Multimedia Didactics for Software Engineering

Ralf S. Engelschall

2023-07-31, Helmut-Balzert-Preis, Jury-Presentation

Software Engineering Academy

TECHNISCHE UNIVERSITÄT MÜNCHEN

.msg

About Author

- Dr. **Ralf S. Engelschall** (50)
- Science Context:
  - Computer Science Study at *TU München*
  - Computer Science Doctorate at *Uni Augsburg*
- Work Roles:
  - Director of *msg Research*
  - Director of *Software Engineering Academy*
  - Lecturer at *TU München*
- Education Activities:
  - *Software Engineering in der industriellen Praxis* (SEIP)
  - *Software Engineering Fundamentals* (SEF)
  - *Software Architecture Fundamentals* (SAF)
  - *IT-Architekten-Ausbildungsprogramm* (ITAAP)
  - *Trend Competence Program (TCP)*
  - *Container-Based Deployment (CBD)*

Software Engineering Academy

My Education Challenges

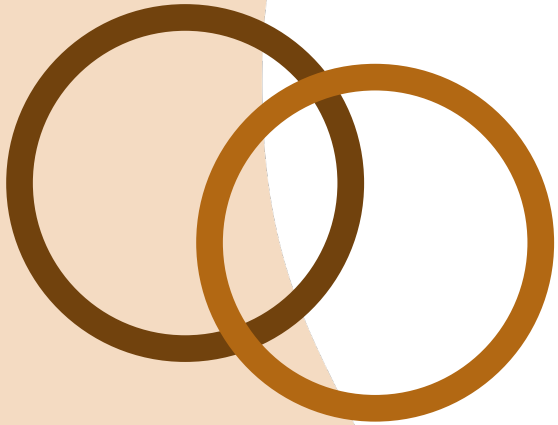
- Different Education Contexts
  university lectures, industry trainings,
  different durations/brandings, etc
- Unsexy and Complex Topic
  non-agile upfront thinking,
  many Software Engineering disciplines, etc.
- Generation-Z Low Attention Threshold
  social acceleration, information overflow,
  strong catchyness expectation, etc.
- Post-Pandemic New Work
  online-first, home-office-driven,
  reduced traveling (sustainability), etc.

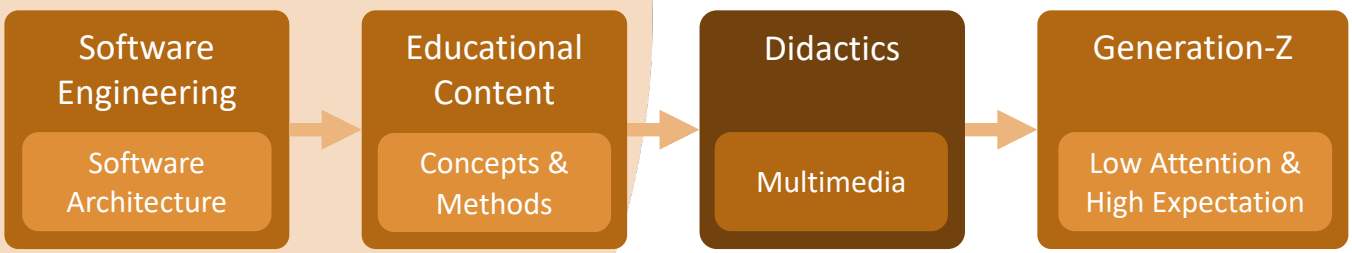
Software Engineering Academy

My Education Context

Software Engineering Academy

- Didactics:
  the approaches, methods and activities of systematic teaching.
- Multimedia:
  the form of communication that combines different content forms (text, images, video, audio).

