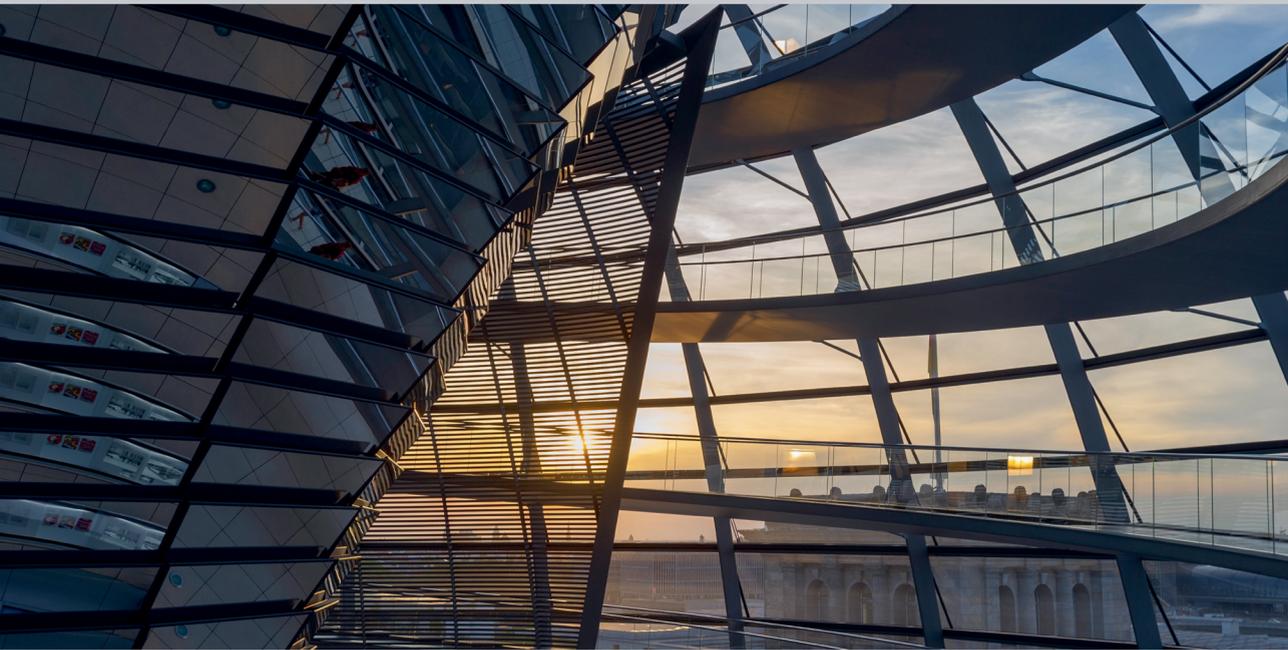


SOFTWARE ARCHITECTURE FOUNDATION

GERNOT STARKE & ALEXANDER LORZ



CPSA-F[®] EXAM PREPARATION

Software Architecture Foundation

Other publications by Van Haren Publishing

Van Haren Publishing (VHP) specializes in titles on Best Practices, methods and standards within four domains:

- IT and IT Management
- Architecture (Enterprise and IT)
- Business Management and
- Project Management

Van Haren Publishing is also publishing on behalf of leading organizations and companies: ASL[®]BiSL Foundation, BRMI, CA, Centre Henri Tudor, CATS CM, Gaming Works, IACCM, IAOP, IFDC, Innovation Value Institute, IPMA-NL, ITSq[®], NAF, KNVI, PMI-NL, PON, The Open Group, The SOX Institute.

Topics are (per domain):

IT and IT Management

ABC of ICT
ASL[®]
CMMI[®]
COBIT[®]
e-CF
ISO/IEC 20000
ISO/IEC 27001/27002
ISPL
IT4IT[®]
IT-CMF[™]
IT Service CMM
ITIL[®]
MOF
MSF
SABSA
SAF
SIAM[™]
TRIM
VeriSM[™]

Enterprise Architecture

ArchiMate[®]
GEA[®]
Novius Architectuur
Methode
TOGAF[®]

Project Management

A4-Projectmanagement
DSDM/Atern
ICB / NCB
ISO 21500
MINCE[®]
M_o_R[®]
MSP[®]
P3O[®]
PMBOK[®] Guide
Praxis[®]
PRINCE2[®]

Business Management

BABOK[®] Guide
BiSL[®] and BiSL[®] Next
BRMBOK[™]
BTF
CATS CM[®]
DID[®]
EFQM
eSCM
IACCM
ISA-95
ISO 9000/9001
OPBOK
SixSigma
SOX
SqEME[®]

For the latest information on VHP publications, visit our website: www.vanharen.net.

