

Project

Grundlagen des Software Engineering Fundamentals of Software Engineering

Prof. Dr. Dr. h.c. Dieter Rombach

SS 2015

Organizer

Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down



Prof. Dieter Rombach
rombach@informatik.uni-kl.de
32-423



Alexander Klaus
Alexander.Klaus@iese.fraunhofer.de
32-417



Christian Wolschke
wolschke@cs.uni-kl.de
32-419



Peter Zeller
p_zeller@cs.uni-kl.de
34-407



Max Steiner
steiner@cs.uni-kl.de
32-427



Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down

Apply engineering methods and techniques for the **systematic development of software-intensive systems**



Requirements
Engineering



Architectural design



Component
Engineering



Testing

Project Planning

Introduction

Demo LogiFlash

Reengineering+ Porting

Requirements

Architecture

Component Engineering

Testing

Project Touch Down

- You will work in **teams of up to 5 students**
- You will only be able to finish your tasks if you **continuously work** on them
 - 8 ECTS Points -> 240 hours, 13 weeks -> ~ 18,5 hours/week
- Each phase is completed by a **testate**
 - Check if team achieved expected results
 - Check if each team member significantly contributed to the results
 - Check if you are able to communicate your work

Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

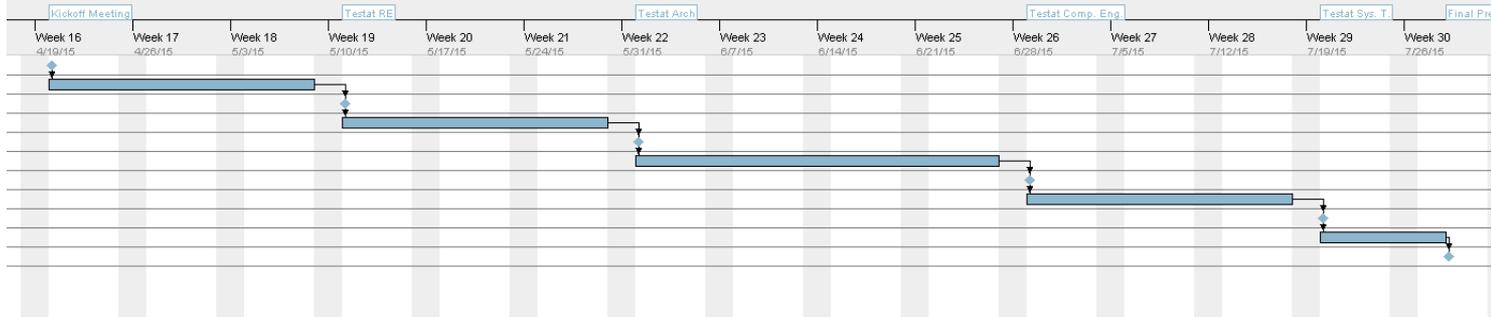
Architecture

Component
Engineering

Testing

Project Touch
Down

Schedule



Name	Begin date	End date
● Kickoff Meeting	20.04.	20.04.
● Requirements Engineering	20.04.	08.05.
● Testat RE	11.05.	11.05.
● Architecture	11.05.	29.05.
● Testat Arch	01.06.	01.06.
● Component Engineering	01.06.	26.06.
● Testat Comp. Eng.	29.06.	29.06.
● System Test	29.06.	17.07.
● Testat Sys. T.	20.07.	20.07.
● Final Rework + Lessons L...	20.07.	28.07.
● Final Presentation	29.07.	29.07.

