Learning Technique for FPSS

Lecture: Future-Proof Software-Systems

«Hierarchical Transformation»

Prof. Dr. Frank J. Furrer

FPS *Learning* Technique

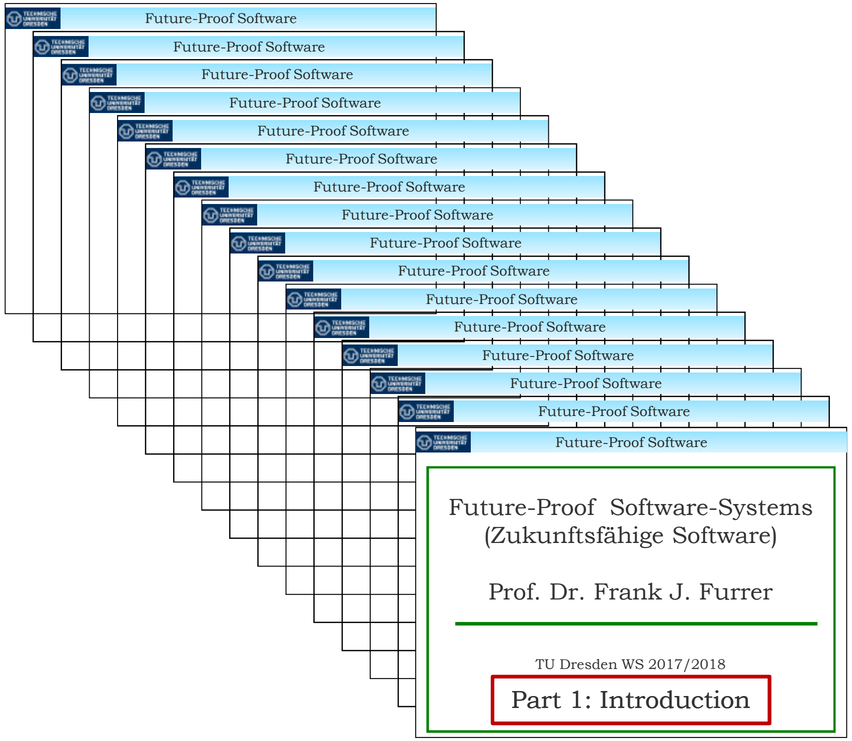
- The FPS lecture presents a lot of material
- The FPS lecture covers many topics

Which is an optimal learning technique?

https://talentcampushamburg.files.wordpress.com

Structure of the Lecture Material

= *Linear* Sequence of ppt-Slides



Future-Proof Software

Future-Proof Software-Systems
(Zukunftsfähige Software)