Software Architecture Foundation

CPSA-F® Exam Preparation

Gernot Starke, Alexander Lorz



Colophon

Title	Software Architecture Foundation CPSA-F® Exam Preparation
Authors:	Gernot Starke, Alexander Lorz
Publisher:	Van Haren Publishing, 's-Hertogenbosch-NL www.vanharen.net .
ISBN Hard copy:	978 94 018 0740 1
ISBN eBook (pdf):	978 94 018 0741 8
ISBN ePUB:	978 94 018 0742 5
Edition:	First edition, first impression, May 2021
Layout and Design:	Coco Bookmedia, Amersfoort-NL
Copyright:	© Van Haren Publishing, 2021

Nothing from this publication may be reproduced, recorded in an automated database or published on or via any medium, either electronically, mechanically, through photocopying or any other method, without prior written permission from the publisher.

This publication was produced with the utmost care and attention. Nevertheless, the text may contain errors. The publisher and the authors are not liable for any errors and/or inaccuracies in this text.

Trademark notices:

CPSA and iSAQB are trademarks of iSAQB e.V.

Contents

QUICK OVERVIEW	XI
-----------------------------	-----------

FOREWORD BY MIRKO HILLERT	XII
--	------------

FOREWORD BY STEFAN TILKOV	XIII
--	-------------

FOREWORD BY PETER HRUSCHKA	XIV
---	------------

PART I: INTRODUCTION	1
-----------------------------------	----------

About Software Architecture	1
About this book	2
About iSAQB	4
About certification	6
iSAQB Foundation	7
iSAQB examination overview	9
Preparation for the Foundation Level examination	14

PART II: CPSA-F LEARNING GOALS EXPLAINED	17
---	-----------

How learning goals are explained	17
List of learning goals	17

CPSA-F CHAPTER 1: BASIC CONCEPTS OF SOFTWARE ARCHITECTURE	19
--	-----------

LG 1-1: Definitions of software architecture	19
LG 1-2: Goals and benefits of software architecture	23
LG 1-3: Software architecture in the software lifecycle	25
LG 1-4: Software architects' tasks and responsibilities	28
LG 1-5: Software architects and other stakeholders	34
LG 1-6: Development approach and software architecture	37
LG 1-7: Short- and long-term goals	39
LG 1-8: Explicit statements versus implicit assumptions	41
LG 1-9: Software architects in the greater architectural context	43
LG 1-10: Types of IT systems	45
LG 1-11: Distributed systems	47

CPSA-F CHAPTER 2: DESIGN AND DEVELOPMENT OF SOFTWARE ARCHITECTURES . . . 49

LG 2-1: Approaches and heuristics for architecture development.....	49
LG 2-2: Design software architectures.....	56
LG 2-3: Influencing factors and constraints	64
LG 2-4: Cross-cutting concepts	68
LG 2-5: Patterns	70
LG 2-6: Design principles	79
LG 2-7: Managing dependencies	91
LG 2-8: Achieve quality requirements	99
LG 2-9: Design and define interfaces	102

CPSA-F CHAPTER 3: SPECIFICATION AND COMMUNICATION OF SOFTWARE ARCHITECTURES 105

LG 3-1: Quality requirements for technical documentation	105
LG 3-2: Describe and communicate software architectures	108
LG 3-3: Models and notations to describe software architectures.....	111
LG 3-4: Architectural views.....	118
LG 3-5: Context view	123
LG 3-6: Document cross-cutting concepts.....	126
LG 3-7: Describe interfaces	129
LG 3-8: Document architectural decisions	131
LG 3-9: Documentation as written communication.....	133
LG 3-10: Resources and tools for documentation.....	135

