

# iSAQB® Glossary of Software Architecture Terminology

2025.1-rev0-EN-20250110





Table of Contents

Introduction ..... 1

    Personal Comments ..... 1

    Terms Can Be Referenced ..... 1

    License ..... 1

    Acknowledgements ..... 2

    Contributing ..... 2

Terms ..... 3

Translation Tables ..... 4

    English to German ..... 4

    German to English ..... 10

References and Resources ..... 16

Appendix ..... 19

    The iSAQB® e. V. Association ..... 19

    About the Authors ..... 20

    About our Cause ..... 23

## Introduction

This book contains a glossary of *software architecture terminology*.

It can aid in preparation for the iSAQB® e. V. examination *Certified Professional for Software Architecture - Foundation Level*®.

Please be aware: This glossary is **not** intended to be a primer or course book on software architecture, but just a collection of definitions and links to further information.

Furthermore, you find proposals for [translations](#) of the iSAQB® terminology, currently between English and German (and vice-versa).

Finally, this book contains numerous [references](#) to books and other resources, many of which we quoted in the definitions.



This book is work in progress.

Errors or omissions can also be reported in our issue tracker on [GitHub](#), where the authors maintain the original sources for this book.

## Personal Comments

Several of the terms contained in this book have been commented by one or several authors:



Comment (Gernot Starke)

Some terms might be especially important, or sometimes there are some subtle aspects involved. Comments like these give a personal opinion and do **not** necessarily reflect the iSAQB®.

## Terms Can Be Referenced

All terms in the glossary have unique URLs to the (free) online version of the book therefore they can be universally referenced, both from online- and print documentation.

Our URL scheme is quite simple:

- The base URL is <https://public.isaqb.org/glossary/glossary-en.html>
- We just add the prefix **#term-** in front of the term to be referenced, then the term itself, with hyphens ("-") instead of blanks.

For example our description of the term *software architecture* can be referenced (hyperlinked) with <https://public.isaqb.org/glossary/glossary-en.html#term-software-architecture>

Nearly all terms are hyperlinked with their full names, with very few examples that are referenced by their (common) abbreviations, like UML or DDD.

## License



This book is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/). The following is only a brief summary and no substitution for the real license.

The **CC BY 4.0** license means that you might:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material for any purpose, even commercially.
- The licensor cannot revoke these freedoms as long as you follow the license terms.

You must:

- Give appropriate credit,
- Provide a link to the license (<https://creativecommons.org/licenses/by/4.0/>), and
- Indicate if (and which) changes were made with respect to the original.

## Acknowledgements

Several parts of this glossary have been contributed by the following volunteers and sponsors (apart from the numerous [authors](#).)

- The definitions of about 120 terms have been donated by Gernot Starke, originally compiled for one of his [books](#).
- A number of definitions in context of system improvement and evolution was contributed by the [aim42](#) open source project.

## Contributing



### Contributions are welcome

In case find errors, omissions or typos, or want to contribute additional content - please feel free to do this via one of the following ways:

1. Open an issue in our [GitHub repository](#)
2. Fork the repository and create a pull request.
3. Write an email to the authors,

Your input is highly appreciated by the authors.

## Terms

Unresolved directive in 1-terms/0-structure.adoc - include::0-structure-EN.adoc[tags=\*\*;EN;!\*]

## Translation Tables

Here you find translations of the terms between English and German (see below) and [German-to-English](#) (next section).

Several of these terms are based in the legal and organizational foundations of the iSAQB® association (and therefore not related to software architecture).

The following translations are maintained<sup>[^generateTranslation]</sup> in a simple JSON input file, contained within this books' open source [GitHub repository](#).

<sup>[^generateTranslation]</sup>: The documentation found in <https://github.com/isaqb-org/glossary> contains all information required to generate the translation tables. Currently, only English and German are supported. The translation tables are maintained in JSON format, suggestions for improvements are highly welcome!

## English to German



Please note: This translation table is not supposed to be complete, several English terms will not be translated but preferably used in their original language (e.g. several pattern names).