Prof. Dr. Frank J. Furrer

TU Dresden WS 2017/2018

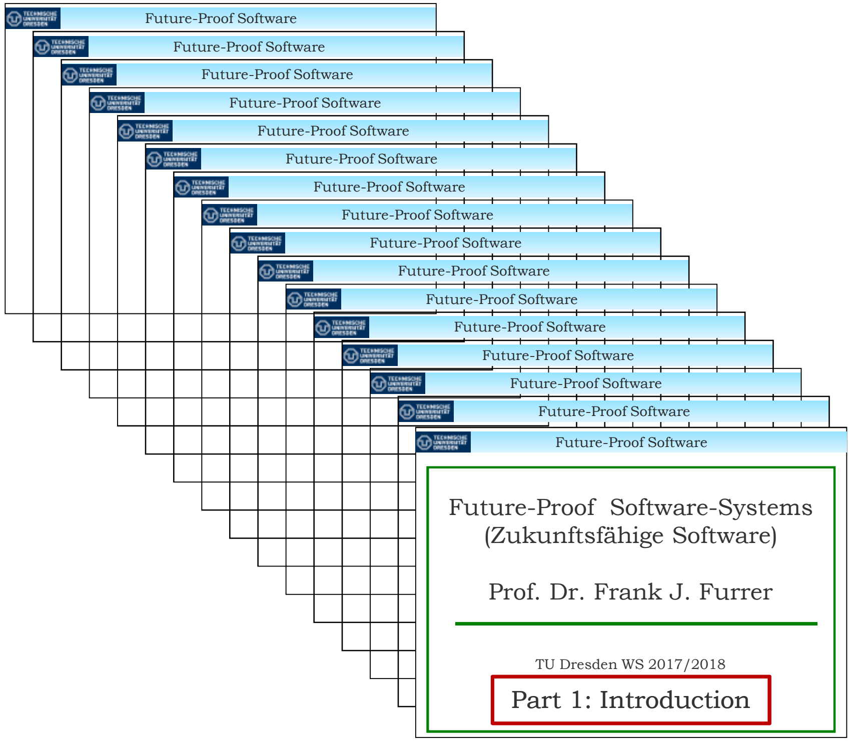
Part 1: Introduction

Fact 1:
Well suited for a presentation by a teacher

http://teaching.monster.com

Structure of the Lecture Material

= *Linear* Sequence of ppt-Slides



Future-Proof Software-Systems
(Zukunftsfähige Software)

Prof. Dr. Frank J. Furrer

TU Dresden WS 2017/2018

Part 1: Introduction

Fact 2:
NOT GOOD for individual learning

www.123rf.com

https://talentcampushamburg.files.wordpress.com

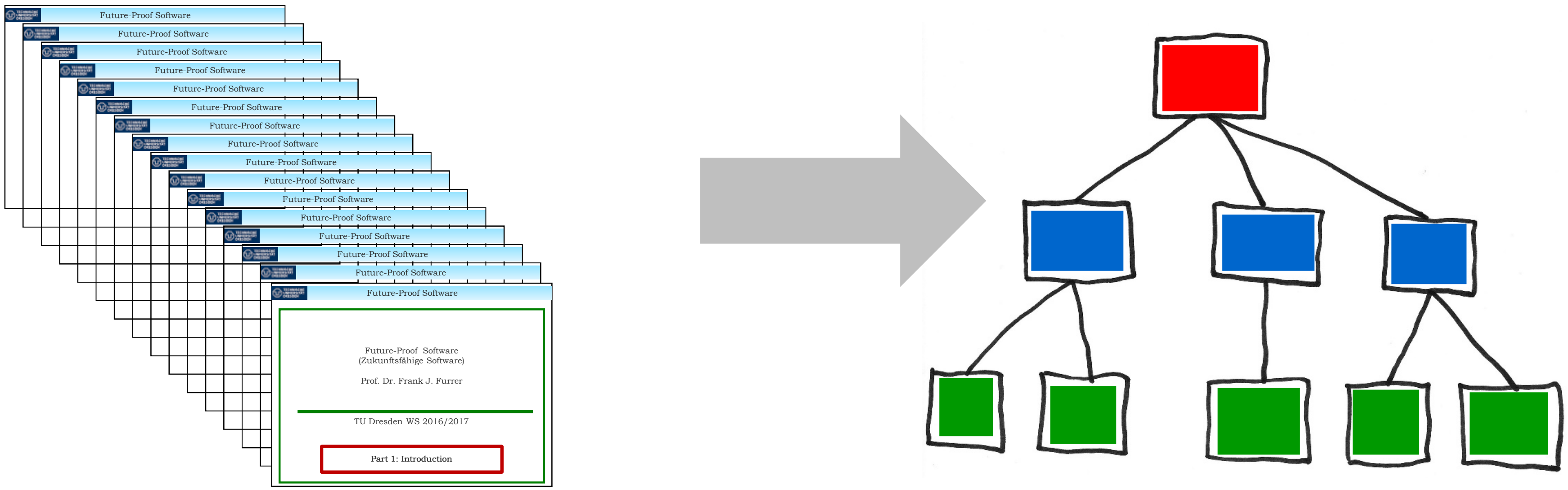
Optimal FPSS learning technique

<u>Step 1:</u>
Transformation from linear (slides) ⇒ hierarchical (structure)

<u>Step 2:</u>
Representation as «learning cards»

Step 1:

Transformation from linear (slides) ⇒ hierarchical (structure)



Future-Proof Software

Future-Proof Software
(Zukunftsfähige Software)