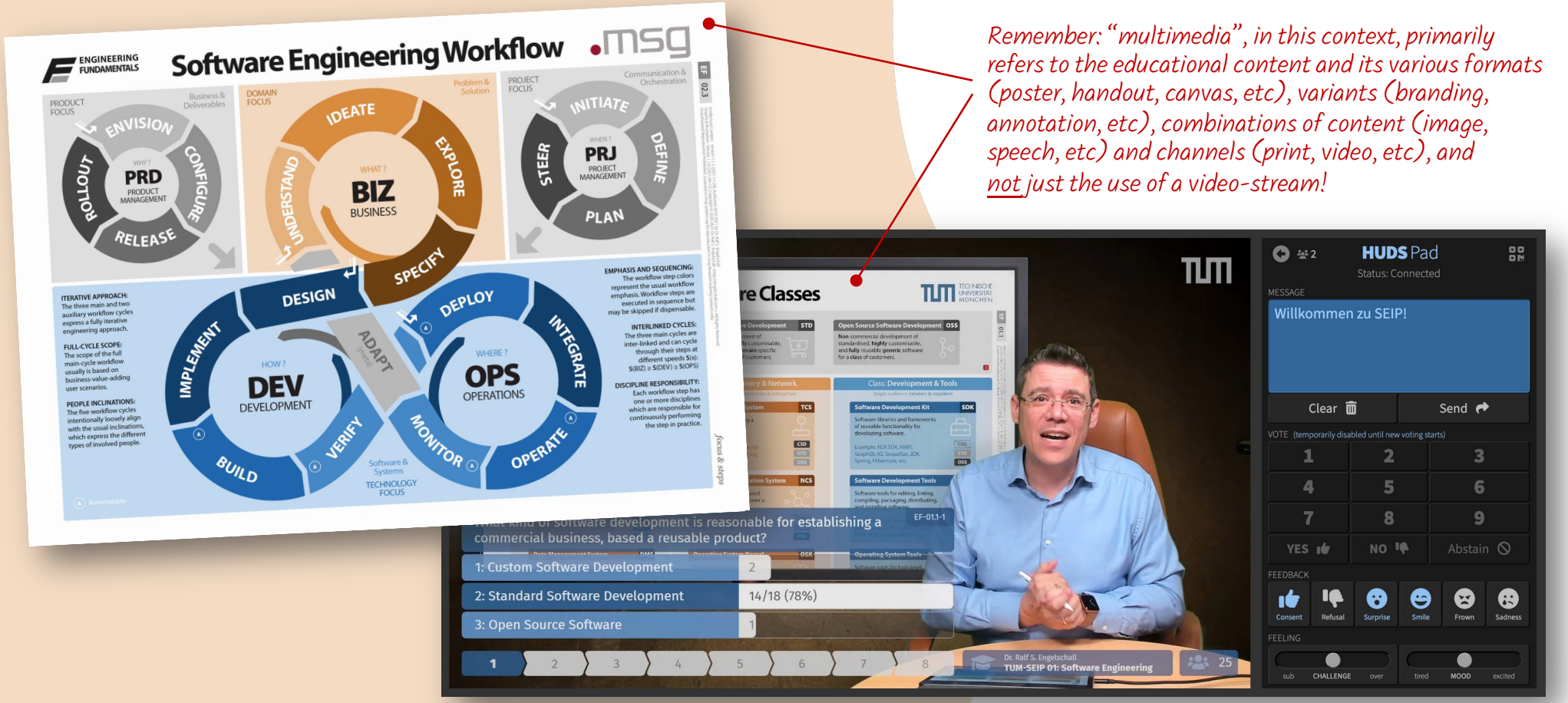




My Education Solution

Multimedia Didactics for Software Engineering

Software Engineering Academy

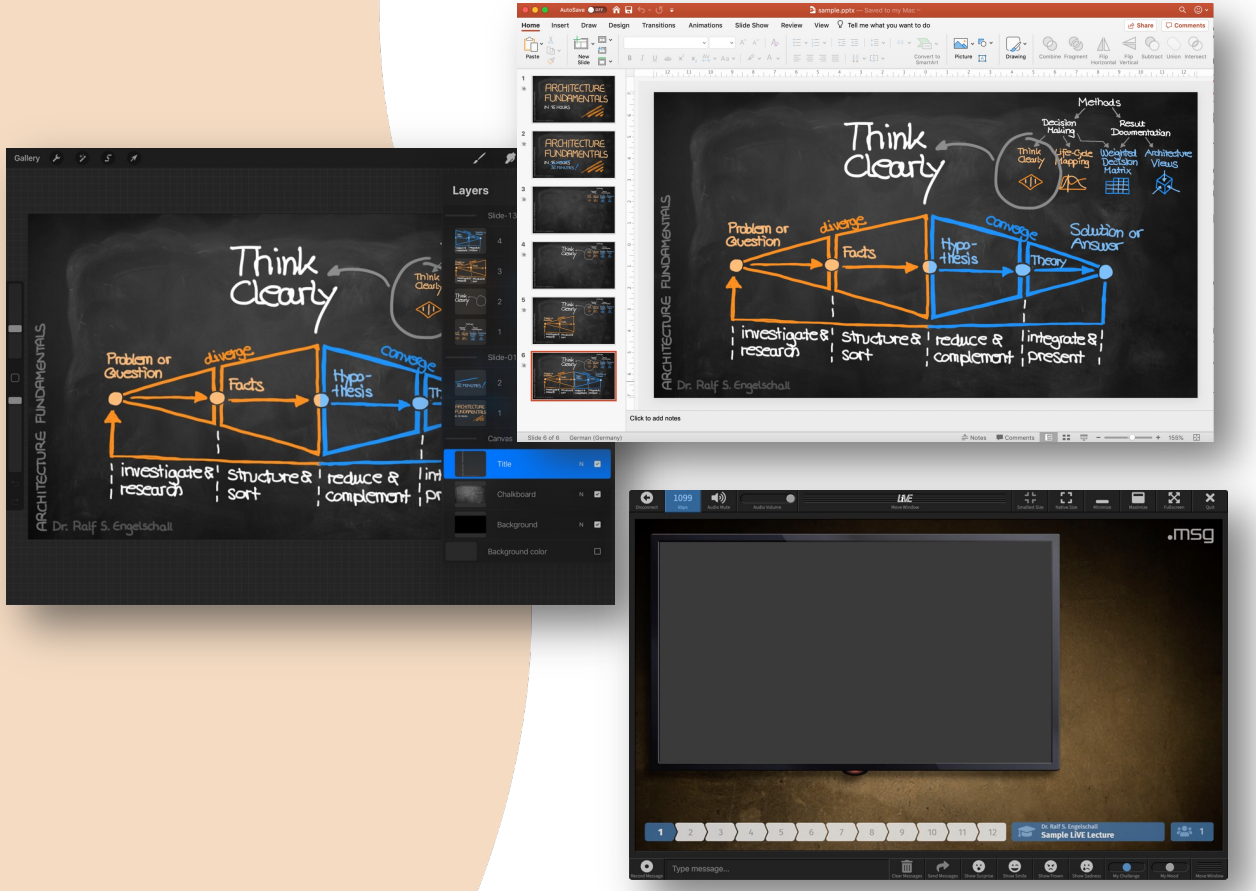


Remember: "multimedia", in this context, primarily refers to the educational content and its various formats (poster, handout, canvas, etc), variants (branding, annotation, etc), combinations of content (image, speech, etc) and channels (print, video, etc), and not just the use of a video-stream!

My Didactics Journey

- *…many experiments…*
- 2019: Hand-Drawn Slides *(https://github.com/rse/psd2pptx)* due to non-vector format, does not allow flexible content zooming and annotation.
- 2020: Desktop Player App *(https://github.com/rse/live-receiver)* download and executable signing causes too high entry barrier.
- *…and even more…*

Software Engineering Academy

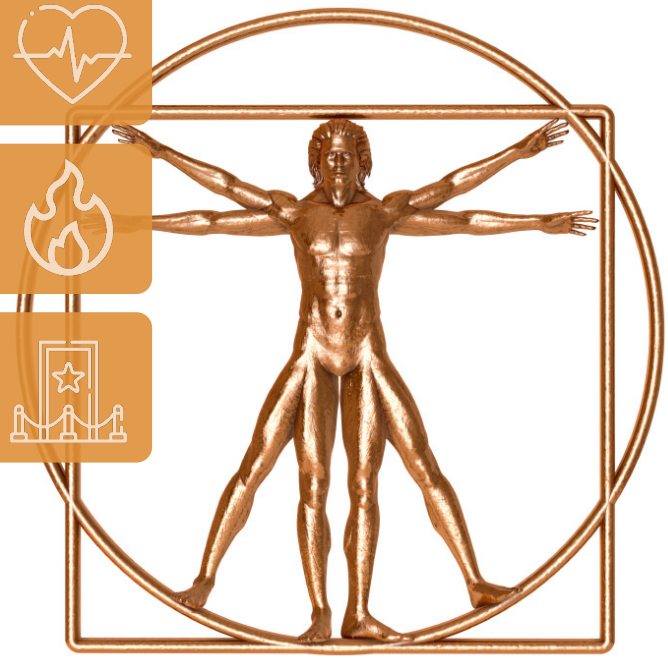


My Education Building Blocks

12 essential building blocks for a
really strong overall lecture experience

Software Engineering Academy

- What-Why-Where Paradigm ■
- Online-First Approach
- Gamification Quizzes ■
- Practice Back-Pressure
- Blended Studio Reality ■
- Live Attendee Feedback
- Diagram-Driven Education ■
- Content Zooming and Annotation ■
- Perception Amplifier
- Generative Lecture Content
- Live Questions and Answers
- Backstage Pass



■ Didactics Crux and Game Changer

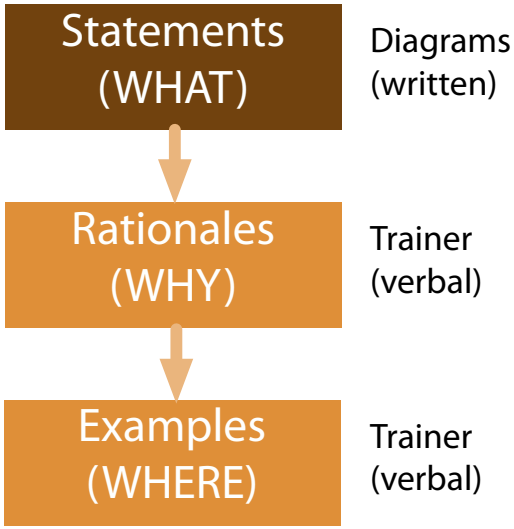
What-Why-Where Paradigm

Building Block 1/12

Software Engineering Academy

- WHAT:
  show the *What* on written material,
  vocally explain the *Why*,
  vocally tell about the practical *Where.*
- WHY (DIDACTICS):
  the *What* has to be remembered,
  the *Why* is the key for acceptance,
  the *Where* is the example to understand it.
- WHY (OTHER):
  students still need both the content (written) and lecturer (vocally), so still attend the lecture.

Didactics Crux!



Practice
Back-Pressure

Building Block 2/12

Software Engineering Academy

- WHAT:
  provide *What* artifacts for practice, use them 1:1 also for education, minimize education-only artifacts.
- WHY (DIDACTICS):
  avoid mental gaps between education and subsequent practice, and practice relevance always convinces.
- WHY (OTHER):
  (none)

