

Curriculum for
CPSA Certified Professional for
Software Architecture®

– Advanced Level –

**Module:
ADOK**

Architecture Documentation



Version 1.5 (February 2015)

**© (Copyright), International Software Architecture Qualification Board e. V.
(iSAQB e. V.) 2012**

The curriculum may only be used subject to the following conditions:

1. You wish to obtain the CPSA Certified Professional for Software Architecture Advanced Level[®] certificate. For the purpose of obtaining the certificate, it shall be permitted to use these text documents and/or curricula by creating working copies for your own computer. If any other use of documents and/or curricula is intended, for instance for their dissemination to third parties, for advertising etc., please write to contact@isaqb.org to enquire whether this is permitted. A separate license agreement would then have to be entered into.
2. If you are a trainer, training provider or training organizer, it shall be possible for you to use the documents and/or curricula once you have obtained a usage license. Please address any enquiries to contact@isaqb.org. License agreements with comprehensive provisions for all aspects exist.
3. If you fall neither into category 1 nor category 2, but would like to use these documents and/or curricula nonetheless, please also contact the iSAQB e. V. by writing to contact@isaqb.org. You will then be informed about the possibility of acquiring relevant licenses through existing license agreements, allowing you to obtain your desired usage authorizations.

We stress that, as a matter of principle, this curriculum is protected by copyright. The International Software Architecture Qualification Board e. V. (iSAQB[®] e. V.) has exclusive entitlement to these copyrights. The abbreviation "e. V." is part of the iSAQB's official name and stands for "eingetragener Verein" (registered association), which describes its status as a legal person according to German law. For the purpose of simplicity, iSAQB e. V. shall hereafter be referred to as iSAQB without the use of said abbreviation.

Table of contents

0	<u>INTRODUCTION: GENERAL INFORMATION ON THE ISAQB ADVANCED LEVEL</u>	5
0.1	WHAT DOES AN ADVANCED LEVEL MODULE TEACH?	5
0.2	WHAT CAPABILITIES DO GRADUATES OF THE ADVANCED LEVEL (CPSA-A) ACQUIRE?.....	5
0.3	REQUIREMENTS FOR CPSA-A CERTIFICATION	5
1	<u>INTRODUCTION TO THE ARCHITECTURE DOCUMENTATION MODULE</u>	6
1.1	STRUCTURE OF THE CURRICULUM FOR ARCHITECTURE DOCUMENTATION, AND THE RECOMMENDED TIME ALLOCATION	6
1.2	DURATION, DIDACTICS AND OTHER DETAILS.....	6
1.3	PREREQUISITES FOR THE "ARCHITECTURE DOCUMENTATION" MODULE	6
1.4	STRUCTURING OF THE MODULE UNITS BASED ON LEARNING OBJECTIVES	7
1.5	SUPPLEMENTARY INFORMATION, TERMS AND TRANSLATIONS	7
1.6	CREDIT POINTS FOR THIS TRAINING MODULE	7
2	<u>INTRODUCTION TO THE ISAQB CERTIFICATION PROGRAM</u>	8
2.1	TERMS AND CONCEPTS.....	8
2.2	LEARNING OBJECTIVES	8
3	<u>BASIC TERMS USED IN SOFTWARE ARCHITECTURE DOCUMENTATION</u>	9
3.1	TERMS AND CONCEPTS.....	9
3.2	LEARNING OBJECTIVES	9
4	<u>DOCUMENTATION AND PROCEDURES</u>	10
4.1	TERMS AND CONCEPTS.....	10
4.2	LEARNING OBJECTIVES	10
5	<u>COMPONENT ELEMENTS OF ARCHITECTURE DOCUMENTATION</u>	11
5.1	TERMS AND CONCEPTS.....	11
5.2	LEARNING OBJECTIVES	11
6	<u>TOOLS</u>	13
6.1	TERMS AND CONCEPTS.....	13
6.2	LEARNING OBJECTIVES	13
7	<u>REVIEWING DOCUMENTATION</u>	14

7.1	TERMS AND CONCEPTS.....	14
7.2	LEARNING OBJECTIVES	14
8	<u>EXAMPLES OF DOCUMENTATION FOR SOFTWARE ARCHITECTURES</u>	15
8.1	TERMS AND CONCEPTS.....	15
8.2	LEARNING OBJECTIVES	15
8.3	REFERENCES	15
9	<u>SOURCES AND REFERENCES ON ARCHITECTURE DOCUMENTATION</u>	16

0 Introduction: General information on the iSAQB Advanced Level

0.1 What does an Advanced Level Module teach?

- The iSAQB Advanced Level offers modular training in three areas of competence with flexible academic approaches. It considers individual leanings and focuses.
- The certification is achieved by writing a term paper. Experts designated by the iSAQB perform the assessment and administer the oral examination.