CPSA-F CHAPTER 4: SOFTWARE ARCHITECTURE AND QUALITY 137

LG 4-1: Quality models and quality characteristics.....	137
LG 4-2: Clarify quality requirements.....	141
LG 4-3: Qualitative analysis	145
LG 4-4: Quantitative evaluation	148

CPSA-F CHAPTER 5: EXAMPLES OF SOFTWARE ARCHITECTURE 151

LG 5-1: From requirements to solution.....	151
LG 5-2: Rationale of a technical implementation	152

PART III: BACKGROUND 153

Additional patterns	153
Appropriateness: it always depends.	159

PART IV: MOVING ON 161

 Advanced Level modules 161

 Advanced Level certification 166

PART V: APPENDICES 169

APPENDIX A: ABOUT THE AUTHORS 170

APPENDIX B: GLOSSARY OF TERMS 172

APPENDIX C: REFERENCES AND FURTHER READING 201

INDEX 207

Quick overview

This book covers everything you need to master the iSAQB® Certified Professional for Software Architecture - Foundation Level (CPSA-F®) certification.

This internationally renowned education and certification schema defines various learning paths for practical software architects.

This book concentrates on the Foundation Level examination. It explains and clarifies all 40+ learning goals of the CPSA-F® curriculum. In addition, you will find a step-by-step preparation guide for the examination.

Please beware: This book is *not* meant as a replacement for existing software architecture books and courses, but strongly focuses on explaining and clarifying the iSAQB CPSA-F foundation.

Foreword by Mirko Hillert

As a sub-discipline of software engineering, software architecture acquired an increasing importance in the 1990s, both in industrial and academic environments. This IT discipline has developed rapidly since then, and nowadays there is consensus among experts that it is the architecture of a software system that decisively determines its quality. In the course of this steadily increasing importance of software architectures, the specific occupational profile of software architects within development teams evolved. This complex role can hardly be mastered today without profound knowledge of common technologies and frameworks as well as methodical and communicative skills.

Ever since its foundation as a non-profit organization in 2008, the International Software Architecture Qualification Board (iSAQB®) has been working steadily to establish an internationally recognized standard in education and training of software architects. And with great success: Over 24,000 IT professionals worldwide have already been certified within the scope of the Certified Professional for Software Architecture (CPSA®) certification scheme developed by the iSAQB.

The new book by Dr. Alexander Lorz and Dr. Gernot Starke offers a comprehensive compendium of basic knowledge in modern software architecture to anyone who wants to embark on the career path of a software architect. In excellent didactic and content-related manner, it conveys all the necessary fields of knowledge that are required in order to pass the Certified Professional for Software Architecture (CPSA®) exam at Foundation Level. It is equally suitable for self-study as well as accompanying literature for CPSA-Foundation Level training courses.

Both authors are long-standing members of the iSAQB Foundation Level working group and have played a leading role in shaping the content of the curriculum and the examination, based upon their many years of experience in both practical software engineering and teaching technical subjects to a wide variety of audiences.

I wish you, the reader, many new insights from this book, and much success on your way as a software architect.

Mirko Hillert

CEO of iSAQB GmbH, Berlin.

Responsible for iSAQB international activities.

Foreword by Stefan Tilkov

Software architecture belongs to the main success factors in modern software development. It enables the development of high-quality software systems and the flexible adaption of these to changing requirements, and technologies. In addition, software architecture facilitates delivering on schedule, and helps teams to work cost-efficiently, for the entire life span of systems.

For more than a decade, the international Software Architecture Qualification Board (iSAQB) has succeeded in establishing a diverse set of widely accepted curricula and learning paths for this important engineering discipline, for both foundational and advanced topics. During this time, Gernot and Alexander have helped to shape the Foundation Level curriculum and exams into their current form.

In this book, they share their understanding of methodological and practical software architecture and software engineering with a focus on the preparation for the iSAQB CPSA-F exam. They combine their didactical and practical experience from numerous training sessions and industry projects to provide a concise and profound introduction into the relevant topics of Foundation Level training.

Over the years I have taught numerous software architecture classes myself.

A book like this, covering all the different learning goals, is a very welcome companion and study guide for learners.

