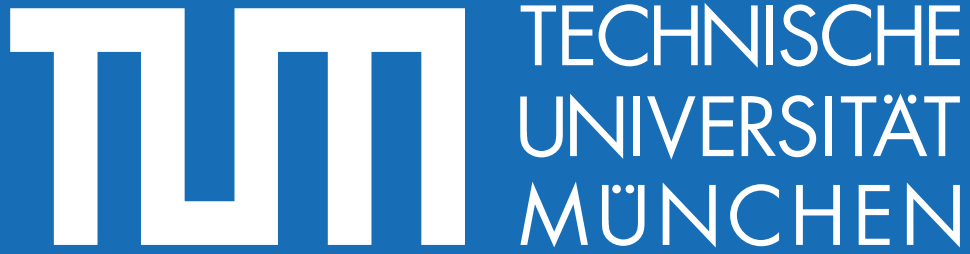


Software Engineering in der industriellen Praxis (SEIP)

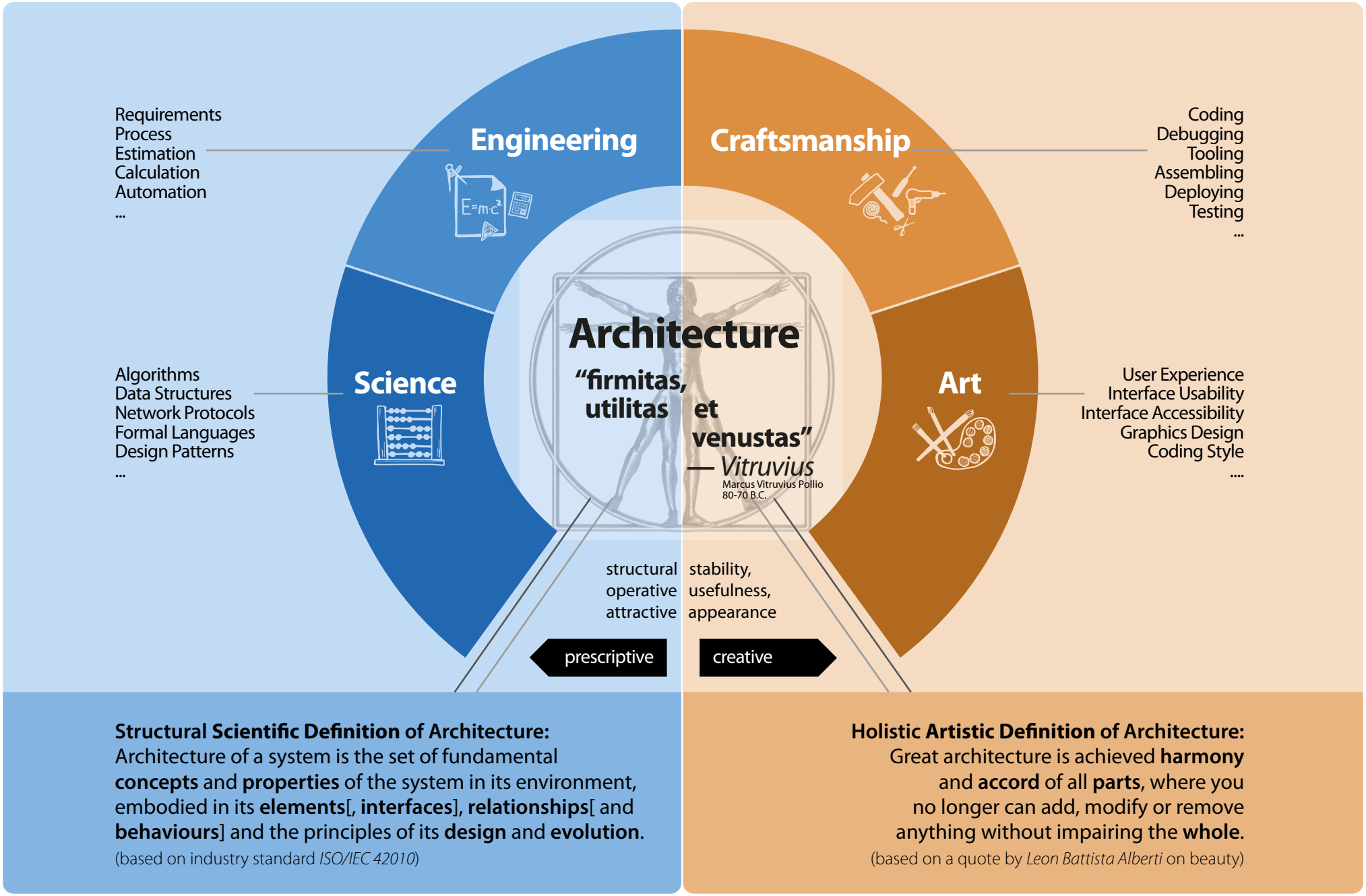
Dr. Ralf S. Engelschall

ARCHITECTURE FUNDAMENTALS

Architecture Stargate

TECHNISCHE UNIVERSITÄT MÜNCHEN

AF 01.1



Structural Scientific Definition of Architecture:
Architecture of a system is the set of fundamental **concepts** and **properties** of the system in its environment, embodied in its **elements**[, **interfaces**], **relationships**[and **behaviours**] and the principles of its **design** and **evolution**.
(based on industry standard *ISO/IEC 42010*)

Holistic Artistic Definition of Architecture:
Great architecture is achieved **harmony** and **accord** of all **parts**, where you no longer can add, modify or remove anything without impairing the **whole**.
(based on a quote by *Leon Battista Alberti* on beauty)

Intellectual Content: Version 1.0.8 (2020-11-28), Authored 2010-2020 by Dr. Ralf S. Engelschall