0.2 What capabilities do graduates of the Advanced Level (CPSA-A) acquire?

CPSA-A graduates are capable of the following:

- Independent and method-based design of medium- to large-scale IT systems.
- Responsibility for technology and content of IT systems of medium to high criticality.
- Development, design and documentation of measures for achieving non-functional requirements. Support of development teams in implementation of these measures.
- Control and implementation of architecture-related communication in medium to large development teams.

0.3 Requirements for CPSA-A certification

- Successful training and certification as a CPSA-F (Certified Professional for Software Architecture, Foundation Level)
- At least three years of full-time career experience in the IT sector, with participation in design and development of at least two different IT systems
 - Exceptions are possible on application (such as participation in open source projects)
- Training and advanced qualification within the framework of iSAQB Advanced Level courses comprising at least 70 credit points from all three different areas of competence (details in section 1.6).
 - Existing certifications can be accredited for these credit points on application. The list of current certificates accredited as credit points is available on the iSAQB website.
 - Other training and advanced qualifications can also be accredited on application to the iSAQB if they are relevant for software architecture. This will be decided on an individual basis by the iSAQB Advanced Level working group.
- Successful completion of the CPSA-A certification examination.

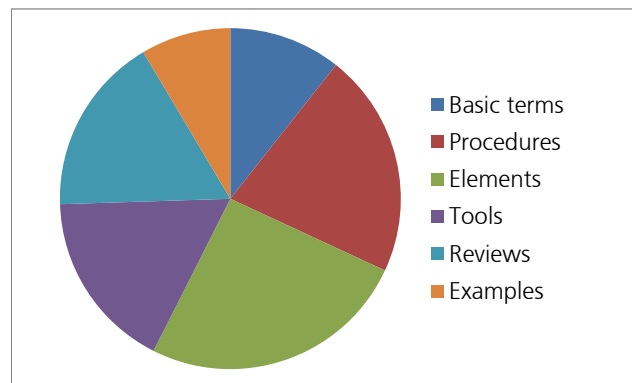


1 Introduction to the Architecture Documentation module

1.1 Structure of the curriculum for architecture documentation, and the recommended time allocation

Basic terms used in software architecture documentation	(min. 1.25h)
Documentation and procedures	(min. 2.5h)
Component elements of architecture documentation	(min. 3h)
Tools	(min. 2h)
Reviewing documentation	(2h)
Examples of documentation for software architectures	(min. 2h)

(All times include exercises.)



1.2 Duration, didactics and other details

The times stated below are recommendations. The minimum duration of a course on web architecture should be 3 days, but it can be longer. Providers can differ with respect to the duration, methodology, type and structure of the exercises as well as the detailed course outline. In particular, the curriculum leaves the type of examples and exercises completely open

The module can be attended independent of a CPSA-F certification.

1.3 Prerequisites for the “Architecture Documentation” module

Participants **should** have the following knowledge and/or experience:

- Fundamentals of describing architectures with the aid of different views, general concepts, design decisions, conditions & constraints etc., as taught at the CPSA-F Foundation Level.
- Experience in the creation and maintenance of technical documentation of software is desirable, in particular documentation of the architecture of software systems or software-intensive systems.

In addition, the following knowledge is **advantageous** for an understanding of some of the concepts covered:

- Knowledge of typical challenges in the creation and maintenance of technical documentation:
 - Selection of appropriate documentation structures, notations and result types (stakeholder orientation)

- Handling of large volumes of documentation (in particular existing or outdated documentation).
- Selection, configuration and implementation of tool chains (what resources are available for appropriate creation and maintenance of documentation),
- Versioning of documents (models and texts)
- Documentation in teams, collaborative creation and maintenance
- Content-related and formal reviews of documentation.

1.4 Structuring of the module units based on learning objectives

The individual units of the curriculum are specified using the following structure:

- **Terms/concepts:** The key terms associated with this topic.
- **Lesson/practical exercise duration:** Defines the minimum time that must be allocated to the teaching and practical exercises for this topic in an accredited course.
- **Learning objectives:** Describe the content to be taught, including the associated key terms and concepts.

This section therefore also outlines the skills to be acquired in corresponding courses. The learning goals are differentiated in the following categories and sub-chapters:

- What should participants **be able to** do? Participants should be able to use this content independently after the course. This content is covered during the course by exercises and is part of the architecture documentation module examination and/or the final examination of the iSAQB Advanced Level.
- What should participants **understand**? This content can be tested in the architecture documentation module examination.
- What should participants **know**? This content (terms, concepts, methods, practices, etc.) can support understanding or motivate the subject. This content is not part of the examinations and will be mentioned in courses, but not necessarily taught in detail.
- **References:** References to secondary literature, standards or other sources. A detailed list of books and other sources is available on the iSAQB website under “Specialized sources”.

