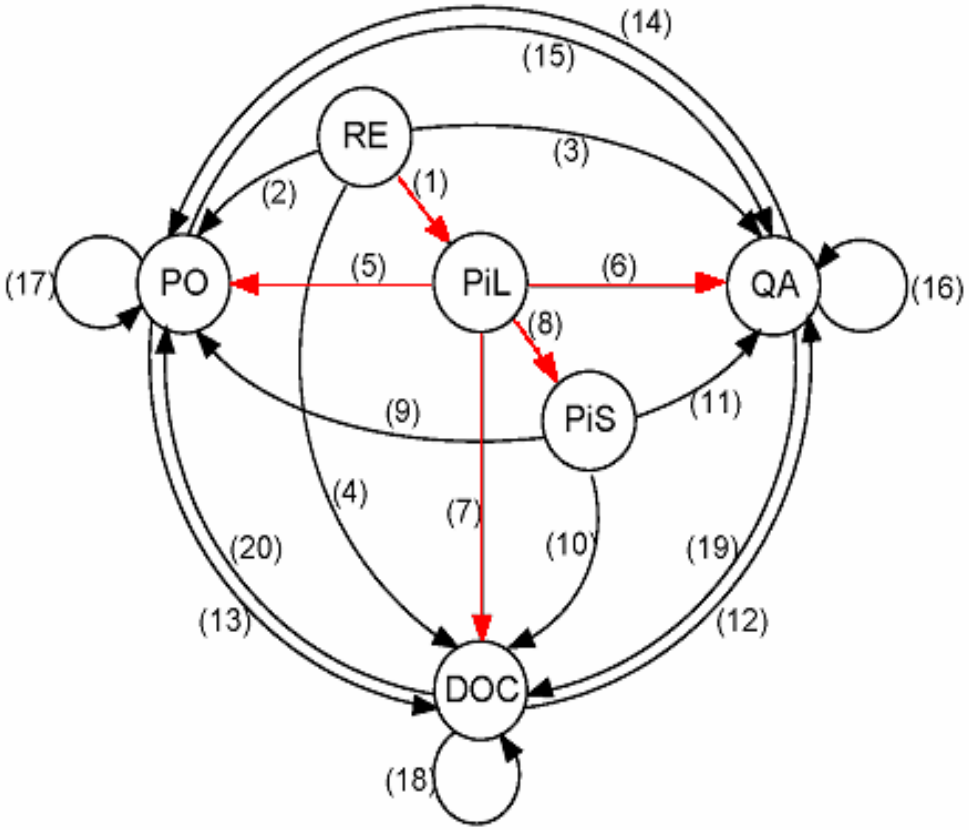
- Analysis for verification
- Stepwise construction, internal analysis, analysis back and forward
- Final check back and forward



Programming in the Large / Design / Architecture Modelling

- Analyze requirements specification under design aspects
- Stepwise design using modules and subsystems
- Stepwise verification of resulting design specification fragments: internal, against requirements specification, feasibility for implementation, integration, and maintenance
- Final verification against requirements specification
- Final evaluation against realizability
- Formulation of design specification in a programming language: Coding in the Large
- Integration and functional check of modules, subsystems, overall system corresponding to the software architecture
- Performance test of modules, subsystems, and overall system …
- Installation of overall system from components …
- Change of architecture (esp. during maintenance) by repeating the above steps
- …

Activity Area Graph Model



working areas and dependencies

RE Requirements Engineering

PiL Programming in the Large

PiS Programming in the Small

PO Project Organization

QA Quality Assurance

Explanation of Some Relations

(1) Architecture documents have to be consistent with the requirements specification documents

(5) Project organization documents for cost planning (white box estimation) but especially for management and supervision are highly dependent on the architecture

(6) The architecture is reviewed or checked giving rise to dependency relations between architecture documents and review protocols, check lists, etc.

(7) The architecture determines most of the technical documentation not only from its structure but also from its components

(8) The modules to be implemented are determined by the decisions made during architecture modelling. Only the implementation is to be worked out

To Be Formulated:

- RE: to build the “right” system
- Architecture: long-term properties
  maintainability
  adaptability
  portability
  …
  reuse
- Management: costs
  risks