Graphical Illustration: Version 1.1.7 (2020-07-26), Copyright © 2011-2020 Dr. Ralf S. Engelschall <http://engelschall.com> All Rights Reserved.
Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contexts only.

ARCHITECTURE FUNDAMENTALS

Adequacy and Beauty

TECHNISCHE UNIVERSITÄT MÜNCHEN

AF 01.2

ADEQUACY

1 **Measure**, how **suitable**
2 something,
3 in its environment,
1 fulfills
4 its imposed functional and qualitative requirements (fitness for purpose),
5 from the **individual** and **subjective** perspective of a person.

Beauty does imply **Adequacy**

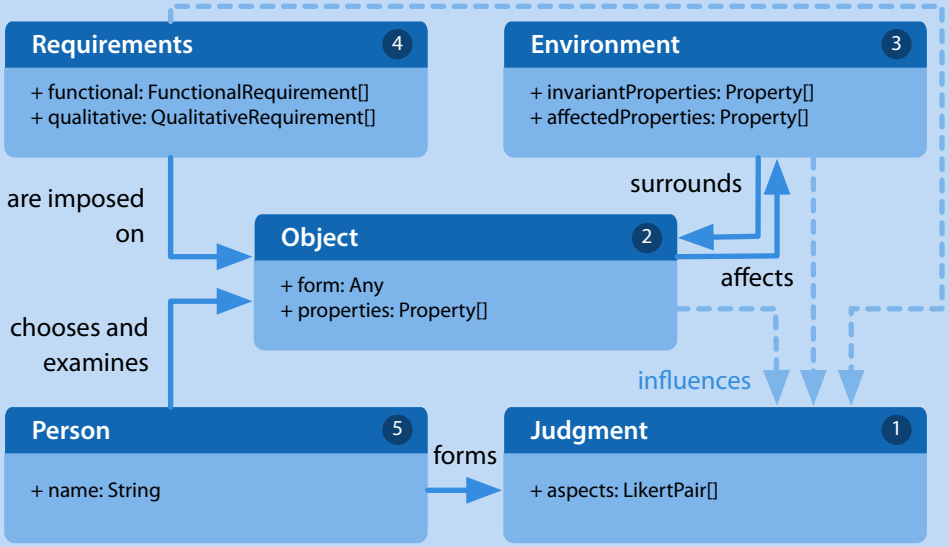
No Beauty does *not* imply No Adequacy

No Adequacy does imply **No Beauty**

Adequacy does *not* imply **Beauty**

BEAUTY

Perception, how **simple, elegant, natural, securing and evident** (SENSE) something is, from the **evolutionary** and **cultural** perspective of a person.