1.5 Supplementary information, terms and translations

If necessary for understanding of the curriculum, we have included technical terms in the iSAQB glossary, with definitions and, as needed, translations of the original literature.

1.6 Credit points for this training module

Courses licenced by the iSAQB e. V. based on this curriculum result in 20 credit points in methodical competence.

2 Introduction to the iSAQB certification program

Duration: 15 min (optional)	Practice time: none
-----------------------------	---------------------

This section is not relevant for the examination. This section can be omitted if participants are already CPSA-F certified.

2.1 Terms and concepts

iSAQB, Advanced Level certification and prerequisites for the same.

2.2 Learning objectives

Participants become familiar with the iSAQB certification program and the corresponding examinations and examination procedures.

2.2.1 What should participants know?

- The iSAQB as an association
- Advanced Level as opposed to other levels
- Constraints and procedures of the iSAQB certification program

3 Basic terms used in software architecture documentation

Duration: 55 minutes	Practical exercises: approx. 20 minutes
----------------------	---

3.1 Terms and concepts

Software architecture, documentation, stakeholder, notation, procedure model.

3.2 Learning objectives

3.2.1 What should the participants be able to do?

- Describe and convincingly explain the usefulness and the various goals of architecture documentation.
- Explain the principal users and the significance of architecture documentation for different stakeholders.
- Differentiate between work results from the architecture documentation and results from other disciplines (e. g., requirements management, project management, ...).
- Cite typical notations for architecture documentation.
- Cite specific procedure models and their “contribution” in terms of architecture documentation.

3.2.2 What should the participants understand?

- That architecture documentation is not an end in itself.
- That architecture documentation plays a supporting role in the development of a solution, and in communication of the solution to the team and others (e. g., customers).
- That different procedure models impose detailed to absolutely no requirements in terms of specific results.
- That significantly different documentation effort is necessary, depending on the procedure used and the conditions and constraints of the project.
- That UML is well established in conjunction with architecture documentation, but is not absolutely necessary.

3.2.3 What should the participants know?

Not applicable.

4 Documentation and procedures

Duration: 90 minutes	Practical exercises: approx. 60 minutes
----------------------	---

4.1 Terms and concepts

Procedure model, role, artefact, iterative/incremental, life cycle, agile.

4.2 Learning objectives

The participants can define/develop an appropriate procedure for the documentation for a given situation.

4.2.1 What should the participants be able to do?

- Create architecture documentation for the following situations:
 - Development of new systems
 - Further development of existing systems with documentation available
 - Retrospective documentation of existing, undocumented systems
 - Documentation in exceptional situations (low budget, limited time available, limited information sources available etc.).
- Determine the target groups and the purposes of the documentation.
- Select the component elements to be created, in line with the needs.
- Be able to argue the advantages of fixed layouts and structures for (technical) documentation.

4.2.2 What should the participants understand?

- That documentation is ideally created in parallel with the architecture and development, not afterwards.
- That agility and documentation are not mutually exclusive.

4.2.3 What should the participants know?

- The role of architecture documentation in various specific procedure models
- Different proposals and templates for standard layouts (e. g., arc42, SEI).

5 Component elements of architecture documentation

Duration: 90 minutes	Practical exercises: approx. 90 minutes
----------------------	---

5.1 Terms and concepts

Views, decisions, general concepts and topics, interfaces, structures, conditions & constraints, risks, quality objectives.

5.2 Learning objectives

5.2.1 What should the participants be able to do?

- Independently document the architecture of medium to large IT systems, both in parallel with the development and where necessary also retrospectively.
- Further develop and maintain existing architecture documentation, while at the same time systematically improving the quality of the documentation.
- Develop or select appropriate architecture documentation types and structures for specific medium to large systems.
- Identify and appropriately document factors influencing a software architecture, e. g.,
 - Conditions and constraints
 - Quality objectives
 - Technical risks.
- Document architecture decisions in a traceable manner.
- Produce different views of a software architecture with the aid of appropriate description methods and tools:
 - Explain and document different aspects of module views (presentation with hierarchical refinement, documentation of recurring structures or patterns, description of internal interfaces, correlation of modules to source code).
 - Run-time scenarios, in particular for fulfilment of the main system functions, description of incomplete scenarios, description across different levels of refinement
 - Notations (e. g., UML, SysML etc.) for the description of views.
- Describe interfaces.
- Prepare technical and/or general concepts in an appropriate format for the target group(s).
- Check and ensure consistency between different component elements of the architecture documentation.
- Structure component elements in documents.

5.2.2 What should the participants understand?

- That to achieve an understandable software architecture, documentation of the solution alone is not sufficient. Requirements and justifications of relevance to the architecture must also be documented.