English	German
Accessibility	Barrierefreiheit, Zugänglichkeit
Accountability	Rechenschaft, Verantwortlichkeit
Accreditation contract	Akkreditierungsvertrag
Accreditation fee	Akkreditierungsgebühr
Action	Maßnahme
Adaptability	Adaptierbarkeit
Adaption	Anpassung
Adequacy	Angemessenheit
Analysability	Analysierbarkeit
Approach	Ansatz
Appropriateness	Angemessenheit
Appropriateness Recognizability	Erkennbarkeit der Brauchbarkeit, Verständlichkeit
Architectural objective	Architekturziel
Architectural pattern	Architekturmuster
Architectural view	Architektursicht, Sicht
Architecture assessment	Architekturanalyse, Architekturbewertung
Architecture evaluation	Architekturbewertung, Architekturanalyse
Architecture objective	Architekturziel
Articles of association	Satzung des Vereins
Artifact	Artefakt
Aspect	Aspekt, Belang

Assessment	Bewertung, Begutachtung, Einschätzung, Untersuchung
Association	Verein, Beziehung
Attack Tree	Angriffsbäume
Authenticity	Authentifizierbarkeit
Availability	Verfügbarkeit
Bounded Context	Kontextgrenze
Building block	Baustein
Building block view	Bausteinsicht
Business	Fachlichkeit, Domäne
Business architecture	fachliche Architektur, Geschäftsarchitektur
Business context	Fachlicher Kontext
Cabinet (as methaphor for template)	Schrank (als Metapher für Template)
Capacity	Kapazität
Cash audit	Rechnungsprüfung
Cash auditor	Rechnungsprüfer
Certification authority	Zertifizierungsstelle
Certification body	Zertifizierungsstelle
Chairman	Vorsitzender
Channel	Kanal
Co-Existence	Koexistenz
Cohesion	Kohäsion, innerer Zusammenhalt
Commensurability	Angemessenheit, Messbarkeit, Vergleichbarkeit
Compatibility	Kompatibilität
Compliance	Erfüllung, Einhaltung
Component	Baustein, Komponente
Concern	Belang
Confidentiality	Vertraulichkeit
Constraint	Randbedingung, Einschränkung
Context (of a term)	Einordnung (eines Begriffes) in einen Zusammenhang
Context view	Kontextabgrenzung
Coupling	Kopplung, Abhängigkeit
Cross-cutting	Querschnittlich
Curriculum	Lehrplan
Decomposition	Zerlegung
Dependency	Abhängigkeit, Beziehung
Deployment	Verteilung
Deployment unit	Verteilungsartefakt

Deployment view	Verteilungssicht
Deputy chairman	Stellvertretender Vorsitzender
Design	Entwurf
Design approach	Entwurfsansatz, Entwurfsmethodik
Design decision	Entwurfsentscheidung
Design principle	Entwurfsprinzip
Domain	Fachdomäne, Fachlicher Bereich, Geschäftsbereich
Domain event	Fachliches Event
Domain-related architecture	fachliche Architektur
Drawing Tool	Mal-/Zeichenprogramm
Economicalness	Sparsamkeit, Wirtschaftlichkeit
Embedded	Eingebettet
Encapsulation	Kapselung
Enterprise IT architecture	Unternehmens-IT-Architektur
Estimation	Schätzung
Evaluation	Bewertung
Examination question	Prüfungsfrage
Examination rules and regulations	Prüfungsordnung
Examination sheet	Prüfungsbogen
Examination task	Prüfungsaufgabe
Examinee	Prüfling
Examiner	Prüfer
Executive board	Vorstand
Fault Tolerance	Fehlertoleranz
Fees rules and regulations	Gebührenordnung
Fitness Function	Fitnessfunktion
Functional Appropriateness	Funktionale Angemessenheit
Functional Completeness	Funktionale Vollständigkeit
Functional Correctness	Funktionale Korrektheit
Functional Suitability	Funktionale Eignung
General meeting	Mitgliederversammlung
Improvement	Verbesserung
Improvement action	Verbesserungsmaßnahme
Influencing Factor	Einflussfaktor
Information hiding principle	Geheimnisprinzip
Installability	Installierbarkeit
Integrity	Integrität
Interdependency (between design decisions)	Abhängigkeit (zwischen Entwurfsentscheidungen)