Organizational Issues

- Project environment is provided
 - Technical support: Christian Wolschke, Thomas Schneider
 - Group Mailing Lists
 - Subversion Repo (with external access)



Thomas Schneider
tschneid@cs.uni-kl.de
32-418

Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down

LogiFlash

Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down

- LogiFlash is an **educational logic simulator**
 - **Complete** logic simulator (gates, flip-flops, registers,...)
 - Automatic **task evaluation**
 - Interfacing to **e-learning environments** via SCORM
- Platform/Language: Adobe Flash, ActionScript 2.0
 - **But:** Flash might become extinct
 - **Also:** Not a good basis for an Open-Source project
- **Goal:** Shift to **Open-Source, Cross-Platform**

The customer

Markus Damm
damm@cs.uni-kl.de
32-424



Introduction

Original LogiFlash Viewer

Demo LogiFlash

Reengineering+
Porting

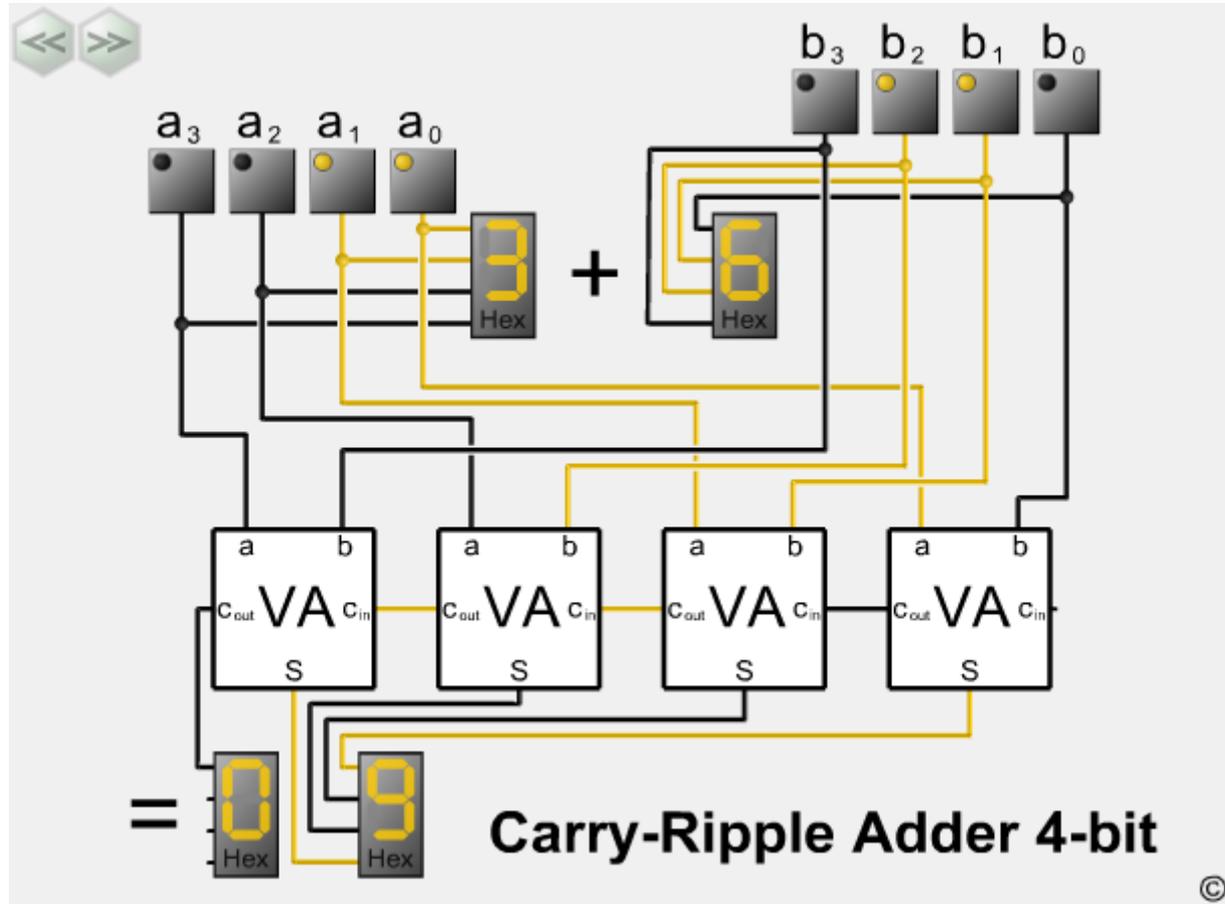
Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down



