Embedded Systems Software Engineering

Jarosław Kuchta

Design in software life cycle
Classic lifecycle

Planning → Analysis → Design → Implementation → Testing → Maintenance
Design after analysis

Planning

Requirements extraction

Analysis

Static analysis
Functional analysis
Behavioral analysis
Dynamic analysis

System design

System architecture design
Application logic design
User interface design
Data structure design

Detailed design

SRS

modeling

Requirements extraction
Analytic models and aspects

- Class model (static analysis)
- Use case model (functional analysis)
- Collaboration model (functional analysis)
- Activity model (functional analysis)
- State transition model (behavioral analysis)
- Sequence model (dynamic analysis)
Analysis vs. Design

• **Analysis**
  – Determines *what* should be done
  – Modeling real world system

• **Design**
  – Determines *how* it should be done
  – Modeling software and hardware system
Design aspects

• **System architecture design**
  – Overall system infrastructure, technology choose, main solution decisions, system components specification

• **Application logic design**
  – Software classes specification

• **User interface design**
  – User – system interaction, interface structure and navigation

• **Data structure design**
  – Ways to store data in a file system or a data base
  – Ways to interchange data between components and with other systems
Design artifacts

- System architecture design
- Application logic design
- User interface design
- Data structure design
Design level models and diagrams

- Class model
- Use case model
- Collaboration model
- Activity model
- State transition model
- Sequence model

- Data Flow Diagram
- Windows Navigation Diagram
- Entity-Relationship Diagram
- Components Diagram
- Deployment Diagram
Bibliography

• Roger S. Pressman: Software Engineering. A Practitioner's Approach (book, PDF)