Interface	Schnittstelle
Interface description	Schnittstellenbeschreibung, Schnittstellendokumentation
Interoperability	Interoperabilität
Learnability	Erlernbarkeit
Learning goal	Lernziel
License fee	Akkreditierungsgebühr
Licensee	Lizenznehmer
Licensing agreement	Lizenzvertrag, Lizenzvereinbarung, Akkreditierungsvertrag
Local court	Amtsgericht
Maintainability	Wartbarkeit
Maturity	Reifegrad
Means for describing	Beschreibungsmittel
Means for documenting	Beschreibungsmittel
Measurability	Messbarkeit
Members' meeting	Mitgliederversammlung
message-driven	Nachrichten-zentrisch
Modeling Tool	Modellierungswerkzeug
Modifiability	Modifizierbarkeit
Modularity	Modularität
Module	Komponente, Modul, Baustein
Node	Knoten
Non-exclusive license	Einfache Lizenz
Non-profit	Gemeinnützig
Non-repudiation	Nichtabstreitbarkeit
Normal case	Normalfall
Notification	Benachrichtigung
Objective	Ziel
Operability	Bedienbarkeit
Operational processes	Betriebsprozesse (von Software)
Pattern	Muster
Pattern language	Mustersprache, Musterfamilie
Performance Efficiency	Leistungseffizienz, Performance
Perspective	Perspektive
Portability	Portierbarkeit
Principle	Prinzip, Konzept
Quality attribute	Qualitätsmerkmal, Qualitätseigenschaft
Quality characteristic	Qualitätsmerkmal, Qualitätseigenschaft

Quality feature	Qualitätsmerkmal, Qualitätseigenschaft
Rationale	Begründung, Erklärung
Real-time system	Echtzeitsystem
Recoverability	Widerherstellbarkeit
Registered trademark	Marke (gesetzlich geschützt)
Relationship	Beziehung
Relationship (kind of)	Beziehungsart
Reliability	Zuverlässigkeit
Replaceability	Austauschbarkeit
Repository	Ablage
Requirement	Anforderung
resilient	unverwüstlich, selbstwiederherstellend
Resolution	Beschluss
Resource Utilization	Ressourcenverbrauch
Responsibility	Verantwortlichkeit
responsive	reaktionsfähig
Reusability	Wiederverwendbarkeit
Rights of use	Nutzungsrecht
Runtime	Laufzeit
Runtime view	Laufzeitsicht
Security	Sicherheit
Security Goals	Schutzziele, Sachziele
Skill	Fähigkeit, Fertigkeit
Specification (of software architecture)	Beschreibung (von Softwarearchitektur)
sponsoring (board) member	materiell förderndes Mitglied
statutory	satzungsgemäß
Structure	Struktur
Task	Aufgabe
Team regulations	Arbeitsgruppenordnung
Technical context	Technischer Kontext
Term	Begriff
Testability	Testbarkeit
Thriftyness	Sparsamkeit, Wirtschaftlichkeit
Time Behaviour	Zeitverhalten
Tools	Arbeitsmittel, Werkzeug
Tools-and-material-approach	Werkzeug-Material-Ansatz
Tradeoff	Kompromiss, Abwägung, Wechselwirkung
Training provider	Schulungsanbieter
Treasurer	Schatzmeister

Ubiquitous language	Allgegenwärtige Sprache
Usability	Benutzbarkeit, Benutzerfreundlichkeit
User Error Protection	Schutz vor Fehlbedienung
User Interface Aesthetics	Ästhetik der Benutzeroberfläche
Uses relationship	Benutzt-Beziehung, Nutzungsbeziehung
View	Sicht, Architektursicht
Workflow management	Ablaufsteuerung
Working environment	Arbeitsumgebung
Working group	Arbeitsgruppe
Working group head	Arbeitsgruppenleiter

## German to English

In this section we collect the iSAQB® translation of the terms from German to English.



Please note: This translation table is not supposed to be complete, several English terms will not be translated but preferably used in their original language (e.g. many of the design pattern names).

The actual translation tables are generated from a JSON file in the [/translations](#) directory.

German	English
Abhängigkeit	Coupling, Dependency
Abhängigkeit (zwischen Entwurfsentscheidungen)	Interdependency (between design decisions)
Ablage	Repository
Ablaufsteuerung	Workflow management
Abwägung	Tradeoff
Adaptierbarkeit	Adaptability
Akkreditierungsgebühr	Accreditation fee, License fee
Akkreditierungsvertrag	Accreditation contract, Licensing agreement
Allgegenwärtige Sprache	Ubiquitous language
Amtsgericht	Local court
Analysierbarkeit	Analysability
Anforderung	Requirement
Angemessenheit	Adequacy, Appropriateness, Commensurability
Angriffsbäume	Attack Tree
Anpassung	Adaption
Ansatz	Approach
Arbeitsgruppe	Working group
Arbeitsgruppenleiter	Working group head
Arbeitsgruppenordnung	Team regulations
Arbeitsmittel	Tools
Arbeitsumgebung	Working environment
Architekturanalyse	Architecture assessment, Architecture evaluation
Architekturbewertung	Architecture assessment, Architecture evaluation
Architekturmuster	Architectural pattern
Architektursicht	Architectural view, View
Architekturziel	Architectural objective, Architecture objective
Artefakt	Artifact
Aspekt	Aspect
Aufgabe	Task
Austauschbarkeit	Replaceability