Prof. Dr. Frank J. Furrer

TU Dresden WS 2016/2017

Part 1: Introduction

a) Select a top-level concept

b) Identify the important sub-level concepts

c) Attach additional information

… repeat

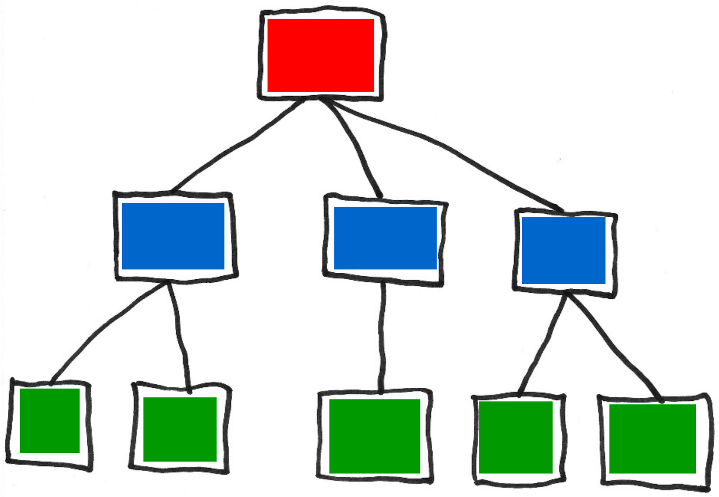
Step 1:

Transformation from linear (slides) ⇒ hierarchical (structure)

Top-level concepts

Important sub-level concepts

Additional information



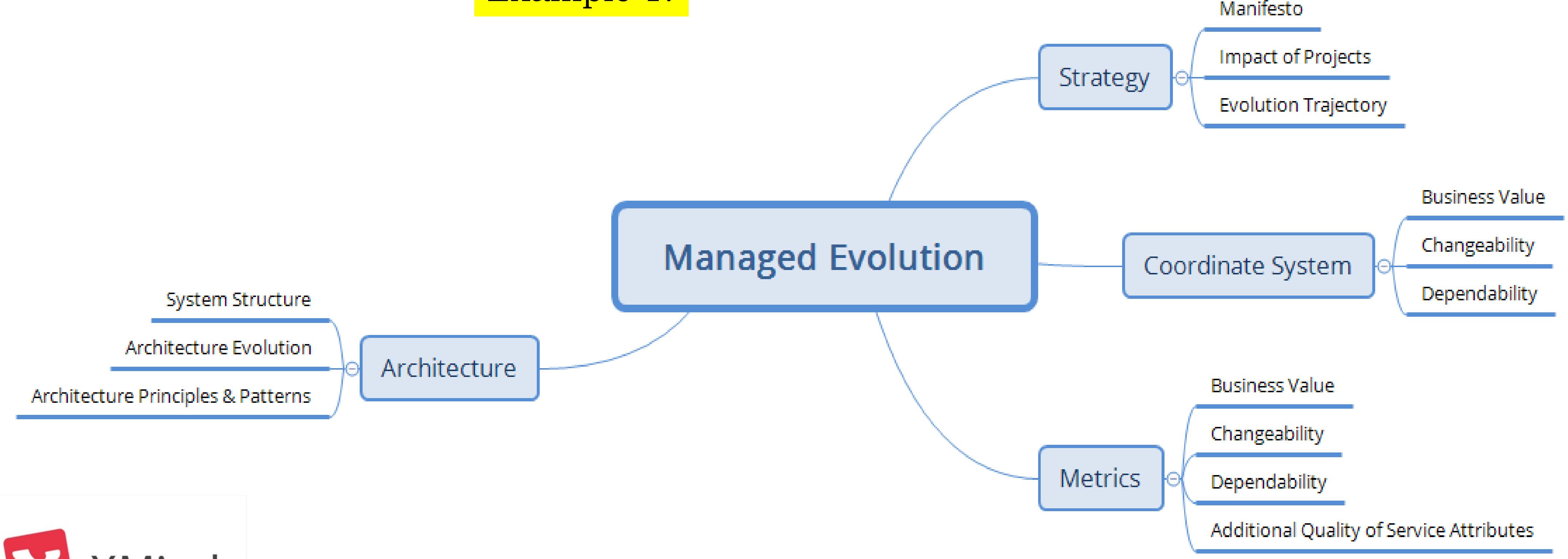
Organizing complex material into

hierarchical structures

is a natural, powerful learning technique («classification»)

Hierarchical Representation:
«Mind Map»

Example 1:

