

## On the Impact of Software Architecture for Mechatronic Systems

Prof. Dr.-Ing. Dagmar Meyer, Department of Electrical Engineering



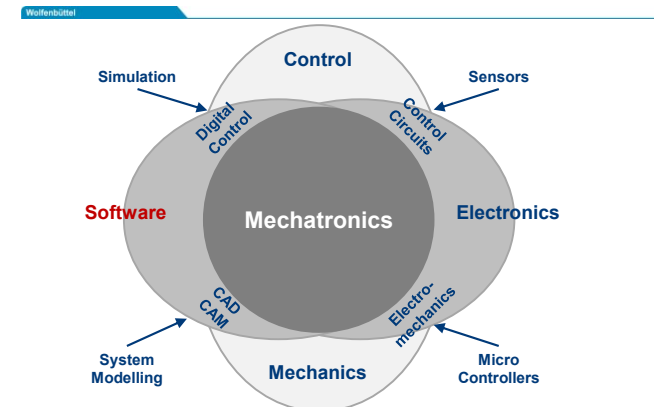
## 10 Years ago ...

“We are all proud of the machine tools which are highly respected in far-flung areas all over the world. German automobiles are highly prestigious products. But one has to know that for example 70 % of the value of the Swabian machine tool consists of microprocessors and software. The newest generations of cars, whether from Wolfsburg, Munich or Sindelfingen, are essentially small data processing centers which are able to drive. Even the most confirmed computer hater will soon have operated five of them before leaving the bathroom.”

*Erwin Staudt, head of the management of IBM Germany, 2002*

## Abstract

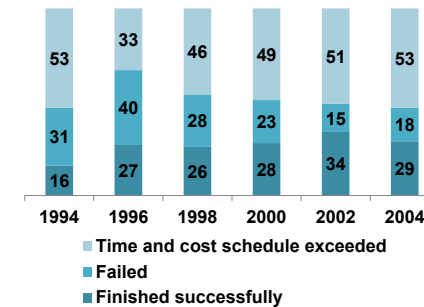
- Motivation
- Software Projects in Practice
- Software Architecture – Definition and Objectives
- Principles, Styles and Patterns
- Conclusions



## Software seizes Mechatronic Systems Example: Electric Servo Drive System

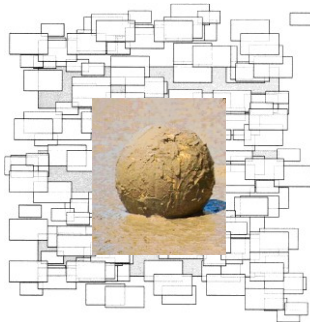
- First generation
  - Analogue current, speed and position control loops
  - Tuning of controller parameters by means of potentiometers
- State of the art
  - Digital control loops
  - Automatic parameter setup
  - Digital I/O
  - Field bus interfaces
  - PLC-like programming
  - Special functions like "flying saw", "electronic shaft" etc.

## Success of Industrial Software Projects



All numbers in percent. Source: Standish Group, Chaos Report 2004

## The "Big Ball of Mud" – inevitable or not?



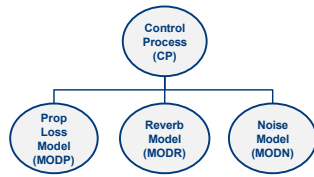
- Many software systems end up in a "big ball of mud"
- Possible reasons
  - Underestimation of complexity
  - Lack of knowledge about methods of software architecture
  - No software architect
  - Many cooks spoil the broth

## What is Software Architecture?

"Architecture is defined by the recommended practice as the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution."

ANSI/IEEE-Standard 1471-2000  
„Recommended Practice for Architectural Description  
of Software-Intensive Systems“

## Descriptions of Software Architecture need a formal Language like UML!



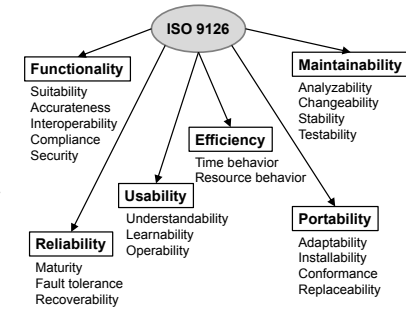
Part of the system description of an acoustic underwater simulation

Bass, L.; Clements, P.; Kazman, R.: Software Architecture in Practice. Addison-Wesley, Boston, 2000.