Authentifizierbarkeit	Authenticity
Barrierefreiheit	Accessibility
Baustein	Building block, Component, Module
Bausteinsicht	Building block view
Bedienbarkeit	Operability
Begriff	Term
Begründung	Rationale
Begutachtung	Assessment
Belang	Aspect, Concern
Benachrichtigung	Notification
Benutzbarkeit	Usability
Benutzerfreundlichkeit	Usability
Benutzt-Beziehung	Uses relationship
Beschluss	Resolution
Beschreibung (von Softwarearchitektur)	Specification (of software architecture)
Beschreibungsmittel	Means for describing, Means for documenting
Betriebsprozesse (von Software)	Operational processes
Bewertung	Assessment, Evaluation
Beziehung	Association, Dependency, Relationship
Beziehungsart	Relationship (kind of)
Domäne	Business
Echtzeitsystem	Real-time system
Einfache Lizenz	Non-exclusive license
Einflussfaktor	Influencing Factor
Eingebettet	Embedded
Einhaltung	Compliance
Einordnung (eines Begriffes) in einen Zusammenhang	Context (of a term)
Einschränkung	Constraint
Einschätzung	Assessment
Entwurf	Design
Entwurfsansatz	Design approach
Entwurfsentscheidung	Design decision
Entwurfsmethodik	Design approach
Entwurfsprinzip	Design principle
Erfüllung	Compliance
Erkennbarkeit der Brauchbarkeit	Appropriateness Recognizability
Erklärung	Rationale
Erlernbarkeit	Learnability

Fachdomäne	Domain
fachliche Architektur	Business architecture, Domain-related architecture
Fachlicher Bereich	Domain
Fachlicher Kontext	Business context
Fachliches Event	Domain event
Fachlichkeit	Business
Fehlertoleranz	Fault Tolerance
Fertigkeit	Skill
Fitnessfunktion	Fitness Function
Funktionale Angemessenheit	Functional Appropriateness
Funktionale Eignung	Functional Suitability
Funktionale Korrektheit	Functional Correctness
Funktionale Vollständigkeit	Functional Completeness
Fähigkeit	Skill
Gebührenordnung	Fees rules and regulations
Geheimnisprinzip	Information hiding principle
Gemeinnützig	Non-profit
Geschäftsarchitektur	Business architecture
Geschäftsbereich	Domain
innerer Zusammenhalt	Cohesion
Installierbarkeit	Installability
Integrität	Integrity
Interoperabilität	Interoperability
Kanal	Channel
Kapazität	Capacity
Kapselung	Encapsulation
Knoten	Node
Koexistenz	Co-Existence
Kohäsion	Cohesion
Kompatibilität	Compatibility
Komponente	Component, Module
Kompromiss	Tradeoff
Kontextabgrenzung	Context view
Kontextgrenze	Bounded Context
Konzept	Principle
Kopplung	Coupling
Laufzeit	Runtime
Laufzeitsicht	Runtime view
Lehrplan	Curriculum

Leistungseffizienz	Performance Efficiency
Lernziel	Learning goal
Lizenznehmer	Licensee
Lizenzvereinbarung	Licensing agreement
Lizenzvertrag	Licensing agreement
Mal-/Zeichenprogramm	Drawing Tool
Marke (gesetzlich geschützt)	Registered trademark
materiell förderndes Mitglied	sponsoring (board) member
Maßnahme	Action
Messbarkeit	Commensurability, Measurability
Mitgliederversammlung	General meeting, Members' meeting
Modellierungswerkzeug	Modeling Tool
Modifizierbarkeit	Modifiability
Modul	Module
Modularität	Modularity
Muster	Pattern
Musterfamilie	Pattern language
Mustersprache	Pattern language
Nachrichten-zentrisch	message-driven
Nichtabstreitbarkeit	Non-repudiation
Normalfall	Normal case
Nutzungsbeziehung	Uses relationship
Nutzungsrecht	Rights of use
Performance	Performance Efficiency
Perspektive	Perspective
Portierbarkeit	Portability
Prinzip	Principle
Prüfer	Examiner
Prüfling	Examinee
Prüfungsaufgabe	Examination task
Prüfungsbogen	Examination sheet
Prüfungsfrage	Examination question
Prüfungsordnung	Examination rules and regulations
Qualitätseigenschaft	Quality attribute, Quality characteristic, Quality feature
Qualitätsmerkmal	Quality attribute, Quality characteristic, Quality feature
Querschnittlich	Cross-cutting
Randbedingung	Constraint