Introduction

A LogiFlash circuit controls an FSM animation

Demo LogiFlash

Reengineering+
Porting

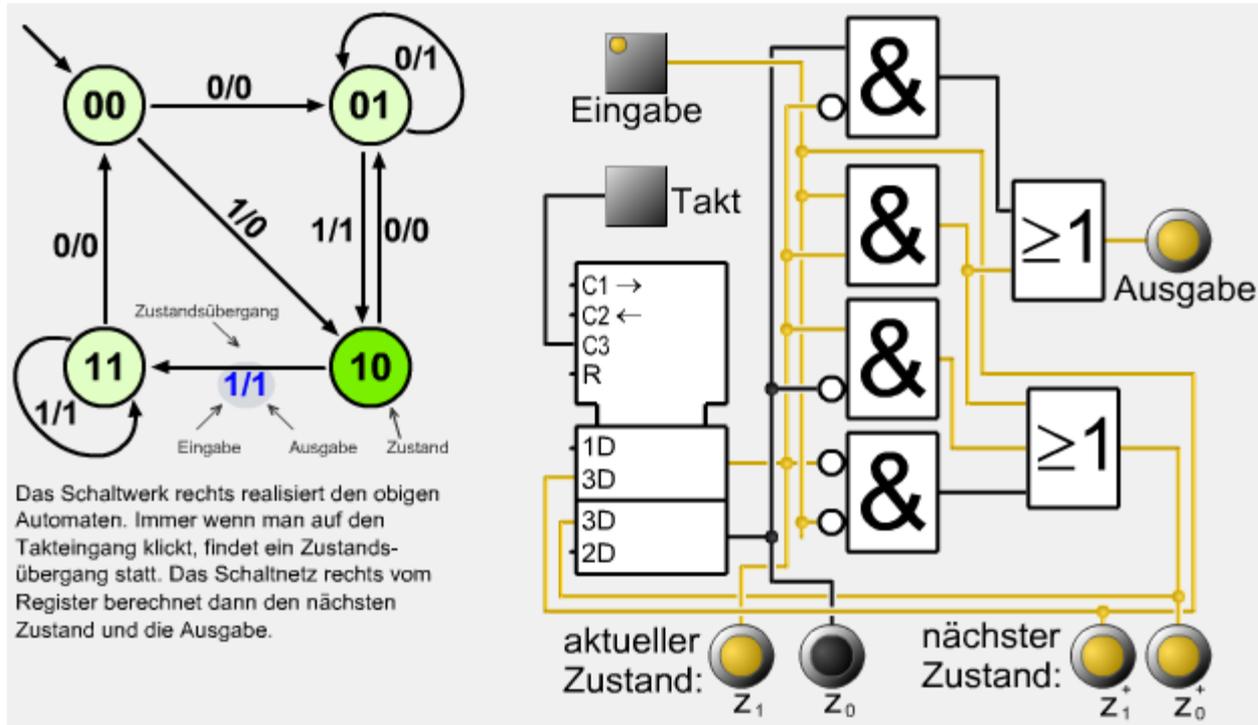
Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down



Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

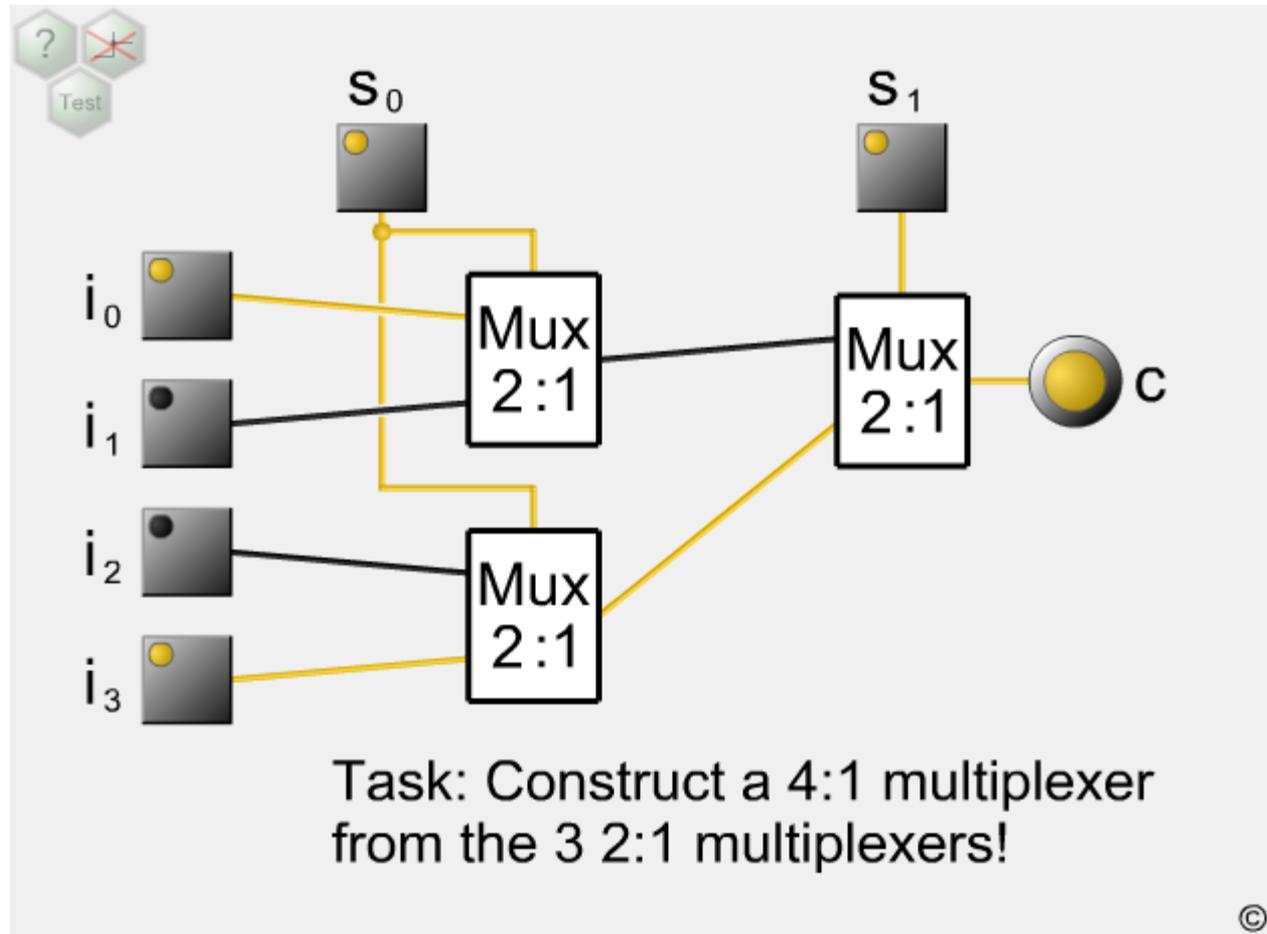
Architecture

Component
Engineering

Testing

Project Touch
Down

LogiFlash Task evaluation