It will complement your training sessions and I am sure it will help you towards a better understanding of software architecture and a successful CPSA Foundation Level exam.

Stefan Tilkov

CEO INNOQ Deutschland GmbH, vice president of iSAQB e.V.

Twitter: @stilkov

Foreword by Peter Hruschka

I began teaching design classes in the late 70s, when software architects were still called chief programmers, chief designers or lead designers. Since then the body of knowledge about software and system architecture has increased dramatically. But even in 2021 I feel that the role of software architects is the least understood role in IT. Every other role has a clearer definition: project manager, requirements engineer, programmer, tester. . .

Helping to improve this situation was a key reason for me to join the iSAQB right from the beginning. The iSAQB has achieved a lot in the last decade, increasing the awareness of the importance of that role and providing curricula as a basis for education and training.

This book by Gernot and Alex is a further important milestone to spread the news about this fascinating but challenging role.

It will help future software architects to better understand the Foundation Level curriculum and prepare themselves for the iSAQB CPSA-F exam.

Hopefully, it will also trigger many new companies in various countries of the world to create training courses based on this book, thus increasing the number of highly educated software architects.

I am looking forward to better designed, trustworthy and enjoyable software intensive products and systems.

Dr. Peter Hruschka

Atlantic Systems Guild,

Co-founder and member of both iSAQB and IREB, consultant and author.

Part I: Introduction

This part explains what this book is all about and introduces you to the iSAQB e.V.¹ standardization organization and their Software Architecture Certification, especially the *Certified Professional for Software Architecture - Foundation Level* (CPSA-F)².

This internationally renowned education and certification schema defines both the subject and corresponding examinations.

Content overview

In this first part, we will answer several fundamental questions:

- Why software architecture?
- Why we wrote this book?
- What is the *International Software Architecture Qualification Board*, iSAQB?
- What benefits are to be gained from a CPSA-F certification?

Next, we:

- Introduce the iSAQB Foundation curriculum.
- Explain the iSAQB Foundation examination process.
- Show various ways to prepare for the iSAQB Foundation examination.

About Software Architecture

What is the typical life span of the IT systems you work on? If you ponder this question for a while, you may come up with a surprisingly long period of time. Many systems we encountered have existed for years and may continue to do so for even longer. Often, they originated as a small system and evolved into a product or other long-term venture.

During such a long time, a lot of things are subject to change: Functionalities and features adapt to shifting customer requirements and business goals. Technologies which the architecture relied upon are no longer available. Experienced people leave the development team, whilst project and product management changes.

Most software systems have to continuously adapt to such changes in requirements, technologies and even team and organizational structures. The *field of software architecture* is the engineering science that enables this adaptation process in an environment which is constrained by factors like cost, time

¹ <https://isaqb.org>

² iSAQB (the International Software Architecture Qualification Board) has copyrighted and trademarked its curricula, logos and other intellectual property. In this book we will not append the © symbol at every possible occasion. All rights remain with the iSAQB e.V.

to market and availability of sufficiently skilled humans. The *software architecture of an IT system* is decisive for feasibility, cost- and time-efficiency of its future development: Better architectures lead to better time-to-market and lower maintenance and operational cost.

“The goal of software architecture is to minimize the human resources required to build and maintain the required system.”

Robert C. Martin

Besides technical decisions, software architecture deals with efficient use of *human resources*, therefore minimizing development and operational costs. It goes way beyond finding a great technical solution, as it aims to find compromises between the sometimes conflicting goals of all stakeholders.

Software architecture helps to achieve qualities like maintainability, reliability, safety, performance, security, scalability and operability. It reduces complexity by breaking systems down into manageable units with defined dependencies, therefore enabling efficient communication and reasoning about the inner workings of systems. Software architecture defines rules and technical decisions to guide the development, maintenance and operation of systems.

About this book

There are already a number of well-established books on software architecture (see Appendix C References for a curated and opinionated list), so why did we write another one?

In contrast to existing books, this one completely covers the iSAQB curriculum in a sufficiently detailed yet compact way and can serve as an efficient and effective study guide.

You will find numerous sample questions, helping you to prepare for the CPSA-F examination.