reaktionsfähig	responsive
Rechenschaft	Accountability
Rechnungsprüfer	Cash auditor
Rechnungsprüfung	Cash audit
Reifegrad	Maturity
Ressourcenverbrauch	Resource Utilization
Sachziele	Security Goals
Satzung des Vereins	Articles of association
satzungsgemäß	statutory
Schatzmeister	Treasurer
Schnittstelle	Interface
Schnittstellenbeschreibung	Interface description
Schnittstellendokumentation	Interface description
Schrank (als Metapher für Template)	Cabinet (as methaphor for template)
Schulungsanbieter	Training provider
Schutz vor Fehlbedienung	User Error Protection
Schutzziele	Security Goals
Schätzung	Estimation
selbstwiederherstellend	resilient
Sicherheit	Security
Sicht	Architectural view, View
Sparsamkeit	Economicalness, Thriftyness
Stellvertretender Vorsitzender	Deputy chairman
Struktur	Structure
Technischer Kontext	Technical context
Testbarkeit	Testability
Unternehmens-IT-Architektur	Enterprise IT architecture
Untersuchung	Assessment
unverwüstlich	resilient
Verantwortlichkeit	Accountability, Responsibility
Verbesserung	Improvement
Verbesserungsmaßnahme	Improvement action
Verein	Association
Verfügbarkeit	Availability
Vergleichbarkeit	Commensurability
Verständlichkeit	Appropriateness Recognizability
Verteilung	Deployment
Verteilungsartefakt	Deployment unit
Verteilungssicht	Deployment view



Vertraulichkeit	Confidentiality
Vorsitzender	Chairman
Vorstand	Executive board
Wartbarkeit	Maintainability
Wechselwirkung	Tradeoff
Werkzeug	Tools
Werkzeug-Material-Ansatz	Tools-and-material-approach
Widerherstellbarkeit	Recoverability
Wiederverwendbarkeit	Reusability
Wirtschaftlichkeit	Economicalness, Thriftyness
Zeitverhalten	Time Behaviour
Zerlegung	Decomposition
Zertifizierungsstelle	Certification authority, Certification body
Ziel	Objective
Zugänglichkeit	Accessibility
Zuverlässigkeit	Reliability
Ästhetik der Benutzeroberfläche	User Interface Aesthetics

## References and Resources

This section contains references that are cited in the glossary or one of the curriculae.

### A

- [Anderson-2008] Ross Anderson, *Security Engineering - A Guide to Building Dependable Distributed Systems*, 2nd edition 2008, John Wiley & Sons. One of the most comprehensive books about information security available.

### B

- [Bachmann et al. 2000] Bachmann, F., L. Bass, et al.: *Software Architecture Documentation in Practice*. Software Engineering Institute, CMU/SEI-2000-SR-004.
- [Bass et al. 2022] Bass, L., Clements, P. und Kazman, R. (2003): *Software Architecture in Practice*. 4th edition 2022, Addison-Wesley. Although the title suggests otherwise, a quite fundamental (and sometimes abstract) book. The authors have a strong background in ultra-large scale (often military) systems - so their advice might sometimes conflict with small or lean kinds of projects.
- [Buschmann+1996] Buschmann, Frank/Meunier, Regine/Rohnert, Hans/Sommerlad, Peter: *A System of Patterns: Pattern-Oriented Software Architecture 1*, 1st edition, 1996, John Wiley & Sons.

Also known as POSA-1. Most likely the most famous and groundbreaking book on architecture patterns.