- What is the meaning of the lines in the diagram?
- What is the intention of ordering the components in the diagram in the way shown?
- Is CP some kind of master component which delegates certain tasks to the other components?
- In which way does the architecture behave at runtime?

## ISO 9126 Quality Model

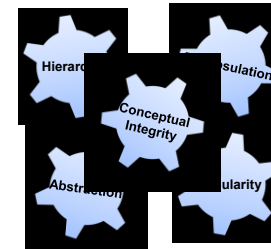
A good software architecture enables a system to meet its requirements concerning quality, lifecycle and behavior.



## Side Effects of the Development Process of a Software Architecture

- Efficient development
  - Software architecture = base for project management
- Conservation of quintessential knowledge
  - Architecture reuse
- Minimization of risks
  - by early consideration of architecture drivers like quality and management requirements
- Establishment of understanding
  - Architecture = communication medium among stakeholders

## Fundamental Design Principles

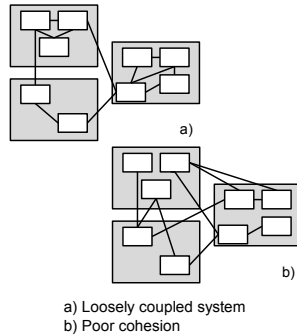


I am more convinced than ever. Conceptual integrity is central to product quality. Having a system architect is the most important single step toward conceptual integrity... After teaching a software engineering laboratory more than 20 times, I came to insist that student teams as small as four people choose a manager, and a separate architect.

Brooks, Frederick. The Mythical Man-Month (20th Anniversary Edition), 1995).

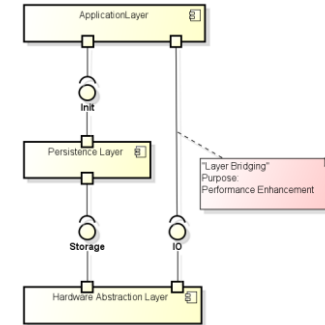
### Important Architecture Principles

- Increase cohesion and reduce coupling
- Separation of concerns
- Information hiding
- Separation of interfaces
- Segregation of interfaces



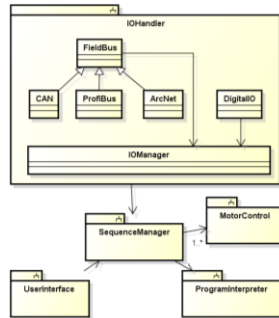
### Architectural Styles

- Avoid reinventing the wheel
- → architecture reuse
- Commonly known architectural styles
  - Layered architecture
  - Implicit invocation
  - Object-oriented style
  - Pipes-and-filters



Layered architecture with layer bridging

### Object-oriented architectural style

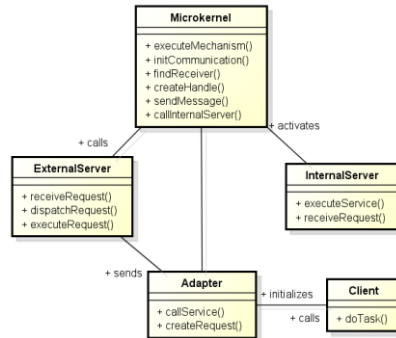


- The object-oriented architectural style promotes separation of concerns
- Architecture is independent of the implementation language → object-oriented style does not necessarily require an object-oriented programming language

### Architectural Patterns

- Architecture reuse improves the quality of the software architecture
- Architectural patterns are widely accepted as recurring solutions that solve problems at the architectural design level
- Patterns address subjects concerning the internal infrastructure of the software system under design
- Independent of the implementation language and the system platform
- Provide a common vocabulary in order to facilitate communication
- Best known source of patterns: Book series "Pattern Oriented Software Architecture" (POSA)

## Example: Microkernel Pattern



Prof. Dr.-Ing. Dagmar Meyer, Department of Electrical Engineering, 2011-09-15

17

## Conclusions

- Due to the complexity of mechatronic systems, a reliable and flexible software architecture is essential
- In practice, the system software of mechatronic systems often evolves incrementally and accidentally without any serious architecture design process
- As the core functionality of mechatronic systems is more and more provided by software, mechatronic engineers must be able to communicate with the software architect of the system under design
- This requires at least basic knowledge of software architecture → knowledge of one or more programming languages is not sufficient for mechatronic engineers.

Prof. Dr.-Ing. Dagmar Meyer, Department of Electrical Engineering, 2011-09-15

18