Simple S
free from *unnecessary* complexity, easy *enough* to comprehend.

Elegant E
broadly applicable, allows combinatorial use-cases.

Natural N
consistent, symmetric, free of anomalies.

Securing S
allows *feeling* safe by captured experiences or omnipresence.

Evident E
obvious, clear and traceable representation.

Adequacy is a **relative** and highly **subjective measure** that defies scientific objectification. Nevertheless, it is an essential concept, as the **three influencing aspects** allow persons in practice to at least better **structure** their **perception and judgment** and **measure** the **differences** over time.

Beauty is a **relative** and highly **subjective perception** that defies scientific objectification. Nevertheless, it is an essential concept, as the **SENSE aspects** *seem* to be universal across persons and allow them to **communicate** their **perception and judgment** and **observe** the **differences** over time.

Intellectual Content: Version 1.0.2 (2022-10-09), Authored 2021-2022 by Dr. Ralf S. Engelschall based on pre-work of Stefan Brütsch.
Graphical Illustration: Version 1.0.2 (2023-10-09), Copyright © 2021-2022 Dr. Ralf S. Engelschall <http://engelschall.com> All Rights Reserved.
Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contexts only.

ARCHITECTURE FUNDAMENTALS

King Discipline Architecture

TECHNISCHE UNIVERSITÄT MÜNCHEN

AF 01.3

Intellectual Content: Version 1.0.0 (2022-10-09), Authored 2022 by Dr. Ralf S. Engelschall
Graphical Illustration: Version 1.0.0 (2023-10-09), Copyright © 2023 Dr. Ralf S. Engelschall <http://engelschall.com> All Rights Reserved
Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contexts only.

Domain

Many specific Domains

N

Domain
Analytics
Business

domain-specific
usage-oriented
specific

Architecture

One (or few) generic Architectures

1

Architecture
Design
Construction

conceptual
constructional
generic

Technology

Many specific Technologies

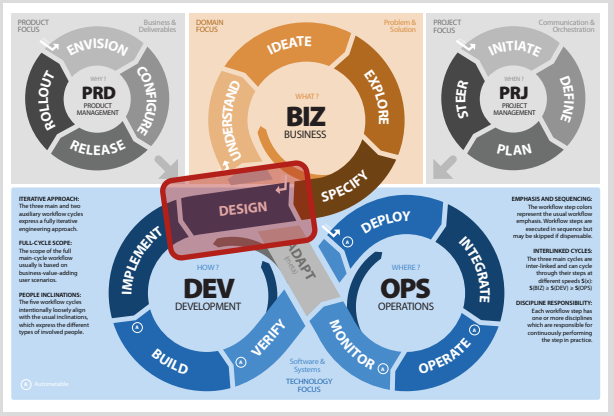
M

Technology
Development
Realisation

technological
programmatic
specific

Software Engineering Disciplines

Software Engineering Workflow



Architecture Manifesto

Manifesto for IT Architecture

Continuously Raising the Bar

Mission As IT Architects we guide the design, implementation and evolution of IT solutions.

Entitlement We continuously strive to raise the bar of professional IT architecture by practicing it and helping others to learn our craft. We achieve maximum value for our clients through our work.

Values Through this work we have come to value aspects of our craft. While we acknowledge the beneficial values in the items on the right, we appreciate the stronger values in the items on the left even more.

Sustainable Concepts	over	**Latest Technologies**
Pragmatic Making	over	**Theoretical Consideration**
Constructive Craftsmanship	over	**Analytical Engineering**
Accredited Creativity	over	**Achieved Industrialization**
Proactive Improvement	over	**Reactive Correction**
Inherent Quality	over	**Tested Robustness**
Operational Delight	over	**Useful Functionality**

Intellectual Content: Version 1.0.2 (2013-11-02), Authored 2013 by Dr. Ralf S. Engelschall
Graphical Illustration: Version 1.1.1 (2018-11-06), Copyright © 2013-2018 Dr. Ralf S. Engelschall <http://engelschall.com> All Rights Reserved.
Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contexts only.