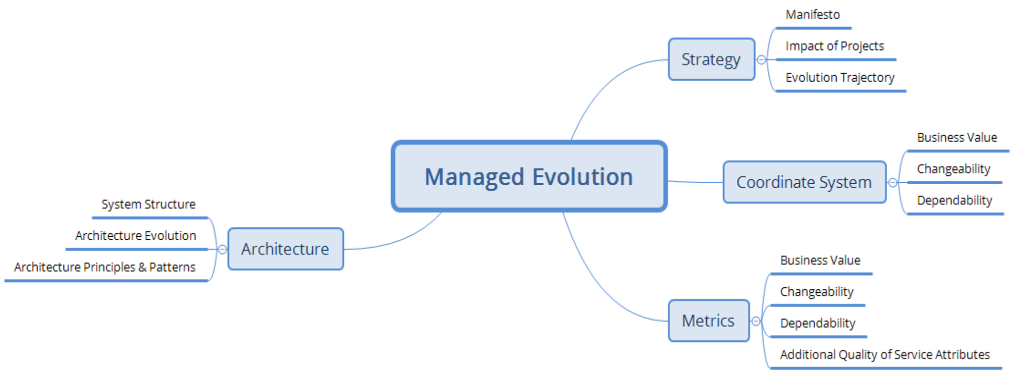


XMind

http://www.xmind.net

Step 2:
Representation as «learning cards»