### C

- [Clements et al. 2003] Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers et al.: *Documenting Software Architectures – Views and Beyond*. Addison Wesley, 2003.
- [Cockburn 2005] Cockburn, Alistair (2005-04-01): *Hexagonal architecture*, online <https://alistair.cockburn.us/hexagonal-architecture/> (retrieved 2024-07-25)

### E

- [Evans-2004] Evans, Eric: *Domain-Driven Design: Tackling Complexity in the Heart of Software*, 1st edition, Addison-Wesley, 2004.

### F

- [Ford+2017] Neil Ford, Rebecca Parsons, Patrick Kua: *Building Evolutionary Architectures: Support Constant Change*. O'Reilly 2017

### G

- [GoF: Design-Patterns] Gamma, Erich/Helm, Richard/Johnson, Ralph/Vlissides, John M. *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st edition, 1994, Addison-Wesley, 1994.

A classic on design patterns.

- [Gang-of-Four, short: GoF] See [\[GoF: Design-Patterns\]](#)

### H

- [Hargis 2004] Hargis, Gretchen et al.: Quality Technical Information: A Handbook for Writers and Editors. Prentice Hall, IBM Press, 2004.
- [Hofmeister+2000] Hofmeister, Christine/Nord, Robert/Soni, Dilip]]]: *Applied Software Architecture*, 1st edition, Addison-Wesley, 1999
- [Hombergs 2024] Hombergs, Tom: Get Your Hands Dirty on Clean Architecture, Packt, 2nd edition 2024.

## I

- [ISO-25010] ISO/IEC 25010:2023(en) Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Product quality model. Terms and definitions online: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-2:v1:en>
- [ISO-25019] ISO/IEC 25019:2023(en) Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Quality-in-use model. Terms and definitions online: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25019:ed-1:v1:en>

## K

- [Kazman+1996] Kazman, R., Abowd, G., Bass, L., & Clements, P.: *Scenario-based analysis of software architecture*, IEEE software, 13(6), 47-55, 1996.
- [Kruchten 1995] Kruchten, P.: Architectural Blueprints – The 4-1 View Model of Architecture. IEEE Software November 1995; 12(6), p. 42-50.

## L

- [Lange 2021] Kenneth Lange: The Functional Core, Imperative Shell Pattern, online: <https://www.kennethlange.com/functional-core-imperative-shell/>
- [Lilienthal-2019] Lilienthal, Carola: *Langlebige Software-Architekturen: Technische Schulden analysieren, begrenzen und abbauen* 3rd edition, dpunkt.verlag, 2019

## M

- [Maguire 2019] Sandy Maguire: Algebra-Driven Design: Elegant Solutions from Simple Building Blocks. Leanpub, 2019.
- [Martin-2003] Martin, Robert C.: *Agile Software Development: Principles, Patterns and Practices*, Prentice Hall, 2003
- [SOLID-principles] Martin, Robert: SOLID-principles. S.O.L.I.D is an acronym for the first five object-oriented design(OOD) principles by Robert C. Martin. Some original papers have been moved around onto various locations - see [Wikipedia](#)
- [McGraw-2006] Garry McGraw, "Software Security - Building Security In", Addison-Wesley 2006 Covering the whole process of software design from a security perspective by the means of risk management, code reviews, risk analysis, penetration testing, security testing abuse case development.

## P

- [Parnas-1972] Parnas, David: *On the criteria to be used in decomposing systems into modules*", Communications of the ACM, volume 15, issue 12, Dec 1972. One of the most influential articles ever written in software engineering, introducing encapsulation and modularity. Thank you, David!

**R**

- [RMIAS-2013] Yulia Cherdantseva, Jeremy Hilton, A Reference Model of Information Assurance & Security, 2013 Eight International Conference on Availability, Reliability and Security (ARES), DOI: 10.1109/ARES.2013.72, <http://users.cs.cf.ac.uk/Y.V.Cherdantseva/RMIAS.pdf> Conference Paper of Yulia Cherdantseva and Jeremy Hilton describing the RMIAS.
- [Rozanski & Woods 2011] Eoin Woods and Nick Rozanski: Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives. 2nd edition 2011, Addison-Wesley. Presents a set of architectural viewpoints and perspectives.