Introduction

Original LogiFlash authoring tool

Demo LogiFlash

Reengineering+
Porting

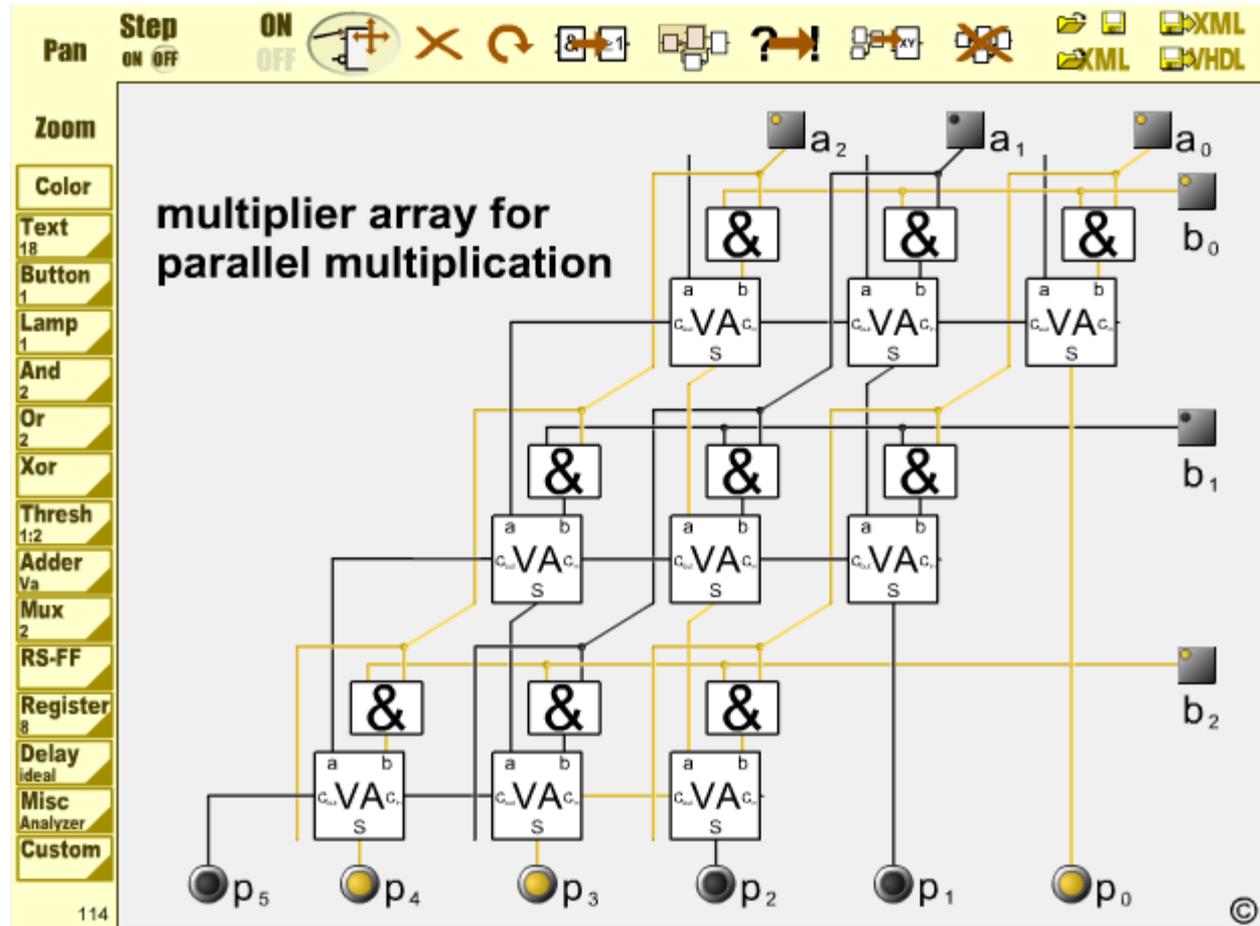
Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down



Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

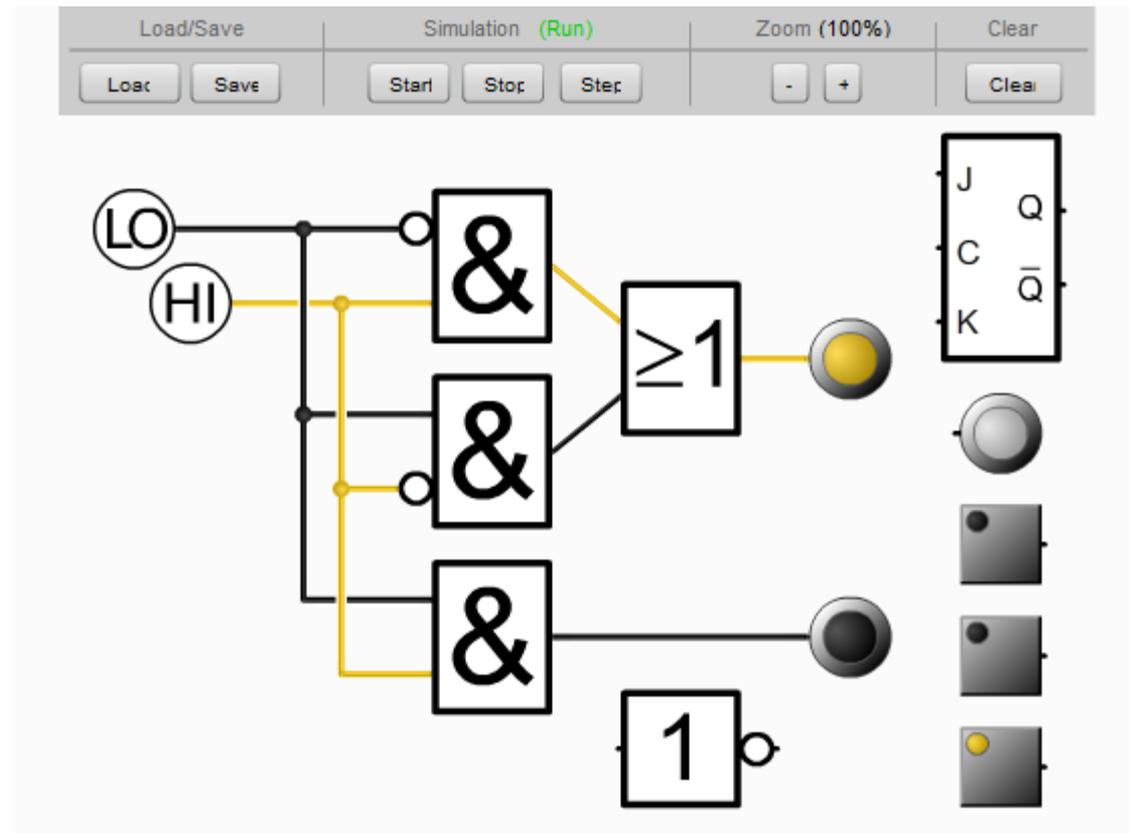
Architecture

Component
Engineering

Testing

Project Touch
Down

A first re-implementation approach with ActionScript 3.0



Goal: LogiFlash re-implementation using OpenFL + haXe

- Enables targets like Flash, HTML5, Android,...
- Implementation of a **LogiFlash-Kernel** based on the ActionScript 3.0 version
 - Some concepts will need revising
 - But certain algorithms (and the graphics) can be re-used
- Provide/Enable new features like:
 - Switch between 2-valued and 4-valued logic
 - Switch gate design DIN ↔ ANSI
 - Context menu help
 - Internationalization



Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down

Reengineering and Porting

Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down

Requirements Engineering

Alexander Klaus

Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down

Requirements Engineering

Introduction

Demo LogiFlash

Reengineering+
Porting

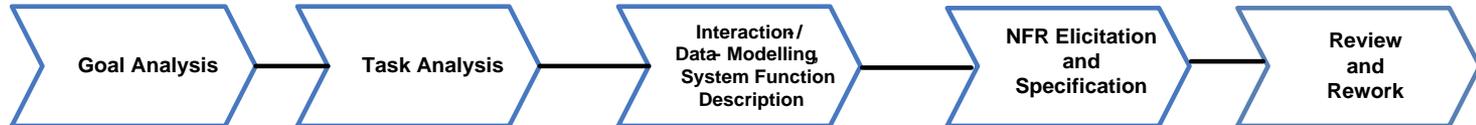
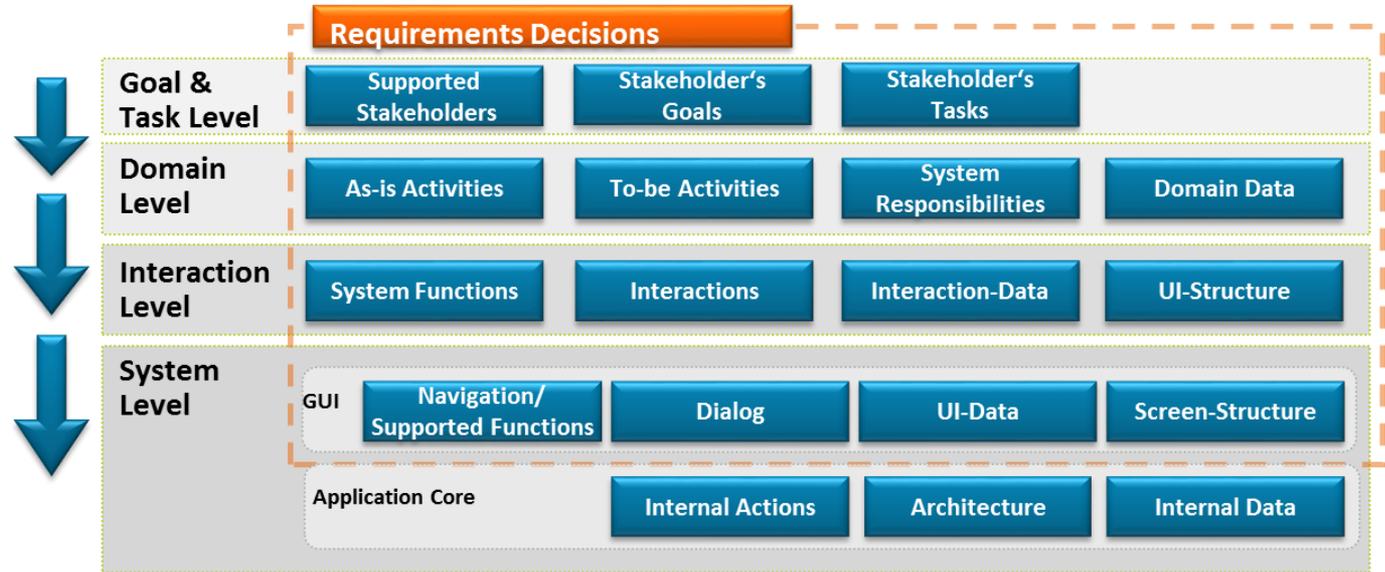
Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down



Requirements Engineering

Reengineering:

→ Some changes to the process

Requirements

1) Get familiar with the existing program:

Architecture

a) try to read old documents

Component Engineering

b) let stakeholders use the old program and watch

“think aloud”-technique

(what can be done? How is it done? ...)

Testing

2) Now start with the interviews (as usual)

Project Touch Down

Architectural Design

Christian Wolschke

Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down

Basic Tasks in Architectural Design

- **Input: Requirements**
- **Plan your solution System**
 - Plan Structure and Interfaces
 - Division of labour
 - Different views
 - Implement non functional requirements
 - Evaluate alternatives + prioritize modules
 - Trace your solution to requirements

Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down

Component Engineering

Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

**Component
Engineering**

Testing

Project Touch
Down

- **Input: Architecture + Priorities**
- **Implement Components**
 - Unit tests first
 - Readable, testable, simple code
 - Code documentation
 - Update & extend architecture (if necessary)
- **Technology: Haxe & OpenFL**
 - Prototype before implementation phase

Testing

Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

Component
Engineering

Testing

- **Test first approach → unit tests during component engineering phase**
- **Test phase:**
 - Integration testing
 - System testing
 - Acceptance testing
- Creating a useful test plans
- Conducting reasonable tests

Project Touch
Down

Project Touch Down

Introduction

Demo LogiFlash

Reengineering+
Porting

- **Final presentation in the end**

Requirements

- **Summarize knowledge and gained experience**

Architecture

Component
Engineering

Testing

**Project Touch
Down**

Introduction

Demo LogiFlash

Reengineering+
Porting

Requirements

Architecture

Component
Engineering

Testing

Project Touch
Down

Next steps

- **Register and get a group assigned**
- **Make appointment with Christian Wolschke and Thomas Schneider for introduction into laboratory**
- **Alexander Klaus will get in contact with you for requirements phase introduction**