Front: Concept Hierarchy

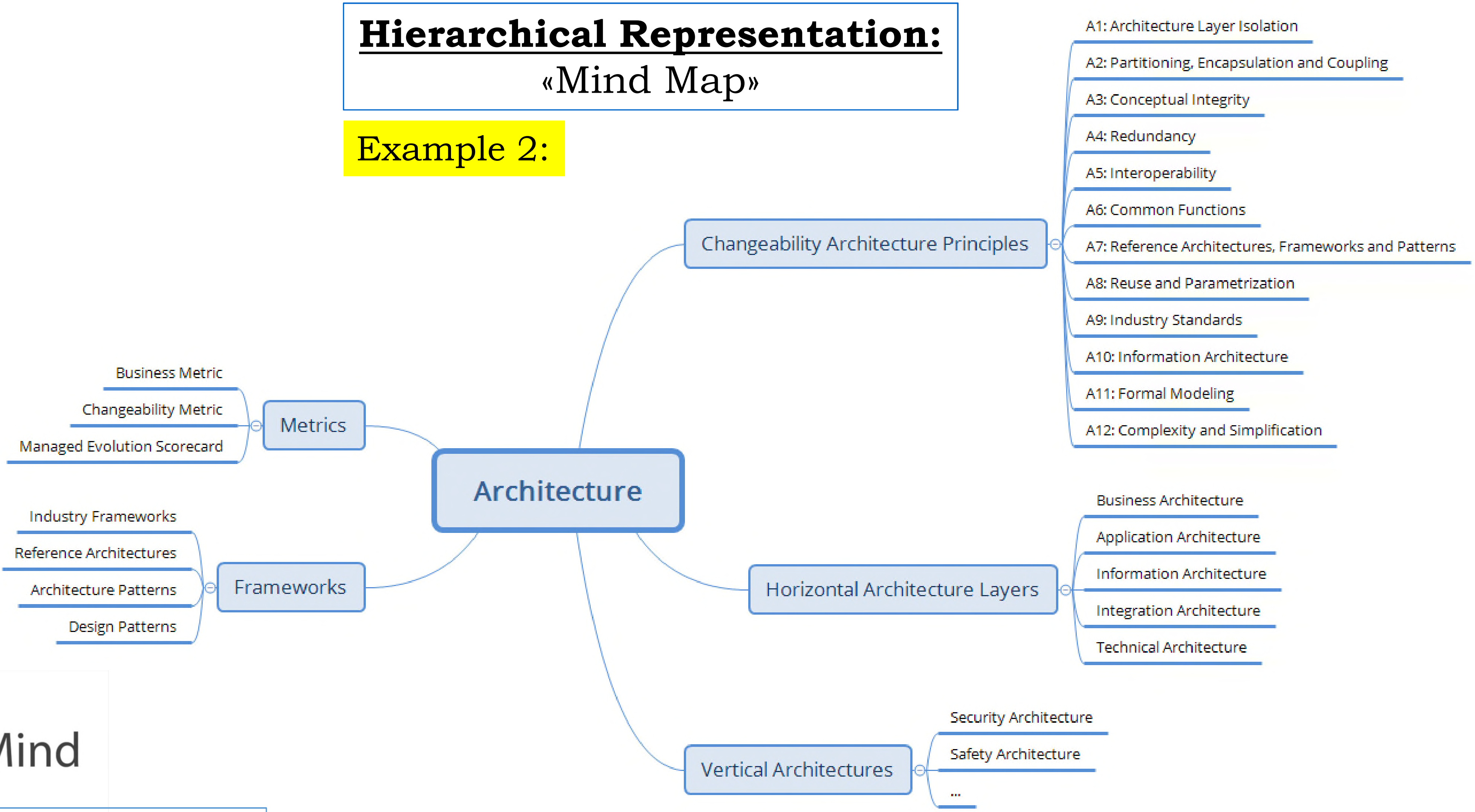


Back: Notes

- Negative: Technical debt, architecture erosion, business + market pressure
- Business value, changeability and dependability are continuously improved
- Managed Evolution coordinate system: ME evolution channel
- Tracking through metrics (BV, TtM, DevC, size)
- Dependability = survival / Changeability = adaptability to new requirements (TtM, DevC)
- Architecture = Key success factor
- Business ↔ IT interests/conflict

Hierarchical Representation: «Mind Map»

Example 2:



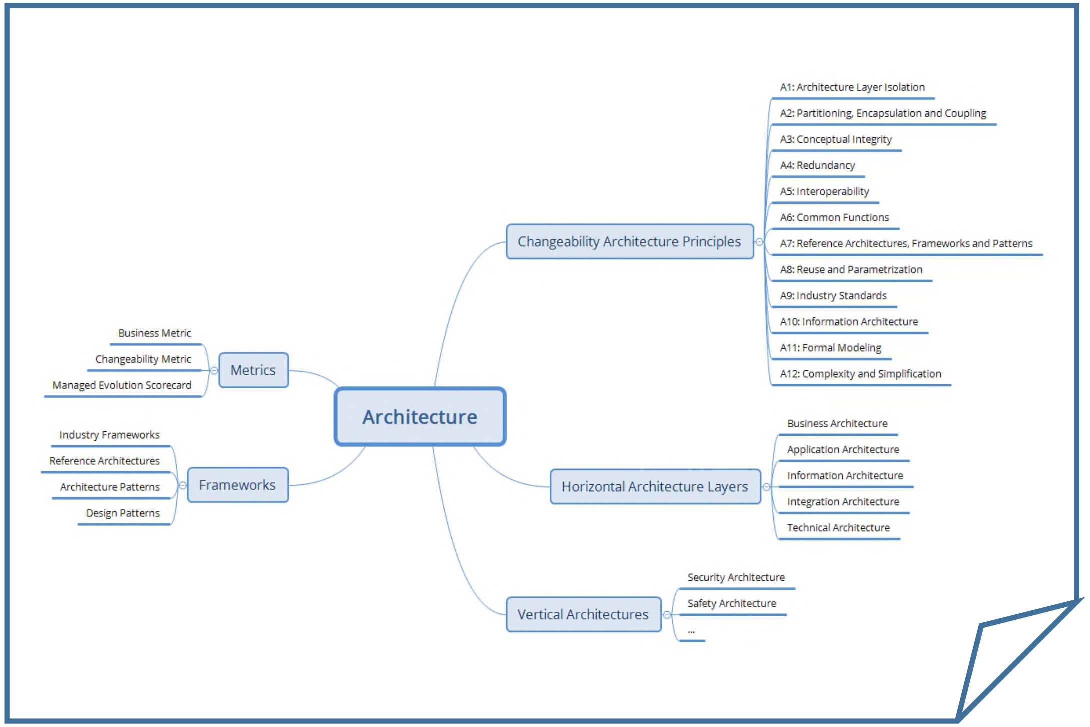
- Architecture
  - Changeability Architecture Principles
    - A1: Architecture Layer Isolation
    - A2: Partitioning, Encapsulation and Coupling
    - A3: Conceptual Integrity
    - A4: Redundancy
    - A5: Interoperability
    - A6: Common Functions
    - A7: Reference Architectures, Frameworks and Patterns
    - A8: Reuse and Parametrization
    - A9: Industry Standards
    - A10: Information Architecture
    - A11: Formal Modeling
    - A12: Complexity and Simplification
  - Metrics
    - Business Metric
    - Changeability Metric
    - Managed Evolution Scorecard
  - Frameworks
    - Industry Frameworks
    - Reference Architectures
    - Architecture Patterns
    - Design Patterns
  - Horizontal Architecture Layers
    - Business Architecture
    - Application Architecture
    - Information Architecture
    - Integration Architecture
    - Technical Architecture
  - Vertical Architectures
    - Security Architecture
    - Safety Architecture
    - ...