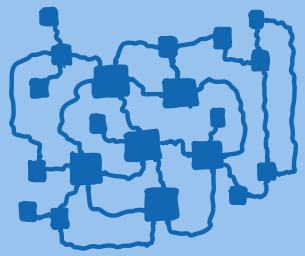
ARCHITECTURE FUNDAMENTALS

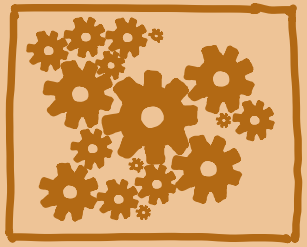
Complex vs. Complicated

TECHNISCHE UNIVERSITÄT MÜNCHEN

AF 01.5

	complex	complicated
FOCUS	refers to the **extrinsic** and **higher**- or **macro**-level difficulty of a system,	refers to the **intrinsic** and **lower**- or **micro**-level difficulty of a system,
RATIONALE	because the system involves many different and **connected** parts	because the system involves many different and **difficult** aspects
CHALLENGE	which take time to **comprehend** and **master** in **total**,	which take time to **understand** and **learn** in **detail**,
INSIGHT	and which nevertheless are **easy** to explain.	and which usually are **hard** to explain.





NOTICE	NOTICE
Simple (non-complicated) systems can be **complex**.	**Clear** (non-complex) systems can be **complicated**.
RECOGNIZE	**RECOGNIZE**
Architecture primarily has to master the **complex** aspects of a system.	**Development** primarily has to master the **complicated** aspects of a system.

Intellectual Content: Version 1.0.0 (2019-10-12), Authored 2019 by Dr. Ralf S. Engelschall
Graphical Illustration: Version 1.0.0 (2019-10-12), Copyright © 2019 Dr. Ralf S. Engelschall <http://engelschall.com> All Rights Reserved
Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contexts only.