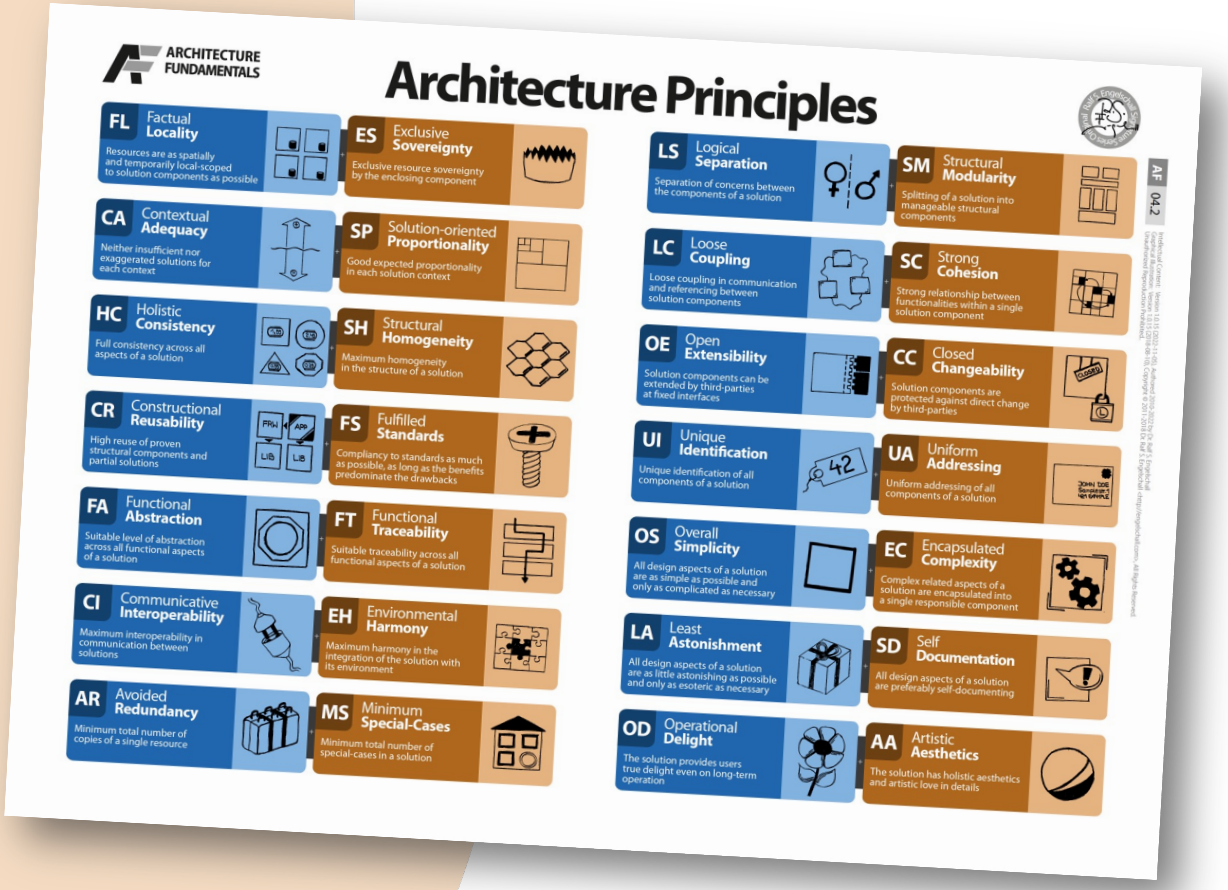


Diagram-Driven Education

Building Block 3/12

Software Engineering Academy

- WHAT:
  describe complex aspects, concepts or methods through visual models on dedicated diagrams.
- WHY (DIDACTICS):
  precision through abstraction and conciseness, and easier comprehension of reality.
- WHY (OTHER):
  visuality, recognition, catchyness.

Didactics Crux!

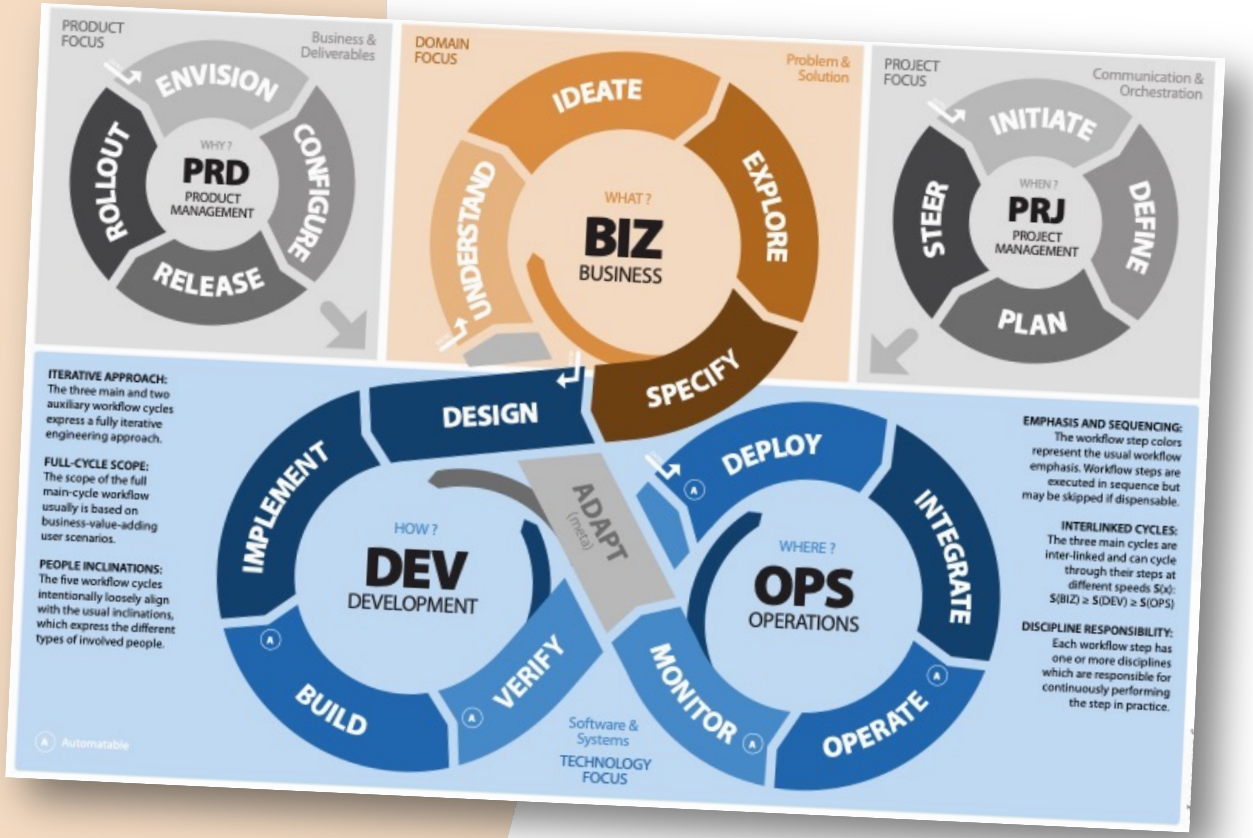
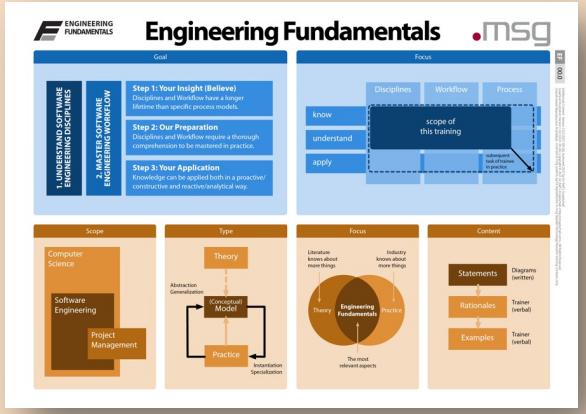
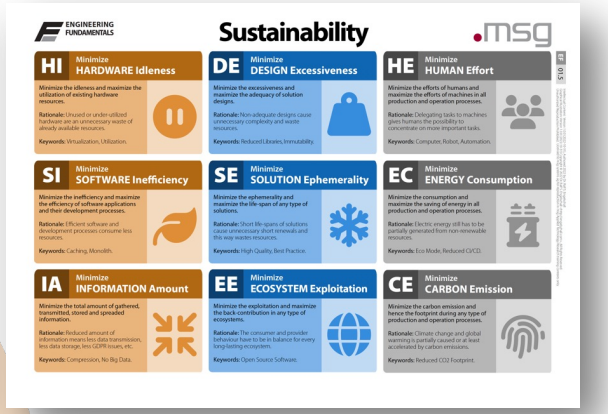