**S**

- [Schmidt, Douglas C/Stal, Michael/Rohnert, Hans/Buschmann, Frank.] Pattern-Oriented Software Architecture, volume 2: *Patterns for Concurrent and Networked Objects*, Wiley & Sons, 2000
- [Schneier, Bruce] Applied Cryptography, 2nd Edition 1996, John Wiley & Sons. Comprehensive survey of modern cryptography.
- [Sperber+2024] Michael Sperber, Stefan Wehr: Datenmodellierung mit Summen und Produkten, 2024. <https://funktionale-programmierung.de/2024/11/25/sums-products.html>. (English translation: Data Modeling with Sums and Products, 2024. <https://funktionale-programmierung.de/2024/11/25/sums-products-english.html>)
- [Starke 2019] Starke, G. Effektive Software-Architekturen - Ein praktischer Leitfaden. 9. Auflage 2019, Carl Hanser Verlag.

**T**

- [Tanenbaum+2016] Andrew Tanenbaum, Maarten van Steen: Distributed Systems, Principles and Paradigms, 2016. <https://www.distributed-systems.net/>
- [Tornhill-2015] Adam Tornhill: Your Code as a Crime Scene. Use Forensic Techniques to Arrest Defects, Bottlenecks, and Bad Design in Your Programs. Pragmatic Programmers, 2015. <https://www.adamtornhill.com>

**Y**

- [Yorgey 2012] Brent A. Yorgey, Monoids: Theme and Variations. Proceedings of the 2012 Haskell Symposium, September 2012 <https://doi.org/10.1145/2364506.2364520>

## Appendix

### The iSAQB® e. V. Association

The International Software Architecture Qualification Board (iSAQB® e. V., <http://isaqb.org>) is a non-profit organization with members from industry, development and consulting firms, education, academia and other organizations.

It is established as an *association* (e. V.) according to German law with the following objectives:

- Creating and maintaining consistent **curricula** for software architects.
- **Defining certification examinations** based upon the various CPSA® curricula
- **Ensuring high quality of teaching** for software architects
- Ensuring a high quality of its software architecture certifications

iSAQB® defines and prescribes training and examination regulations, but does not carry out any training or examinations itself. iSAQB® trainings are carried out by (licensed) training and examination organizations.

iSAQB® monitors and audits the quality of these trainings and all associated processes (e.g. certification procedures).

## About the Authors

### Gernot Starke

Dr. Gernot Starke ([INNOQ Fellow](#)) is co-founder and avid user of the (open source) [arc42](#) documentation template. For more than 20 years he works as software architect, coach and consultant, conquering the challenges of creating effective software architectures for clients from various industries.

In 2008 Gernot co-founded the International Software Architecture Qualification Board ([iSAQB® e. V.](#)) and since then supports it as an active member.

Gernot has authored several (German) books on software architecture and related topics and initiated this glossary.

He lives in Cologne with his wife (*Cheffe Uli*).

### Ulrich Becker

Ulrich Becker works as principal consultant at [Method Park](#), focussing on software architecture and application lifecycle management.

Ulrich studied computer science at the University of Hamburg and the University of Erlangen-Nürnberg. He received his PhD from the University of Erlangen-Nürnberg in 2003 for his work on model-based distribution configuration. He then became group leader for the adaptive system software group at [Fraunhofer IIS](#).

Since 2005 Ulrich works as a trainer, consultant and coach at Method Park where he supports his clients in improving their development processes and methods. Most of his clients are from the automotive industry or other heavily regulated industries.

Ulrich is a founding member of [iSAQB® e. V.](#) where he contributes to the foundation level and advanced level working groups. He lives in Erlangen with his family.

### Matthias Bohlen

Matthias Bohlen, [independent expert](#) for effective product development, started his career as a software developer in 1980. He wrote compilers for the MC68020 processor by Motorola which was quite a revolutionary device in those days where there was no IBM PC, yet. And the compilers really sold well.

Since then, Matthias has worked with countless software teams, helping them to get working software out the door without losing their mind. This is what he still does today.

Matthias is an active member of the [International Software Architecture Qualification Board](#), writes [a blog](#), is being known in the Lean/Agile field, and speaks at conferences for software development.

### Phillip Ghadir

Member of the board of INNOQ Deutschland GmbH. Since many years, Phillip consults clients from various industries in topics around software-architecture, technology and development. He co-founded the iSAQB® and regularly conducts trainings on software architecture.

### Carola Lilienthal

Dr. Carola Lilienthal is software architect at and co-founder of the [WPS Workplace-Solutions](#). For 20 years she has been working as a developer, project manager, coach, consultant and architect. Carola was an

early adopter of Domain-Driven Design and agile movement and has successfully worked for numerous clients from various domains, mainly finance, insurance and logistics.