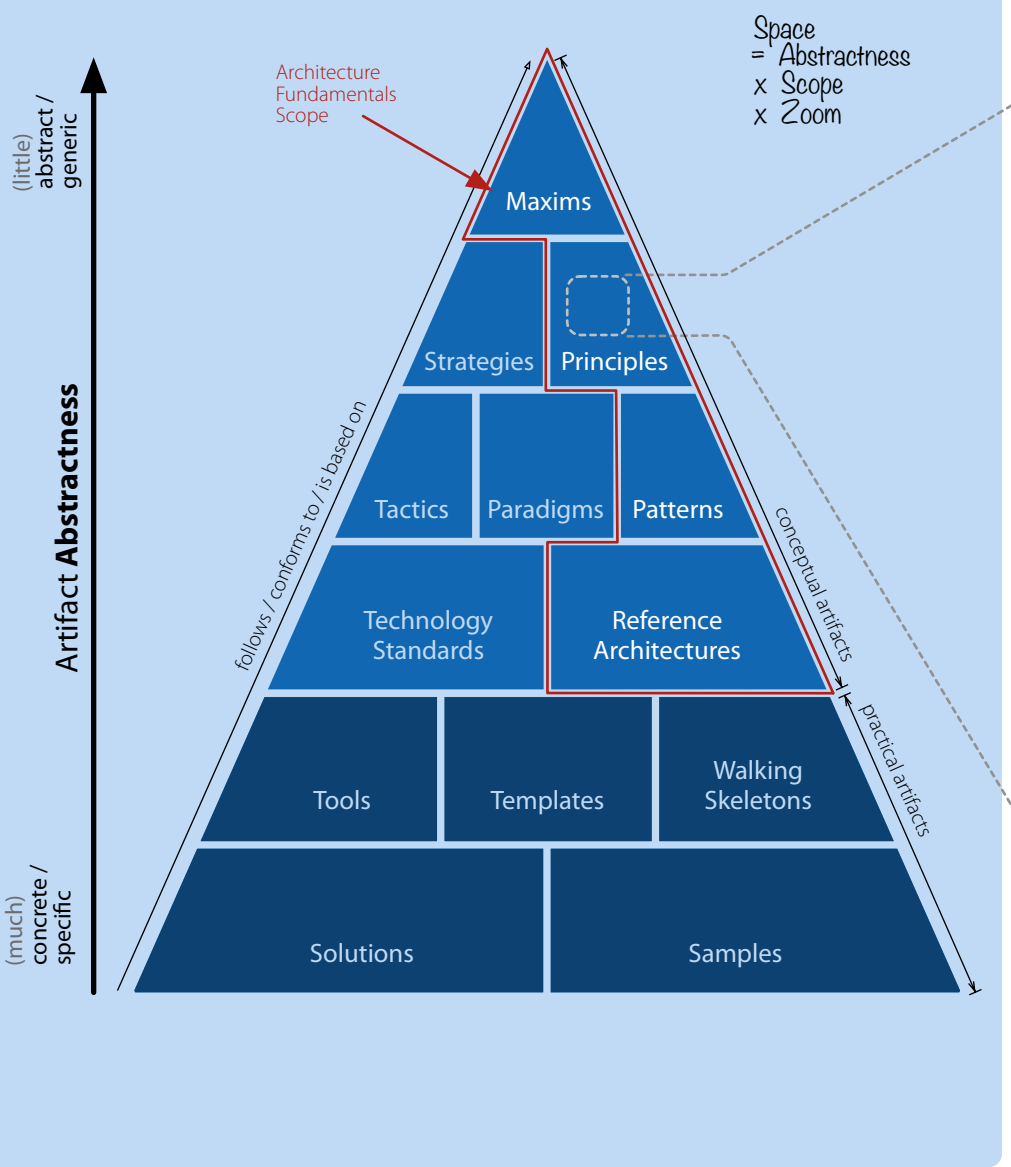
ARCHITECTURE FUNDAMENTALS

Architecture Space

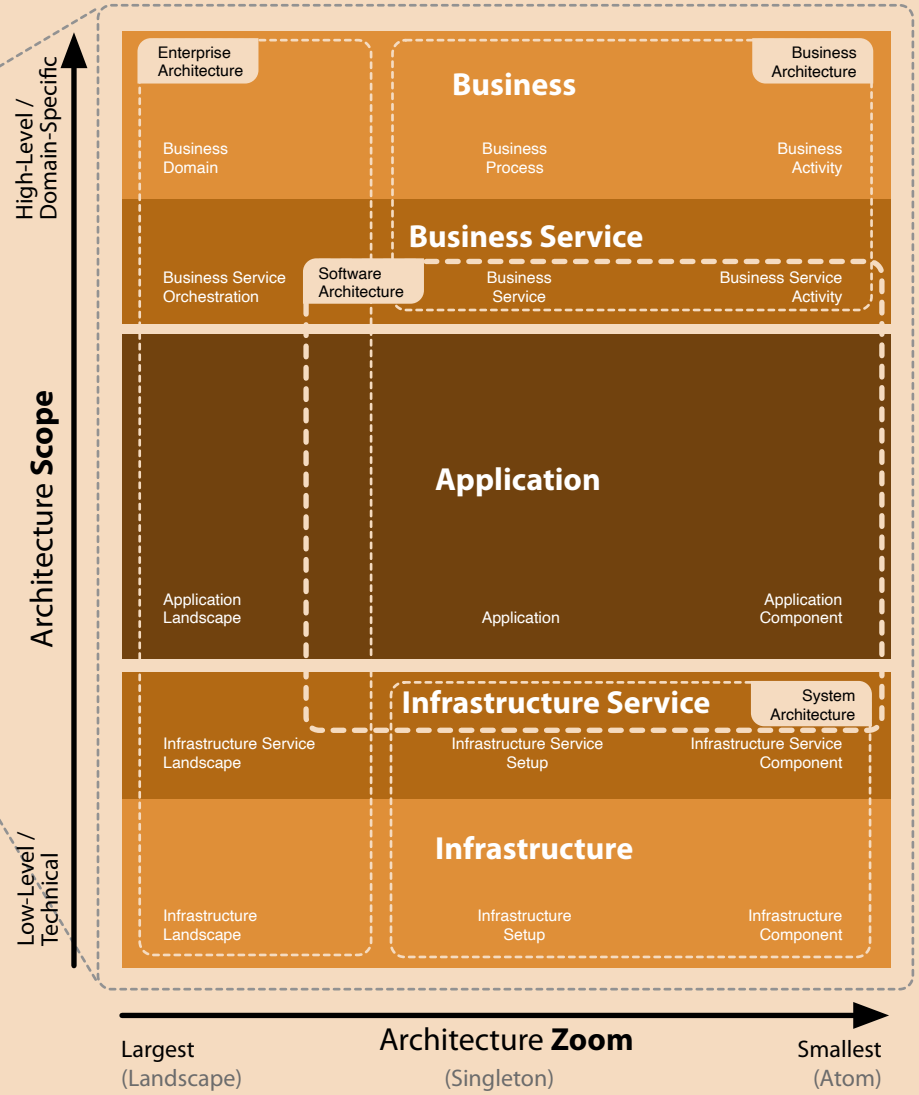
TECHNISCHE UNIVERSITÄT MÜNCHEN

AF 02.1

Artifact Abstractness



Architecture Scope & Zoom

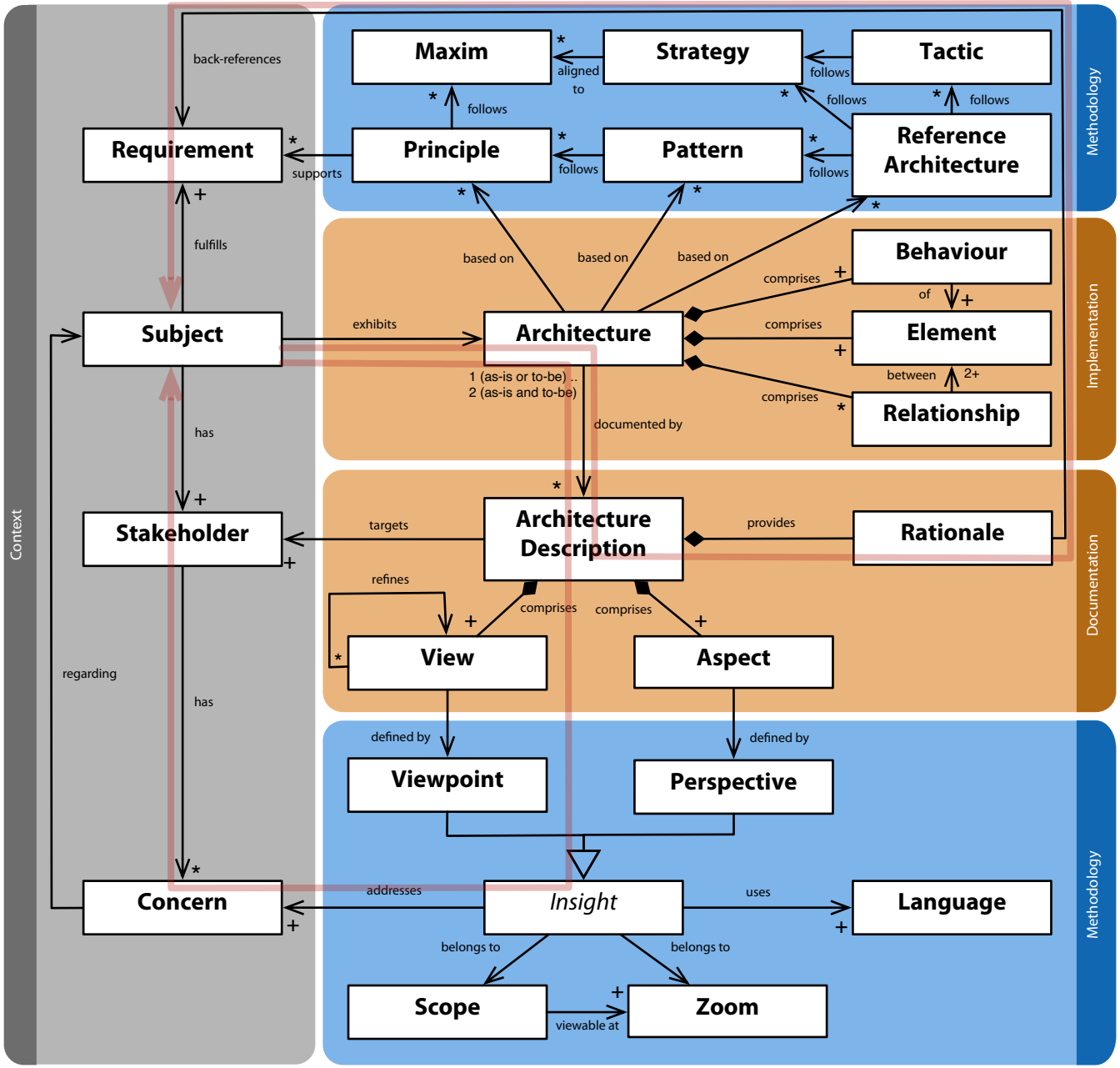


Intellectual Content: Version 1.0.9 (2020-06-23), Authored 2011-2020 by Dr. Ralf S. Engelschall and Marc Mählitzer, based on pre-work of msg Applied Technology Research
Graphical Illustration: Version 1.1.2 (2019-09-15), Copyright © 2011-2019 Dr. Ralf S. Engelschall <http://engelschall.com> All Rights Reserved.
Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contexts only.

ARCHITECTURE FUNDAMENTALS

Architecture Ontology

TECHNISCHE UNIVERSITÄT MÜNCHEN



Requirement: A functional or non-functional demand or imposed obligation on the **Subject**.

Architecture: Inherent static and dynamic structure of a **Subject** which comprise **Elements**, the visible **Behaviour** of **Elements** and **Relationships** between **Elements**.

Maxim: Fundamental, generally valid set of values and rules to guide the architecture discipline (think: law).

Subject: Any type of business process, business service, software application, infrastructure service or infrastructure setup.

Element: Fundamental part from which a **Subject** can be considered to be constructed.

