

## L09 Architecture

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons  
“Attribution-ShareAlike 4.0 International” license.



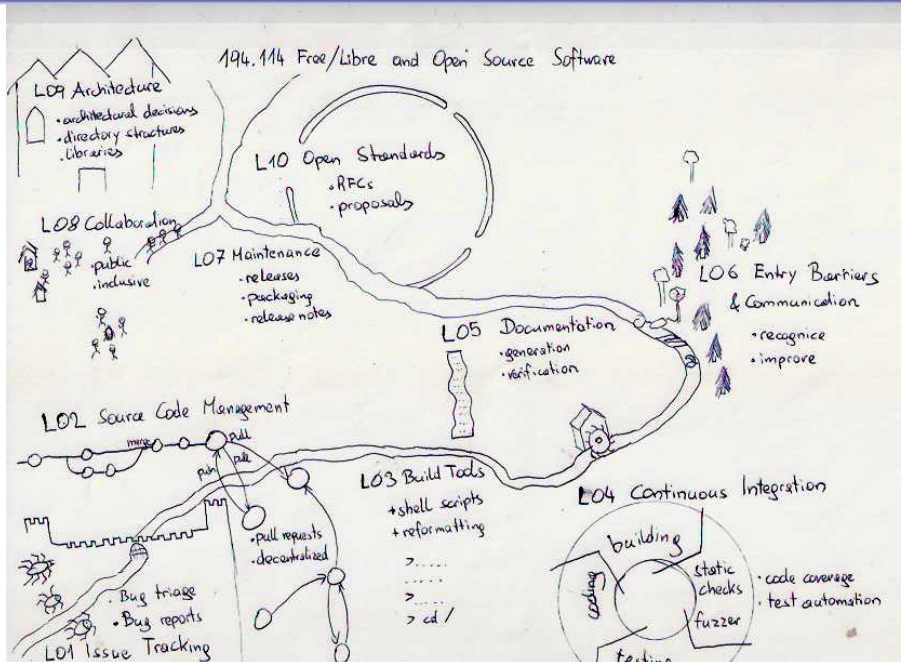
# Directory Structure

- 1 Directory Structure
- 2 Software Architecture
- 3 Architectural Decisions
- 4 Meeting
  - Recapitulation
  - Assignments
  - Preview

# Learning Outcomes

After successful completion of L09 Architecture students will be able to

- reproduce chosen FLOSS software architectures.



# Overviewability

- complementary to traceability  
(i.e. trace from requirements/documentation/issues to affected code and back)
- measurement how long a newcomer needs to find her/his way
- answers: “Where to add new functionality?”
- in FLOSS directory structure in repository essential (= physical view)

## Grouping of Files

- by programming language
- by topics (tests, src, doc)
- by modules/plugins

# Metadata of Files and Directories

- README.md
- file name endings
- LICENSES and .license
- scripts vs. src (executable bits)

# Cleanup

- top-level
- main folders
- file names
- automatic formatting (encodings and line endings)
- reduce dependencies between folders

But also stop worrying if needed by convention.



## Logical Views

- Documentation Generators (“Files” in Doxygen is physical view)
- Building Block View (may be identical to physical)
- Runtime View
- Deployment View

## L09 Architecture

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons  
“Attribution-ShareAlike 4.0 International” license.



# Software Architecture

- 1 Directory Structure
- 2 Software Architecture**
- 3 Architectural Decisions
- 4 Meeting
  - Recapitulation
  - Assignments
  - Preview

# Software Architecture

- architecture is a high-level description of the overall system
- use ready-made patterns and templates for architecture
- e.g., <http://arc42.org/>

# Arc42

- ① Introduction and Goals
- ② Constraints
- ③ Context and Scope
- ④ Solution Strategy
- ⑤ Building Block View
- ⑥ Runtime View
- ⑦ Deployment View
- ⑧ Crosscutting Concepts
- ⑨ Architectural Decisions [1]
- ⑩ Quality Requirements
- ⑪ Risks and Technical Debt
- ⑫ Glossary

## Example

Crosscutting concept “configuration settings”:

- are stored in configuration files
- in data structure KeySet
- modified by configuration management tool using KeySet

more about it in course “configuration management”

# Roles

In FLOSS usually nobody is project manager  
→ but everyone is software architect

# Goals

The most important tasks of software architects are

- to pursue the right goals
- to have good documentation (e.g. with arc42)
- to keep everything as simple as possible
- to communicate the architecture
- maintain community and quality



# Refactoring

- build what community needs at the moment
- change according to current needs
- avoid over-engineering, refactor to KISS

## L09 Architecture

Markus Raab

Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons  
“Attribution-ShareAlike 4.0 International” license.