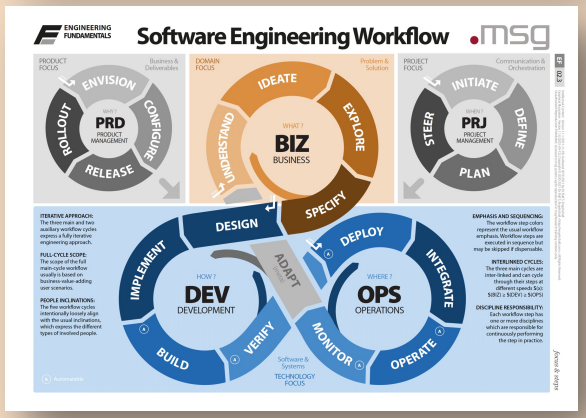
Diagram-Driven Education

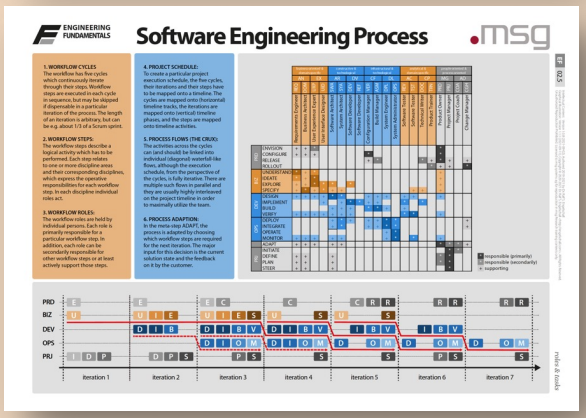
Building Block 3/12

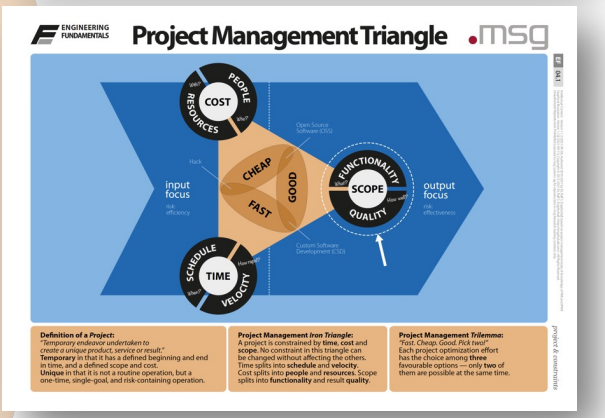
Software Engineering Academy









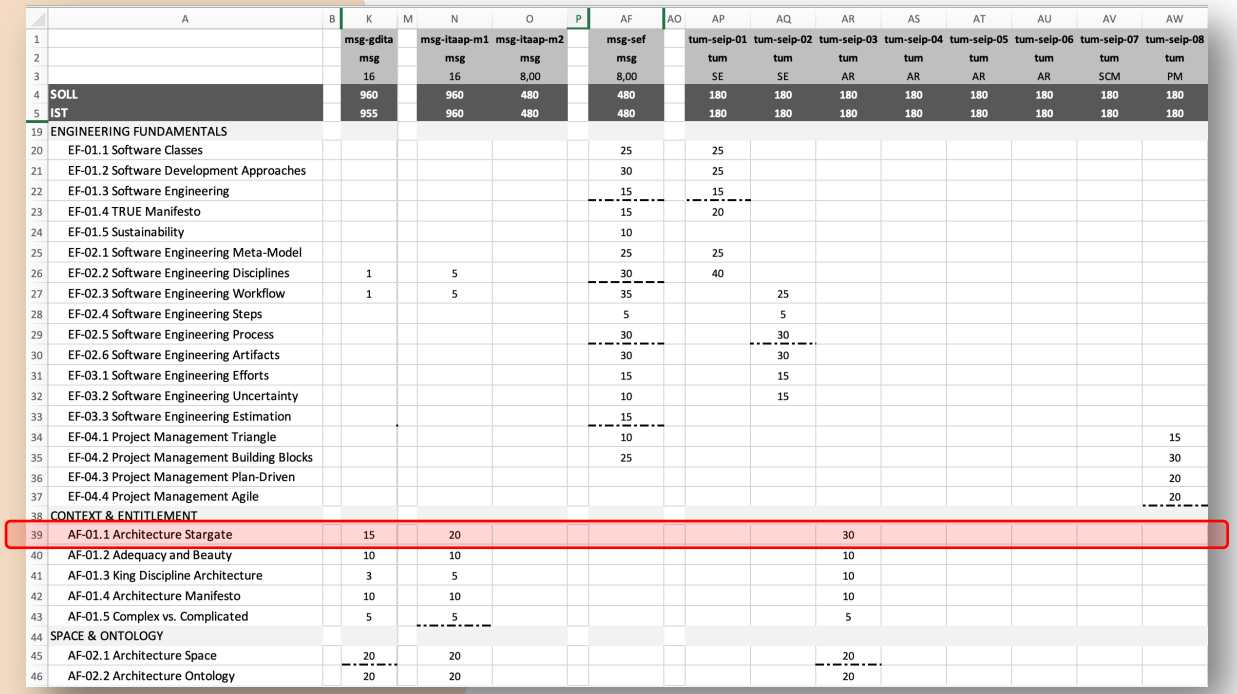


Generative Lecture Content

Building Block 4/12

Software Engineering Academy

- WHAT:
  select arbitrary combination of diagrams, overlay with logo and hints (duration, quiz markers), and generate multiple artifacts (agenda, content, handout, etc).
- WHY (DIDACTICS):
  different content orders, different target groups.
- WHY (OTHER):
  different artifacts, different lecture times, different lecture formats.



	A	K	N	O	AF	AP	AQ	AR	AS	AT	AU	AV	AW
1		msg-gdita	msg-itaap-m1	msg-itaap-m2	msg-sef	tum-seip-01	tum-seip-02	tum-seip-03	tum-seip-04	tum-seip-05	tum-seip-06	tum-seip-07	tum-seip-08
2		msg	msg	msg	msg	tum	tum	tum	tum	tum	tum	tum	tum
3		16	16	8,00	8,00	SE	SE	AR	AR	AR	AR	SCM	PM
4	SOLL	960	960	480	480	180	180	180	180	180	180	180	180
5	IST	955	960	480	480	180	180	180	180	180	180	180	180
19	ENGINEERING FUNDAMENTALS												
20	EF-01.1 Software Classes				25	25							
21	EF-01.2 Software Development Approaches				30	25							
22	EF-01.3 Software Engineering				15	15							
23	EF-01.4 TRUE Manifesto				15	20							
24	EF-01.5 Sustainability				10								
25	EF-02.1 Software Engineering Meta-Model				25	25							
26	EF-02.2 Software Engineering Disciplines	1	5		30	40							
27	EF-02.3 Software Engineering Workflow	1	5		35		25						
28	EF-02.4 Software Engineering Steps				5		5						
29	EF-02.5 Software Engineering Process				30		30						
30	EF-02.6 Software Engineering Artifacts				30		30						
31	EF-03.1 Software Engineering Efforts				15		15						
32	EF-03.2 Software Engineering Uncertainty				10		15						
33	EF-03.3 Software Engineering Estimation				15								
34	EF-04.1 Project Management Triangle				10								15
35	EF-04.2 Project Management Building Blocks				25								30
36	EF-04.3 Project Management Plan-Driven												20
37	EF-04.4 Project Management Agile												20
38	CONTEXT & ENTITLEMENT												
39	AF-01.1 Architecture Stargate	15	20					30					
40	AF-01.2 Adequacy and Beauty	10	10					10					
41	AF-01.3 King Discipline Architecture	3	5					10					
42	AF-01.4 Architecture Manifesto	10	10					10					
43	AF-01.5 Complex vs. Complicated	5	5					5					
44	SPACE & ONTOLOGY												
45	AF-02.1 Architecture Space	20	20					20					
46	AF-02.2 Architecture Ontology	20	20					20					

Generative Lecture Content

Building Block 4/12

Software Engineering Academy

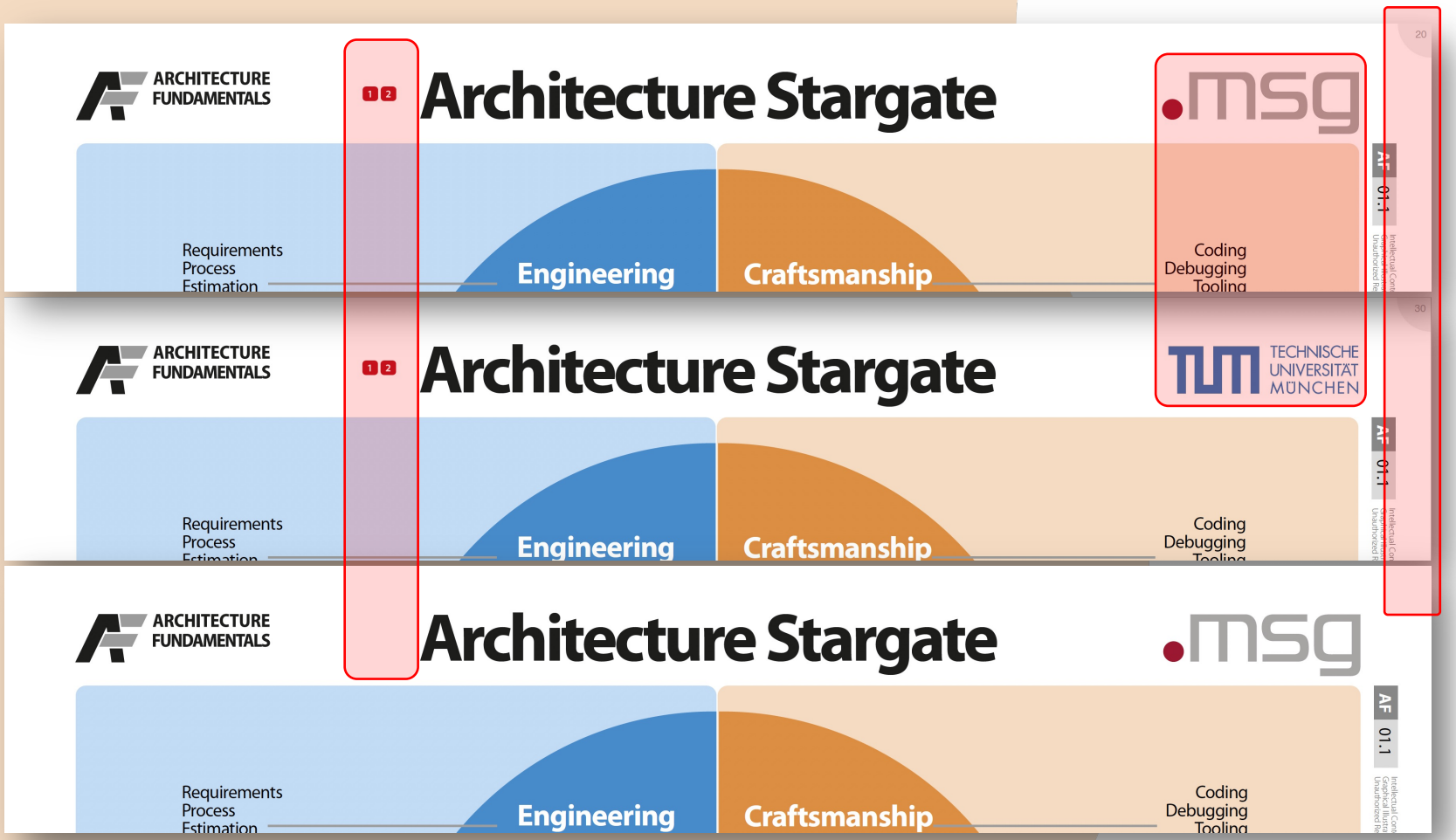
TUM WS22/23, SEIP Modul 03 (2022-11-03)