Strategy: Long-term situation-independent plan of approaches to achieve a particular goal (think: war). Aligned and not in conflict with any **Maxims**.

Stakeholder: Person, group or entity with an interest in or **Concerns** about the **Architecture**.

Behaviour: Run-time characteristic of **Architecture Elements** of the **Subject**.

Tactic: Short-term situation-dependent plan of actions to achieve a particular goal (think: battle). Following and supporting a **Strategy**.

Concern: Requirement, objective, intention, or aspiration a **Stakeholder** has on an **Architecture**.

Relationship: Static or dynamic relationship between **Elements** of an **Architecture** of the **Subject**.

Principle: Fundamental truth, rule, tenet or policy an **Architecture** follows.

Architecture Description: Set of artifacts that document an **Architecture** in a way **Stakeholders** can understand and ensures their **Concerns** are met.

Rationale: Fundamental reasons for a particular chosen **Architecture**, usually strongly based on non-functional **Requirements**.

Pattern: Proven recurring theme, structure, approach or behavior an **Architecture** and its **Elements** can follow.

Viewpoint: Collection of templates and conventions for constructing one type of **View**. It addresses **Concerns** and contains guidelines for constructing a View.

View: Representation of one or more structural aspects of an **Architecture** that illustrates how the **Architecture** addresses one or more **Concerns**.

Reference Architecture: Reusable proven Architecture template, **based** on a set of Patterns and following one or more **Tactics** and **Strategies**.

Perspective: Collection of decisions, guidelines and rules that ensure a **Subject** exhibits a set of non-functional **Requirements**, considered across a number of **Views**.

Aspect: Representation of one or more non-functional aspects of an **Architecture** that illustrate how the **Architecture** addresses one or more **Concerns**.

Scope: Primary area of the **Architecture** space an **Insight** addresses: either Business, Business Service, Application, Infrastructure Service or Infrastructure.

Insight: Superordinate abstract concept to address particular **Concerns** through **Viewpoints** and **Perspectives**.

Language: Formal, semi-formal or even prose language to describe an Insight, so it optimally addresses the **Concerns** of **Stakeholders**.

Zoom: Level of view distance to the **Architecture** an **Insight** has. Also known as the **Insight** detail level, ranging from smallest atoms, over singletons to the largest landscape.