Since 2003, she has been analyzing software systems in Java, C++, C#, PHP, ABAP and gives advice to development teams how to improve the sustainability of their code. Carola speaks regularly on conferences and has written various articles as well as a book on sustainable software architecture.

Since 2008 Carola has been supporting the International Software Architecture Qualification Board ([iSAQB® e. V.](#)) as an active member.

### **Mahbouba Gharbi**

Mahbouba Gharbi is CIO of iTech Progress, book author and conference speaker.

Several years ago Mahbouba became president of the iSAQB. She lives in Mannheim with her family.

### **Simon Kölsch**

Simon Kölsch works as a senior consultant at INNOQ with a focus on web architecture and security.

Simon is enthusiastic about solutions beyond the classical monolithic enterprise application, covering the architecture of distributed systems and their infrastructure, logging and monitoring.

He is not committed to one specific technology or programming language, but has a strong JVM background.

### **Alexander Lorz**

Dr.-Ing. Alexander Lorz is a freelance software architecture trainer, IT consultant and developer. His first contact with IT systems dates back to the mid-1980s, and since then he has refused to give up his fascination for the science and craftsmanship of developing complex systems.

As an active member of the International Software Architecture Qualification Board ([iSAQB® e. V.](#)) and the Foundation Level Working Group he contributes to the evolution of the foundation curriculum.

### **Michael Mahlberg**

Michael Mahlberg runs his own [method consultancy](#) in Germany and spends most of his time supporting clients in their quest for more effective ways to work. Mostly by applying lean and agile concepts.

Running his own computer and software related companies since he was 18, he quickly came to realize that software architecture and (development) processes are in a way timeless aspects of the craft.

Nowadays a lot of his work focuses on processes and human interactions – a field in which he engages both professionally as well as pro bono (for example he is one of the people who started and run the [Limited WIP Society Cologne](#)).

Michael's architectural work therefore tends to be dealing with the impact(s) and implications of architectural and process decisions *on each other* and the relative optimization strategies.

### **Andreas Rausch**

Prof. Dr. Andreas Rausch is head of the chair for Software Systems Engineering at Clausthal University of Technology.

He received his doctorate in 2001 at the Technical University of Munich, at the chair of Prof. Dr. Manfred Broy. His main research interests in the field of software systems engineering are software architectures, model-based software development and process models. He has published more than 70 international papers in these areas.

### **Roger Rhoades**

Roger Rhoades is founder of [Albion](#), a training and consulting company in Germany.

Roger has over 25 years of practical experience in the field of enterprise, business, and software architecture as well as management of international teams and projects. This practical experience is integrated into his training courses to ensure that class participants not only understand the theoretical content, but also the real-world challenges of its implementation.

Since 2012, Roger regularly presents at international conferences (e.g. EAMKon, Lean42 EAM, IT Strategy and Governance).

Since 2014, Roger has been an active member of the International Software Architecture Qualification Board ([iSAQB® e. V.](#)). He actively supports the evolution of the foundation and advanced curricula, exam questions, and case studies in addition to the iSAQB® glossary.

### **Sebastian Fichtner**

Founder of [flowtoolz.com](#). App engineer and consultant. Started coding in 1995. Passionate about architecture ever since. Does original apps, open source and projects for various clients. Loves Apple platforms and the language Swift.



## About our Cause



# ELECTRONIC FRONTIER FOUNDATION

[eff.org](https://eff.org)

All royalties of this book are donated to the EFF. By paying for this book, you support their cause:

"The **Electronic Frontier Foundation** is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

Even in the fledgling days of the Internet, EFF understood that protecting access to developing technology was central to advancing freedom for all. In the years that followed, EFF used our fiercely independent voice to clear the way for open source software, encryption, security research, file sharing tools, and a world of emerging technologies.

Today, EFF uses the unique expertise of leading technologists, activists, and attorneys in our efforts to defend free speech online, fight illegal surveillance, advocate for users and innovators, and support freedom-enhancing technologies.

Together, we forged a vast network of concerned members and partner organizations spanning the globe. EFF advises policymakers and educates the press and the public through comprehensive analysis, educational guides, activist workshops, and more. EFF empowers hundreds of thousands of individuals through our Action Center and has become a leading voice in online rights debates.

EFF is a donor-funded US 501(c)(3) nonprofit organization that depends on your support to continue fighting for users."

(Quote from [eff.org/about](https://eff.org/about))