Begin	End	Duration	Topic
14:00	14:30	00:30	**AF-01.1 Architecture Stargate**
14:30	14:40	00:10	**AF-01.2 Adequacy and Beauty**
14:40	14:50	00:10	**AF-01.3 King Discipline Architecture**
14:50	15:00	00:10	**AF-01.4 Architecture Manifesto**
15:00	15:05	00:05	**AF-01.5 Complex vs. Complicated**
15:05	15:25	00:20	**AF-02.1 Architecture Space**
15:25	15:40	00:15	Break
15:40	16:00	00:20	**AF-02.2 Architecture Ontology**
16:00	17:00	01:00	**AF-04.1 Architecture Maxims**

ITAAP Modul 1

Begin	End	Duration	Topic
09:00	09:20	00:20	Introduction Round
09:20	09:30	00:10	**AF-00.0 Architecture Fundamentals**
09:30	09:35	00:05	**EF-02.2 Software Engineering Disciplines**
09:35	09:40	00:05	**EF-02.3 Software Engineering Workflow**
09:40	10:00	00:20	**AF-01.1 Architecture Stargate**
10:00	10:10	00:10	**AF-01.2 Adequacy and Beauty**
10:10	10:15	00:05	**AF-01.3 King Discipline Architecture**
10:15	10:25	00:10	**AF-01.4 Architecture Manifesto**
10:25	10:30	00:05	**AF-01.5 Complex vs. Complicated**
10:30	10:40	00:10	Break
10:40	11:00	00:20	**AF-02.1 Architecture Space**
11:00	11:20	00:20	**AF-02.2 Architecture Ontology**
11:20	11:25	00:05	**AF-03.1 Requirements Basics**
11:25	12:10	00:45	**AF-04.1 Architecture Maxims**
12:10	12:55	00:45	Lunch
12:55	13:55	01:00	**AF-04.2 Architecture Principles**
13:55	14:25	00:30	**AF-05.1 Component Design**
14:25	14:35	00:10	Break
14:35	15:25	00:50	**AF-05.2 Interface Design**
15:25	15:40	00:15	**AF-05.3 Component Hierarchy**
15:40	15:50	00:10	Break
15:50	16:10	00:20	**AF-06.2 Layer Architecture**
16:10	16:20	00:10	Break
16:20	16:40	00:20	**AF-06.3 Slice Architectures**
16:40	16:55	00:15	**AF-07.1 Flow Architectures**
16:55	17:10	00:15	**AF-07.2 Process Architectures**

Begin	End	Duration	Topic
09:00	09:05	00:05	Recap
09:05	09:25	00:20	**AF-07.3 Cluster Architectures**
09:25	09:45	00:20	**AF-08.1 Networking Architectures**
09:45	10:05	00:20	**AF-08.2 Communication Architectures**
10:05	10:45	00:40	**AF-09.1 Data Structure Architectures**
10:45	10:55	00:10	Break
10:55	11:20	00:25	**AF-10.1 Application Reference Architecture**
11:20	11:35	00:15	**AF-10.2 Client-Server Architecture**
11:35	12:20	00:45	**AF-10.3 Information System Architecture**
12:20	13:05	00:45	Lunch
13:05	13:25	00:20	**AF-10.4 Reactive System Architecture**
13:25	13:35	00:10	**AF-11.3 Application Composition**
13:35	13:45	00:10	**AF-16.1 Think Clearly**
13:45	13:55	00:10	**AF-16.3 Technology Life-Cycles**
13:55	14:10	00:15	**AF-16.4 Open Source Software**
14:10	14:25	00:15	**AF-16.5 Back of the Envelope Calculation**
14:25	14:35	00:10	Break
14:35	14:50	00:15	**AF-16.6 Weighted Decision Matrix**
14:50	15:00	00:10	**AF-16.7 Focus Area Maturity Model**
15:00	15:20	00:20	**AF-17.1 Big Picture**
15:20	15:40	00:20	**AF-17.2 Viewpoints & Perspectives**
15:40	16:05	00:25	AF-Practice: Viewpoints & Perspectives
16:05	16:10	00:05	**AF-19.0 Bibliography**
16:10	16:30	00:20	Summary
16:30	16:50	00:20	Feedback Round

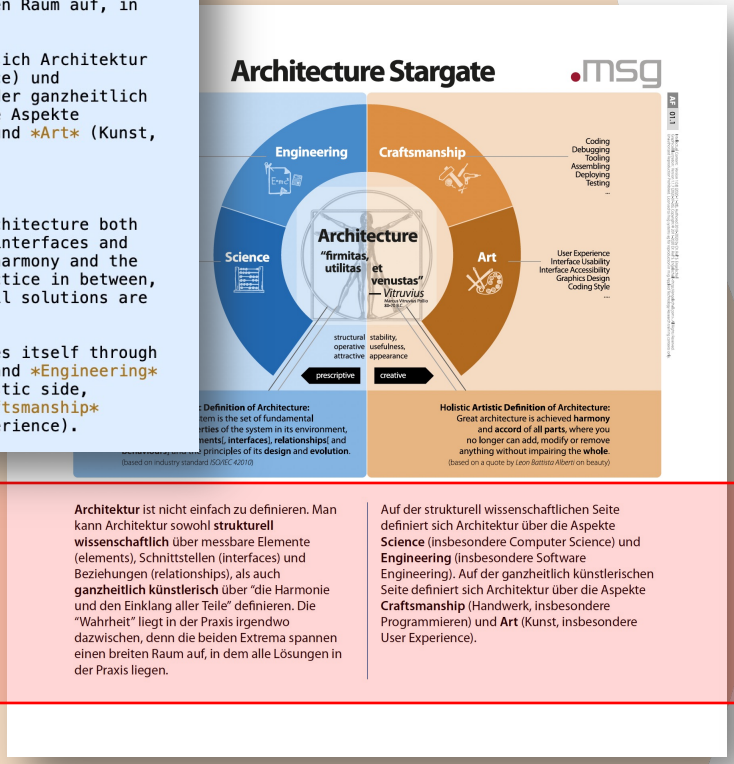

ARCHITECTURE FUNDAMENTALS
1 2
Architecture Stargate
.msg
Requirements
Process
Estimation
Engineering
Craftsmanship
Coding
Debugging
Tooling
AF 01.1
20
ARCHITECTURE FUNDAMENTALS
1 2
Architecture Stargate
TECHNISCHE UNIVERSITÄT MÜNCHEN
Requirements
Process
Engineering
Craftsmanship
Coding
Debugging
AF 01.1
30
ARCHITECTURE FUNDAMENTALS
Architecture Stargate
.msg
Requirements
Process
Estimation
Engineering
Craftsmanship
Coding
Debugging
Tooling
AF 01.1