# Architectural Decisions

- 1 Directory Structure
- 2 Software Architecture
- 3 Architectural Decisions**
- 4 Meeting
  - Recapitulation
  - Assignments
  - Preview

# Architectural Decisions

- describe decisions that lead to the architecture
- decisions are high-level configuration
- patterns/templates are useful [1], e.g.:

# Template

- ① problem
- ② constraints
- ③ assumptions
- ④ considered alternatives
- ⑤ decision
- ⑥ rationale
- ⑦ implications
- ⑧ related
- ⑨ notes

## Example: API Design

- future-proof
- hard to use it wrong vs. easy to use
- consistent concepts, e.g. for resources
- minimal vs. comfort

## Example: Libraries vs. Daemons

- both foster reuse of code
- daemon better if there is dynamic state
- but: daemon creates a single point of failure (KISS)

# Dangers

*Insanity in individuals is something rare – but in groups, parties, nations, and epochs, it is the rule. – Friedrich Nietzsche*

In groups you get confronted with the whole spectrum of psychology:

- Groupthink (conformity)
- Group polarization



## L09 Architecture

Markus Raab

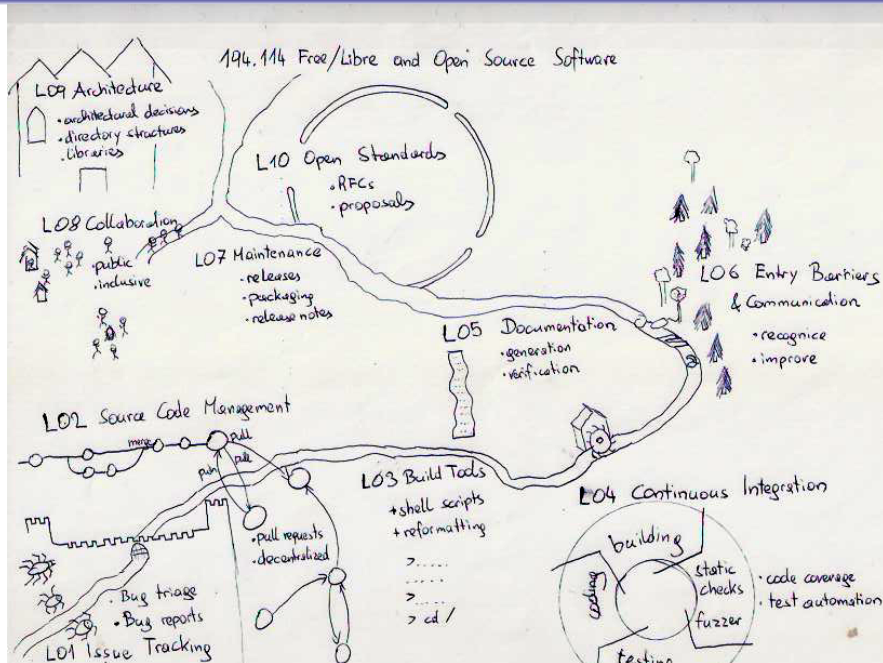
Institute of Information Systems Engineering, TU Wien

This work is licensed under a Creative Commons  
“Attribution-ShareAlike 4.0 International” license.



# Meeting

- 1 Directory Structure
- 2 Software Architecture
- 3 Architectural Decisions
- 4 **Meeting**
  - Recapitulation
  - Assignments
  - Preview



# Learning Outcomes

After successful completion of L09 Architecture students will be able to

- reproduce chosen FLOSS software architectures.

# Overviewability

What is traceability and what is overviewability?

# Overviewability

- complementary to traceability  
(i.e. trace from requirements/documentation/issues to affected code and back)
- measurement how long a newcomer needs to find her/his way
- answers: “Where to add new functionality?”
- in FLOSS directory structure in repository essential (= physical view)

## Discussion: How to Group Files?

- by programming language/projects (monorepo)
- by topics (tests, src, doc)
- by modules/plugins

### Task

How do you like files to be structured?

How do you keep order over time?

Task

Break.



# P09 /e/OS

Dennis Toth

# Arc42

- ① Introduction and Goals
- ② Constraints
- ③ Context and Scope
- ④ Solution Strategy
- ⑤ Building Block View
- ⑥ Runtime View
- ⑦ Deployment View
- ⑧ Crosscutting Concepts
- ⑨ Architectural Decisions [1]
- ⑩ Quality Requirements
- ⑪ Risks and Technical Debt
- ⑫ Glossary

# Goals

## Task

What are the most important tasks of software architects?

# Goals

The most important tasks of software architects are

- to pursue the right goals
- to have good documentation (e.g. with arc42)
- to keep everything as simple as possible
- to communicate the architecture
- maintain community and quality

# Architectural Decisions

## Task

What are architectural decisions?

Please give an example.

# Architectural Decisions

- describe decisions that lead to the architecture
- decisions are high-level configuration
- patterns/templates are useful [1], e.g.:

# Decision Process

Task

Break.



# P09 ESPHome

Jan de Boer

# Reading Text

## Task

Summarize what you found interesting.

Task

Break.

# P1 Reviews, T3 Corrections

## Task

Get all your PRs ready to merge and ask for them to be merged (Label)

# Feedback

Today home and teamwork will finish for this term.

- TISS/ TUWEL Feedback from 12.01.2023, 00:00 to 9.02.2023, 23:59
- filling out before 18.01 supports continuation of this lecture
- TUWEL: feedback for future improvements, TISS: for TU Wien



## Skip last meeting?

- two days earlier deadline (16.1.2023 23:59)
- full use of last meeting (18.1.2023 14:15 – 16:00)

### Task

Vote: Skip last meeting at 25.01.2023?

# L10 Open Standards

- [1] Neil B Harrison, Paris Avgeriou, and Uwe Zdun. Using patterns to capture architectural decisions. *Software, IEEE*, 24(4):38–45, 2007. ISSN 0740-7459. doi: 10.1109/MS.2007.124.