Intellectual Content: Version 1.1.8 (2020-08-13), Authored 2010-2020 by Dr. Ralf S. Engelschall, inspired by ISO/IEC/IEEE 42010:2011, IEEE 1471-2000 and The Open Group TOGAF 9
Graphical Illustration: Version 1.1.1 (2019-09-21), Copyright © 2011-2019 Dr. Ralf S. Engelschall <http://engelschall.com>, All Rights Reserved.
Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contexts only.

ARCHITECTURE FUNDAMENTALS

Architecture Maxims

TECHNISCHE UNIVERSITÄT MÜNCHEN

AF 04.1

Business Drives — BD

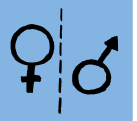
Trigger and support the business with technological feasibilities, but always understand the business domain and its demands and align your architecture accordingly.

Component Orientation — CO

Master complexity in your architecture through stringent bottom-up use of components on all scopes and zoom-levels, loose coupling between and strong cohesion within components.

Separation of Business and Technology — BT

Strictly separate the business, i.e., domain-specific, aspects from the technological, i.e., infrastructural, aspects. Furthermore, ensure the explicit visibility of domain concepts.

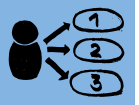


Adequate Description — AD

Provide as much stakeholder-directed architecture description as necessary, and as little as possible.

Use-Case Driven Design — UC

Design is how it works and runs, so support your customers in their daily work by directly designing your architecture along their domain-specific use-cases.

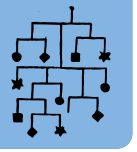


Analytical and Creative Act — AC

Recognize that every good architecture is based on both analytical engineering and creative artistic aspects.

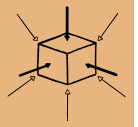
Balance Principles Against Requirements — PR

By weighing them against one another, find a reasonable balance between fundamental architecture principles and your particular non-functional requirements.



Insights through Views & Aspects — VA

Give insights into your architecture through carefully selected stakeholder-directed separate views and aspects. Express each with the most suitable graphical or textual language.



Proven Basis — PB

Never start an architecture from scratch. Instead start from proven reference architectures, patterns and templates. Even if, after some iterations, no initial content is left.

Don't Be Too Clever — TC

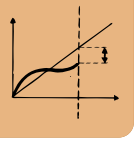
Don't be too clever or tricky, both in your higher-level architecture and lower-level design aspects.

Design for Failure Case — DF

Murphy was an architect: everything which can fail will sometime ultimatively fail. Hence, already design for the failure case (think: "pessimistic").

Continuous Compliance — CC

Continuously check through qualitative inspections and quantitative measurements whether your architecture and the non-functional requirements are followed and do not drift apart.

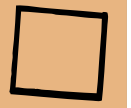


No Silver Bullet — SB

There is no "one-size-fits-all" architecture, so accept that although you should reuse proven architecture aspects as much as possible, you will always need to individualize your designs.

Simplicity Trumps — ST

Create solution parts as simple as possible and only as complex as necessary. And remember: simplicity before generality, use before reuse!



Design to Change — DC

Time changes everything, so your solution is already legacy at the first day of release. Hence, already design for its change (think: "agile").

Integration-Figure Architect — IF

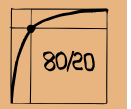
Recognize that you, the architect, are the central integrating figure, having to bridge between the business and technology spheres of people.

Stepwise Refinement — SR

Start with the "big picture" and perform a stepwise top-down refinement of your architecture by going from coarse to fine aspects.

Perfect is the Enemy of Good Enough — GE

Beware of the perfection pitfall and design your architecture only as good as necessary and not as good as ultimately possible.



Explicit Decisions — ED

Record your major architecture decisions and rationales by taking into account and back-referencing the non-functional requirements.

Eat Your Own Dog-Food — OF

Theory and practice usually differ. Hence it is vital that every architect has good hands-on experience and must both be able to craft the solution and is willing to hypothetically intensively use it himself.

Intellectual Content: Version 1.0.13 (2020-08-30) Authored 2009-2020 by Dr. Ralf S. Engelschall <http://engelschall.com>. All Rights Reserved.
Graphical Illustration: Version 1.0.8 (2020-11-28) Copyright © 2012-2020 Dr. Ralf S. Engelschall <http://engelschall.com> All Rights Reserved.
Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contexts only.