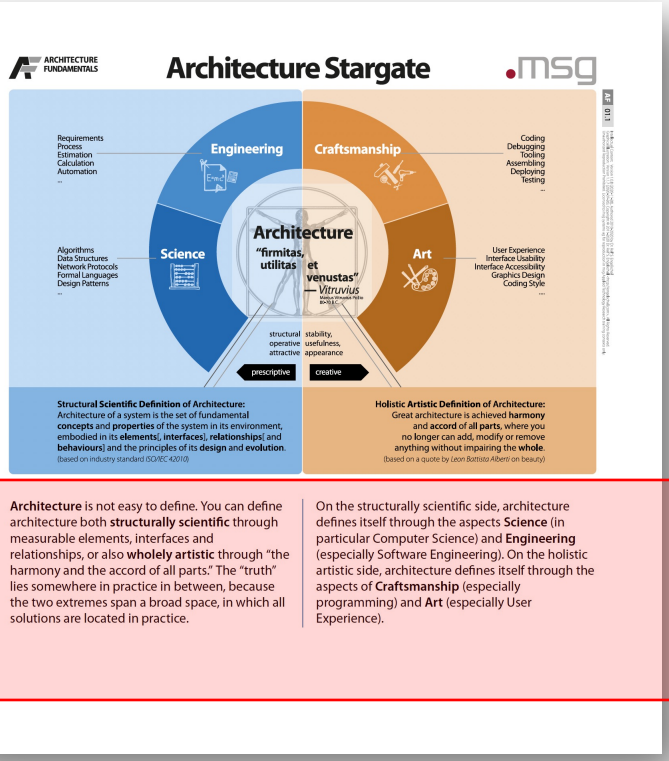
Generative Lecture Content

Building Block 4/12

Software Engineering Academy

```
# Zusammenfassung

*Architektur* ist nicht einfach zu definieren. Man kann Architektur
sowohl *strukturell wissenschaftlich* über messbare Elemente (elements),
Schnittstellen (interfaces) und Beziehungen (relationships), als
auch *ganzheitlich künstlerisch* über "die Harmonie und den Einklang
aller Teile" definieren. Die "Wahrheit" liegt in der Praxis irgendwo
dazwischen, denn die beiden Extrema spannen einen breiten Raum auf, in
dem alle Lösungen in der Praxis liegen.

Auf der strukturell wissenschaftlichen Seite definiert sich Architektur
über die Aspekte *Science* (insbesondere Computer Science) und
*Engineering* (insbesondere Software Engineering). Auf der ganzheitlich
künstlerischen Seite definiert sich Architektur über die Aspekte
*Craftsmanship* (Handwerk, insbesondere Programmieren) und *Art* (Kunst,
insbesondere User Experience).

# Summary

*Architecture* is not easy to define. You can define architecture both
*structurally scientific* through measurable elements, interfaces and
relationships, or also *wholely artistic* through "the harmony and the
accord of all parts." The "truth" lies somewhere in practice in between,
because the two extremes span a broad space, in which all solutions are
located in practice.

On the structurally scientific side, architecture defines itself through
the aspects *Science* (in particular Computer Science) and *Engineering*
(especially Software Engineering). On the holistic artistic side,
architecture defines itself through the aspects of *Craftsmanship*
(especially programming) and *Art* (especially User Experience).
```





Architektur ist nicht einfach zu definieren. Man kann Architektur sowohl **strukturell wissenschaftlich** über messbare Elemente (elements), Schnittstellen (interfaces) und Beziehungen (relationships), als auch **ganzheitlich künstlerisch** über "die Harmonie und den Einklang aller Teile" definieren. Die "Wahrheit" liegt in der Praxis irgendwo dazwischen, denn die beiden Extrema spannen einen breiten Raum auf, in dem alle Lösungen in der Praxis liegen.

Auf der strukturell wissenschaftlichen Seite definiert sich Architektur über die Aspekte **Science** (insbesondere Computer Science) und **Engineering** (insbesondere Software Engineering). Auf der ganzheitlich künstlerischen Seite definiert sich Architektur über die Aspekte **Craftsmanship** (Handwerk, insbesondere Programmieren) und **Art** (Kunst, insbesondere User Experience).

Architecture is not easy to define. You can define architecture both **structurally scientific** through measurable elements, interfaces and relationships, or also **wholely artistic** through "the harmony and the accord of all parts." The "truth" lies somewhere in practice in between, because the two extremes span a broad space, in which all solutions are located in practice.

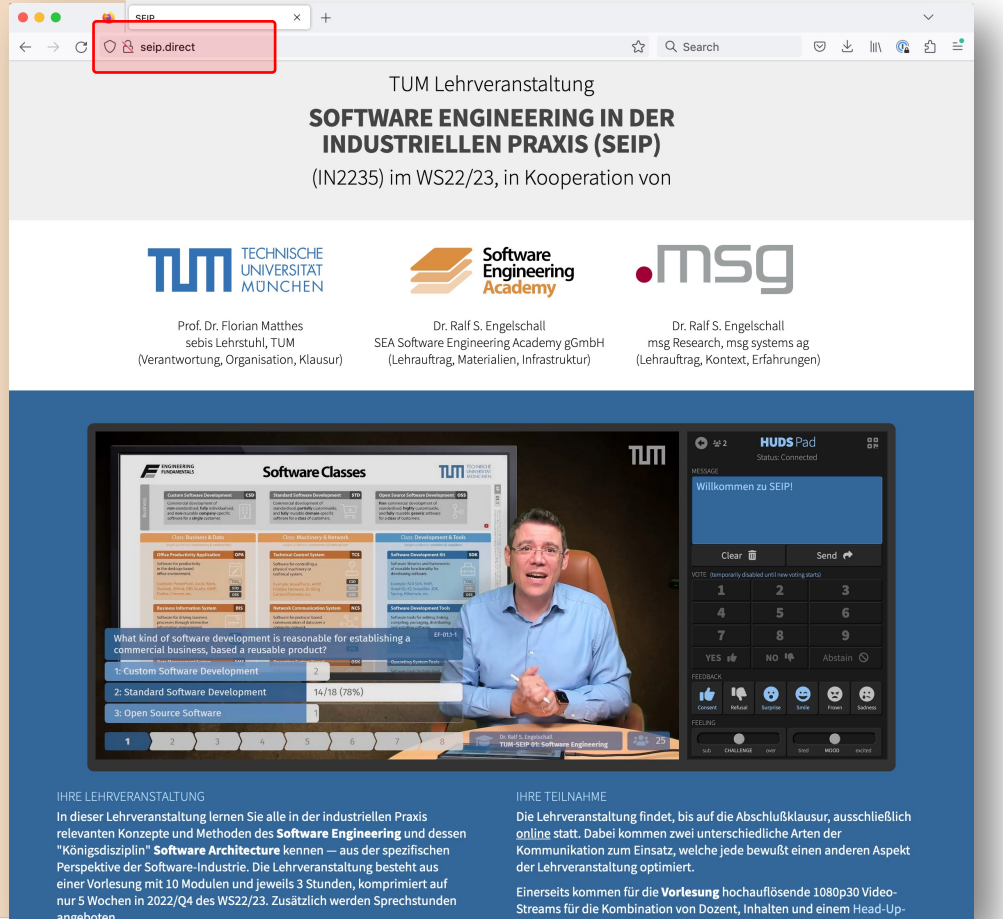
On the structurally scientific side, architecture defines itself through the aspects **Science** (in particular Computer Science) and **Engineering** (especially Software Engineering). On the holistic artistic side, architecture defines itself through the aspects of **Craftsmanship** (especially programming) and **Art** (especially User Experience).

Online-First Approach

Building Block 5/12

Software Engineering Academy

- WHAT:
  design lecture setup primarily for online performance, use reduced setup for on-site performance.
- WHY (DIDACTICS):
  Generation-Z expectations (*YouTube/TikTok*-style consumption), post-pandemic *New Work* era, fresher attendees.
- WHY (OTHER):
  (none)



Blended Studio Reality

Building Block 6/12

Software Engineering Academy

- WHAT:
  produce lecture in studio environment, with real-time video mixing. Blend content on tablet, lecturer at table, *Head-Up-Display (HUD)* and special effects. Use unified colouring theme.
- WHY (DIDACTICS):
  acceptance through consistency and seamless blending.
- WHY (OTHER):
  Generation-Z expectations (video-streaming style), overall user-experience.

Didactics Crux!