XMind

http://www.xmind.net

Step 2:
Representation as «learning cards»

Front: Concept Hierarchy

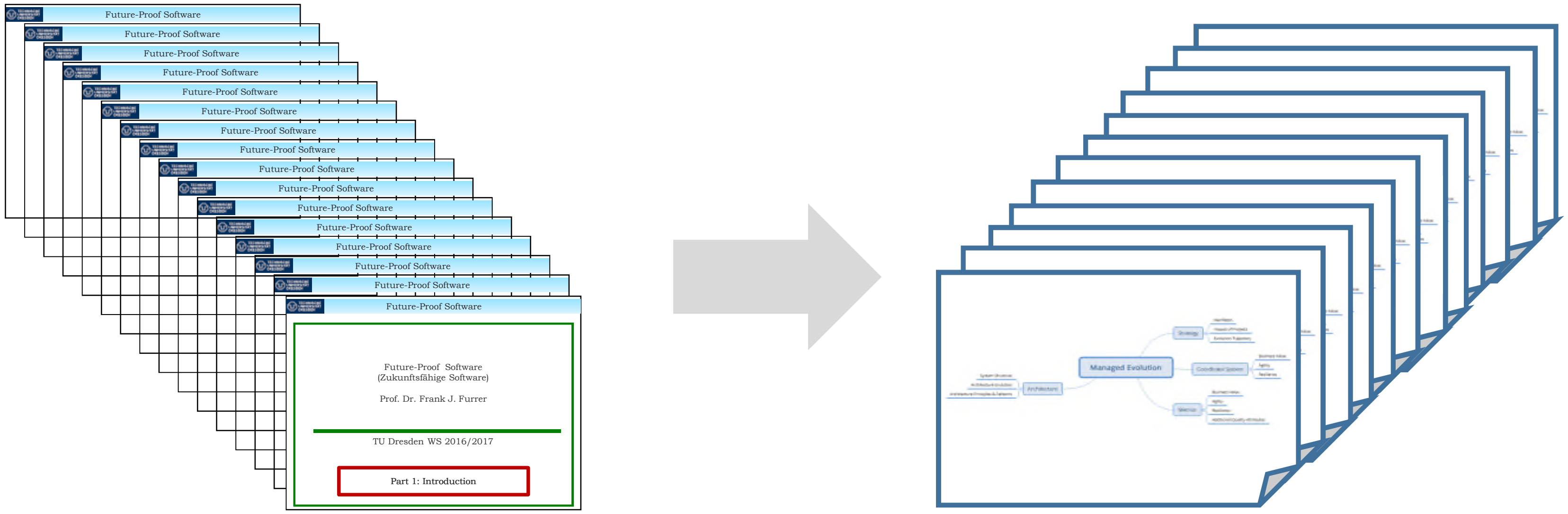


Back: Notes

- Architecture is the single most important key for future-proof software-systems
- Architecture is layered: 5 isolated layers are defined
- Architecture knowledge is contained in principles, patterns, frameworks and reference architectures
- Changeability is founded on 12 application architecture principles A1 … A12
- For the vertical architectures (security, safety, …) a great number of principles and patterns exist
- Architecture is not static: It needs continuous maintenance and good care

Slide Set

Learning Cards



Future-Proof Software
(Zukunftsfähige Software)

Prof. Dr. Frank J. Furrer

TU Dresden WS 2016/2017

Part 1: Introduction

Production = Knowledge organization & knowledge acquisition

Repetition = Knowledge amplification

Completeness = Sufficient time investment

http://www.decision-making-solutions.com

Hints:

- ➢ Produce the learning cards continuously (e.g. after each lecture)
- ➢ Look at the learning cards regularly & randomly
- ➢ Refine and add more notes when necessary
- ➢ **DO IT YOURSELF !!!**