5.2.3 What should the participants know?

- IEEE 1471 (“Recommended Practice for Architecture Description of Software-Intensive Systems”)
- Alternative proposals for specific views, for example:
 - the Rational Unified Process (4+1)
 - the metamodel / view model from the Software Engineering Institute
 - arc42.

- Graphical notations for views
- FMC (Fundamental Modelling Concepts)
- RM-ODP (Reference Model of Open Distributed Processing).

6 Tools

Duration: 90 minutes	Practical exercises: approx. 30 minutes
----------------------	---

6.1 Terms and concepts

Analog and digital tools, modelling tools, tool chain.

6.2 Learning objectives

6.2.1 What should the participants be able to do?

- Select the right tools for the different activities:
 - Creation and maintenance of component elements of architecture documentation
 - Element management
 - Communication of contents with the support of the component elements.
- Use analog and digital documentation tools appropriately for specific situations and needs, and protect the results.
- Select a complete tool chain for architecture documentation based on specific requirements and taking conditions, constraints and other influencing factors into account in a reproducible manner.

6.2.2 What should the participants understand?

- That tools can provide support for a wide range of different activities associated with architecture documentation.
- That documentation, as with source code, should be versioned and subject to release management. For each definitive status of the software it should be possible to restore or generate the associated status of the (architecture) documentation.

6.2.3 What should the participants know?

- The respective strengths, weaknesses and typical challenges when using and integrating commonly available tool categories, in particular:
 - Wikis
 - Modelling tools
 - Drawing software
 - Word processors
 - Version management software
 - Others (e. g., blogs, issue trackers etc.).
- Specific complete tool chains that address all activities associated with architecture documentation.

7 Reviewing documentation

Duration: 75 minutes	Practical exercises: approx. 45 minutes
----------------------	---

7.1 Terms and concepts

Review, checklists, question catalogues.

7.2 Learning objectives

7.2.1 What should the participants be able to do?

- Differentiate between content-related and formal reviews of architecture documentation.
- Differentiate between a review of documentation and a review of the architecture.
- Explain different objectives when reviewing documentation.
- Select, plan and carry out appropriate review steps.
- Define the objectives for a review and communicate them as binding objectives to the reviewers.
- Prepare checklists and question catalogues for reviews.
- During review sessions:
 - In the role of the author, respond appropriately to critical comments.
 - In the role of a reviewer, provide constructive feedback to authors.
 - In the role of the moderator, manage a specialist review.

7.2.2 What should the participants understand?

- That reviewing the documentation's suitability for use is an essential success factor.

7.2.3 What should the participants know?

- IEEE 1028-2008 "IEEE Standard for Software Reviews and Audits"
- Various examples of architecture documentation.

8 Examples of documentation for software architectures

Duration: 60 minutes	Practical exercises: None
----------------------	---------------------------

This section is not relevant for the examination.

8.1 Terms and concepts

Within each accredited training course, at least one example of a documented software architecture must be presented.

The nature and characteristics of the presented examples can vary depending on the training and/or the interests of the participants, and are not specified by the iSAQB.

8.2 Learning objectives

Seeing the documentation of one or more real software architectures and discussing its advantages and disadvantages.

8.2.1 What should the participants be able to do?

Explain how methodical documentation has been implemented in the example(s), and analyse the strengths and weaknesses of the presented example(s).

8.2.2 What should the participants understand?

Not applicable.

8.2.3 What should the participants know?

Further examples of (where possible) realistic architecture documentation.

8.3 References

None. Training providers are responsible for the selection and description of examples.

9 Sources and references on architecture documentation

This section contains references that are referred to in whole or in part in the curriculum.

B

[Bachmann 2000]

Bachmann, F., L. Bass, et al.: Software Architecture Documentation in Practice. Software Engineering Institute, CMU/SEI-2000-SR-004.

[Bass+2003]

Bass, L., Clements, P. und Kazman, R. (2003): Software Architecture in Practice. Addison-Wesley, Reading, Mass

C

[Clements+2003]

Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers et al: Documenting Software Architectures – Views and Beyond. Addison Wesley, 2003.

H

[Hargis+2004]

Hargis, Gretchen et. al: Quality Technical Information: A Handbook for Writers and Editors. Prentice Hall, IBM Press, 2004.

K

[Kruchten 1995]

Kruchten, P.: Architectural Blueprints – The 4-1 View Model of Architecture. IEEE Software November 1995; 12(6), p. 42-50.

S

[Starke 2011]

Starke, G. (2011): Effektive Software-Architekturen - Ein praktischer Leitfaden. 5. Auflage 2011, Carl Hanser Verlag, München.

Z

[Zörner 2012]

Zörner, Stefan: Softwarearchitekturen dokumentieren und kommunizieren, Hanser Verlag 2012