Content Zooming and Annotation

Building Block 7/12

Software Engineering Academy

- WHAT:
  stay at and focus on single diagram for at least 15-20 minutes in total. Zoom around in diagram and on-the-fly annotate it.
- WHY (DIDACTICS):
  strong content focus during lecture, easily adjust “depth of lecturing”.
- WHY (OTHER):
  Anti-*Death-By-PowerPoint.*

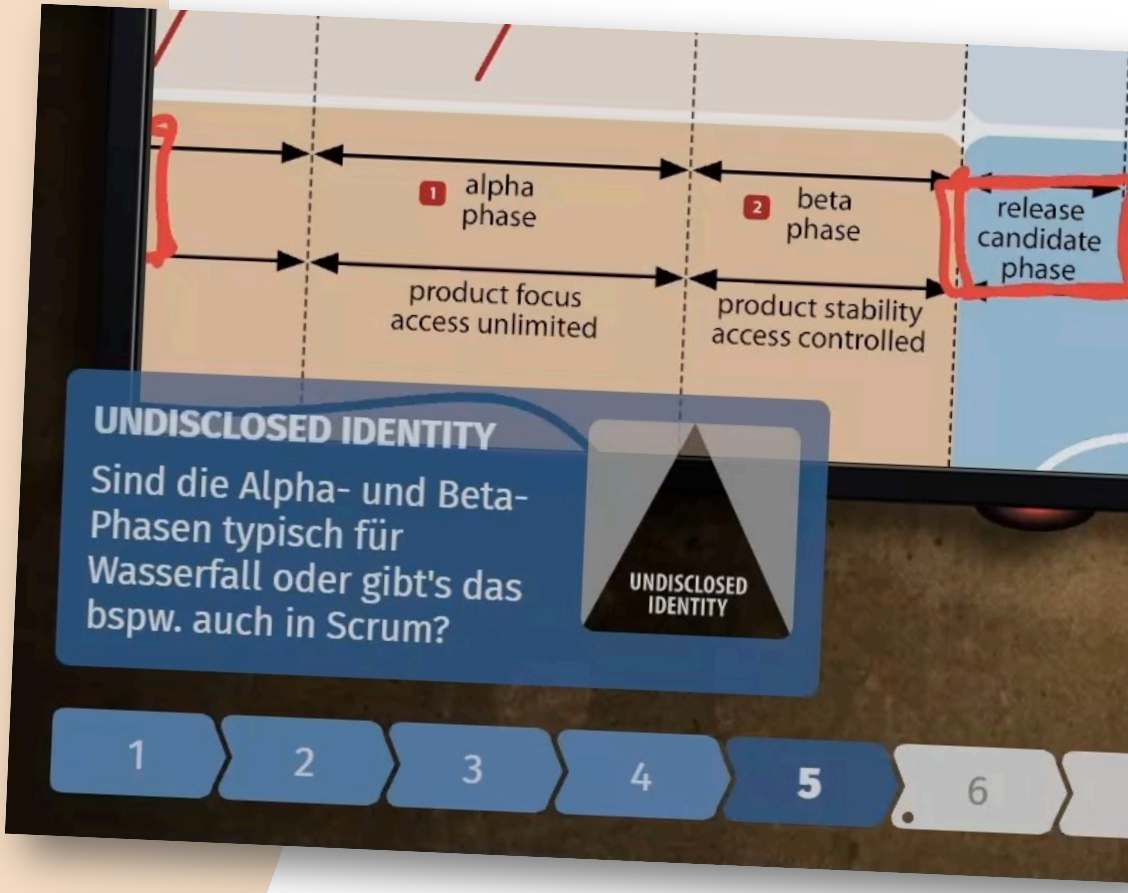
Didactics Crux!

Live Questions and Answers

Building Block 8/12

Software Engineering Academy

- WHAT:
  allow students to (optionally anonymously) raise questions and statements at any time, work them off sequentially.
- WHY (DIDACTICS):
  direct student involvement, no direct lecturer interruption, stronger questions & answers focus, less misunderstandings.
- WHY (OTHER):
  YouTube/Twitch-style chat communication of *Generation-Z*.



Gamification Quizzes

Building Block 9/12

Software Engineering Academy

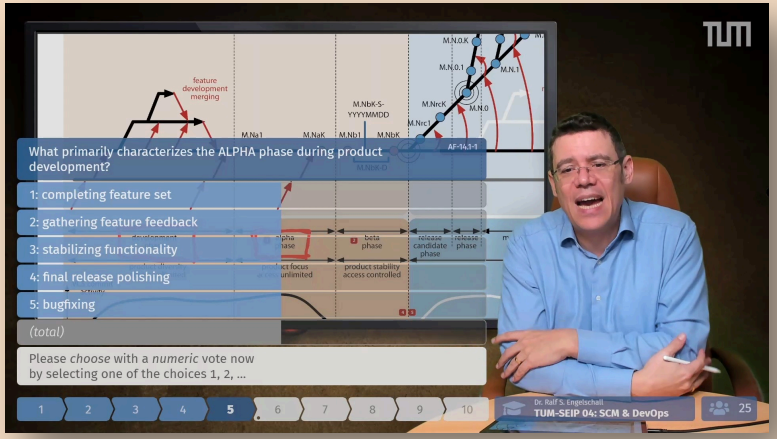
- WHAT:
  raise (potentially “beyond one’s own nose”) questions about every 10-15 minutes, let students vote anonymously and in real-time.
- WHY (DIDACTICS):
  continuous student involvement, loosening up lecture, competition fun.
- WHY (OTHER):
  (none)

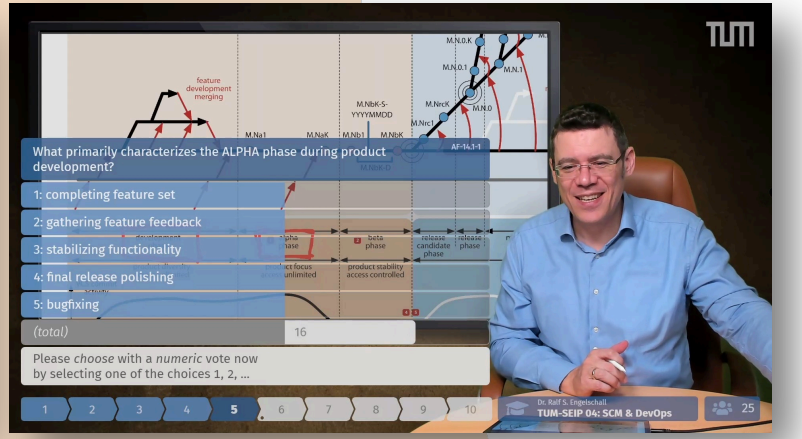
Didactics Crux!

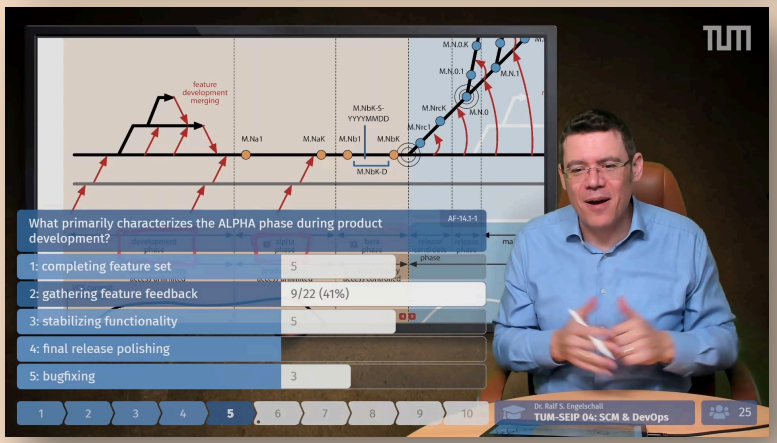
Gamification Quizzes

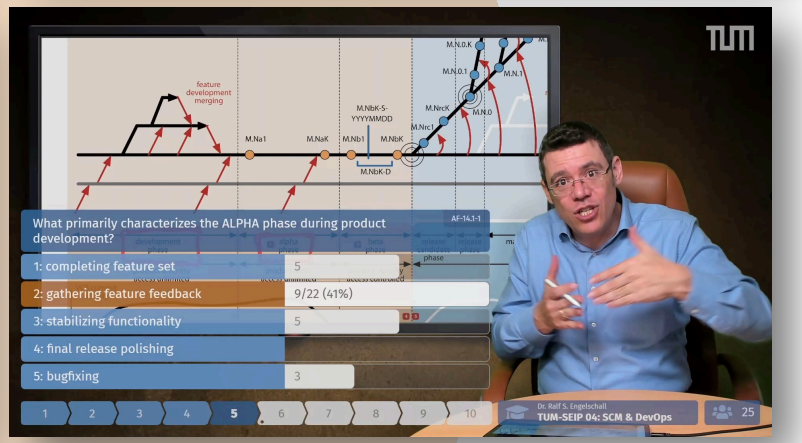
Building Block 9/12

Software Engineering Academy







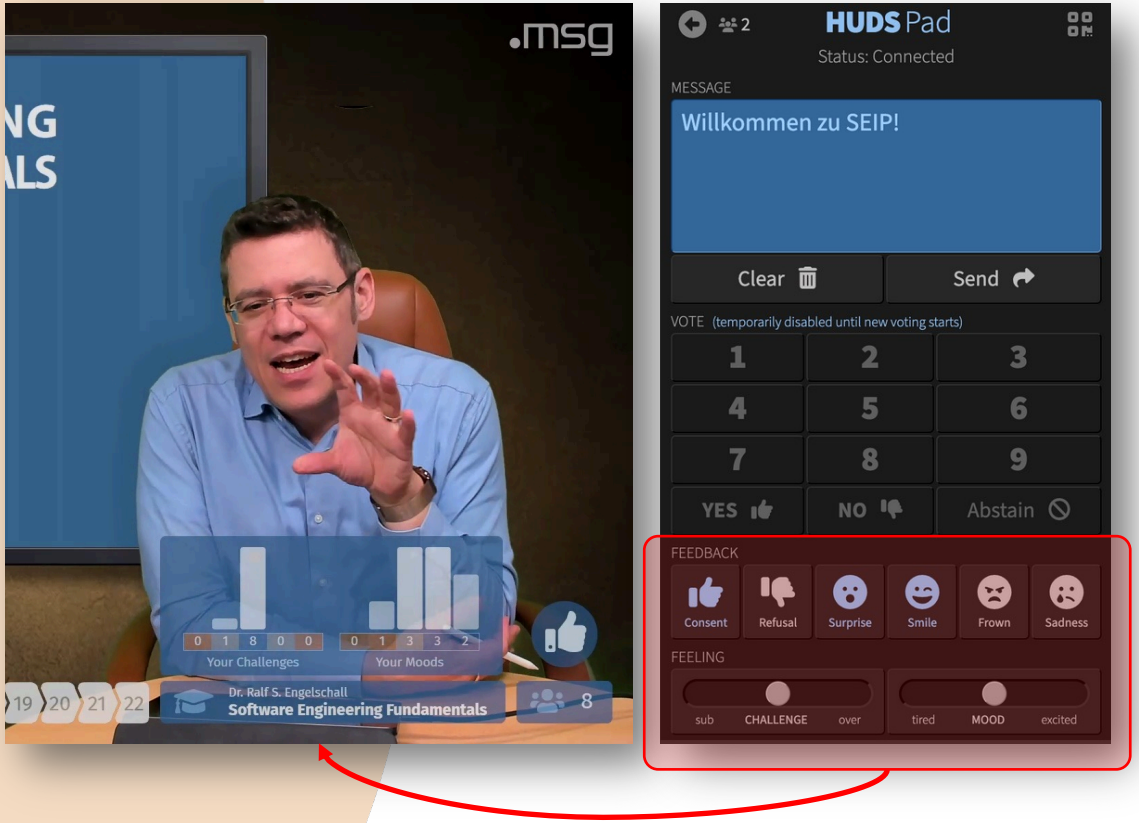


Live Attendee Feedback

Building Block 10/12

Software Engineering Academy

- WHAT:
  allow students to continuously share their emotions (via *Emojis*), content-wise challenge and personal mood.
- WHY (DIDACTICS):
  real-time student feedback, lecturer speed/depth adjustment, inter-student sympathy.
- WHY (OTHER):
  annotation of lecture, perception of even “remote controlling” the lecture.

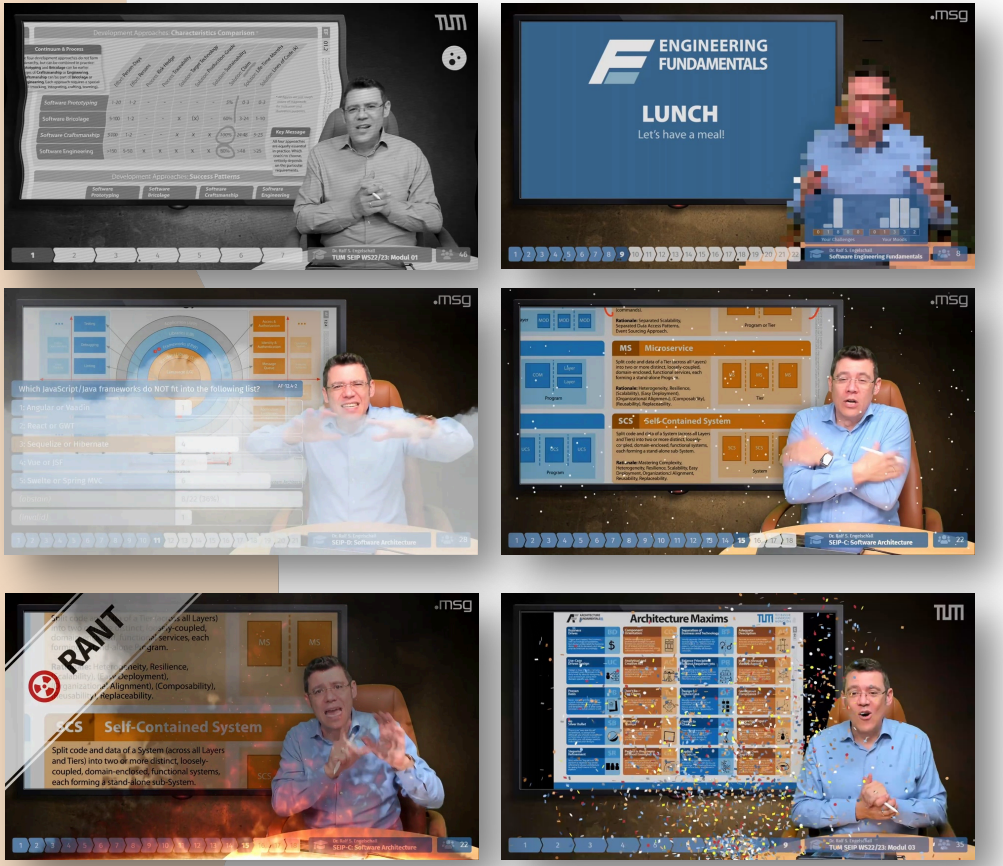


Perception Amplifier

Building Block 11/12

Software Engineering Academy

- WHAT:
  use of curtains, banners, pause clock, and a set of overlayed special effects.
- WHY (DIDACTICS):
  draw extra attention, amplify content perception.
- WHY (OTHER):
  (none)



Backstage Pass

Building Block 12/12

Software Engineering Academy

- WHAT:
  during breaks, lecturer gives students live backstage view and explanation of lecture production.
- WHY (DIDACTICS):
  inspiration and motivation for own presentations.
- WHY (OTHER):
  proof of media competence, also in *Generation-X*.

My Education Status Quo

Some proving excerpts (in german)
from official lecture evaluations:

[…]die Balance aus informativ und unterhaltsam, eigentlich die gesamte Vorlesung ist überragend.
– TUM, SEIP, WS 22/23

Die Professionalität […] übersteigt alles was ich bisher in meiner Studentenlaufbahn erleben durfte.
– TUM, SEIP, WS 22/23

Definitiv die beste Vorlesung, die ich je besucht habe!
– TUM, SEIP, WS 22/23

Eine tolles Maß an "Gamification"
– TUM, SEIP, WS 20/21

[…] eine der didaktisch besten Lehrveranstaltungen, die ich JEMALS besucht habe.
– TUM, SEIP, WS 20/21

Die Vorlesung ist bei weitem die aufwändigste und beste, die ich je an der TUM […] gesehen habe.
– TUM, SEIP, WS 22/23

Die Medien und das Setup sind unfassbar gut!
– TUM, SEIP, WS 21/22

Software Engineering Academy

TECHNISCHE UNIVERSITÄT MÜNCHEN

Company:	SEA Software Engineering Academy gGmbH
Register:	HRB 26 92 37 (Amtsgericht München)
Shareholder:	Dr. Ernst Denert, Dr. Ralf S. Engelschall
Director:	Dr. Ralf S. Engelschall
VAT-Id (USt-IdNr):	DE 351 992 667
Web:	https://softeng.academy
Email:	contact@softeng.academy
Postal:	Weblinger Weg 28, 85221 Dachau, DE
IBAN:	DE93 700 700 24 0 798 1 798 00
Phone:	+49 8131 7799780

Software Engineering Academy

Thanks for your attention!