In case you're interested in our (Alexander's and Gernot's) motivation to write this book - we included some information about ourselves in Appendix A.

Conventions used in this book



Relevant for the examination: Boxes like this one contain tips or hints that are especially relevant for the iSAQB examination.



General tips for your architecture: Boxes like this one contain tips or hints that can help to improve your software architecture work in general, which might be relevant for the iSAQB examination.



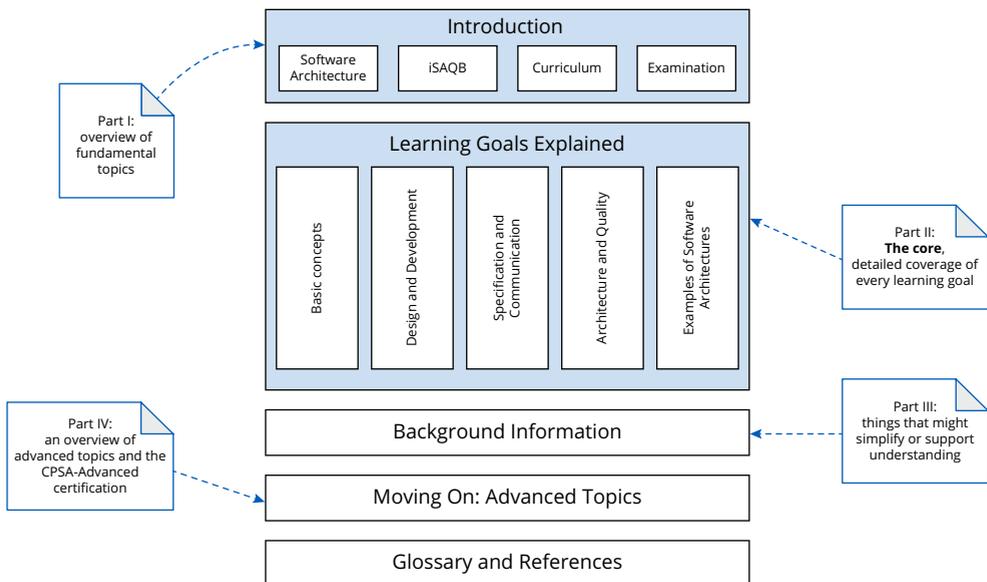
Special information: Sometimes we want to focus your attention - that's where we use information boxes like this.

Our assumptions about you

When writing this book, we had several (potentially silly) assumptions about you (the readers) in mind:

- You work in IT (information technology) and have loads of work to do. Therefore, you want this book to be (relatively) short and compact.
- You have prior experience in developing software systems, and at least a basic understanding of computer programming.
- You want to forward your professional career by passing the iSAQB CPSA-Foundation examination. You may not yet be familiar with some of the iSAQB specific terms, so we have included a glossary, see Appendix B.
- You already have access to books or other resources on software engineering, so we won't repeat all the basics in this book.
- Some of you might want to teach the iSAQB curriculum to others. You rightfully expect a detailed explanation of what is meant by all the learning objectives included in the iSAQB curriculum. In that case, you may be interested in references to additional textbooks and other sources, so you can prepare your personal training material.

Structure of this book



About iSAQB



The *International Software Architecture Qualification Board* 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 a consistent *curriculum* for software architects.
- Defining certification examinations based upon the various CPSA curricula.
- Ensuring high quality of teaching and further training for software architects.
- Ensuring a high quality of certification.

iSAQB defines and prescribes training and examination regulations, but does not carry out any training or examinations itself. iSAQB trainings are delivered by licensed training providers while examinations are handled by separate certifying bodies.

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

iSAQB closes a gap

Software architecture is a relatively young discipline, which is diversely discussed in the IT community despite many publications. There are many different perceptions regarding the roles and responsibilities of software architects and many development projects establish their own understanding.

In other IT disciplines such as project management, business analysis and requirements engineering, or testing, a wider and more generally accepted consensus on job descriptions exists. A variety of independent organizations are offering curricula, that clearly state what knowledge and skills should be transferred in trainings to the participants: e.g., for requirements engineering the IREB³ (International Requirements Engineering Board) sets world-wide standards, for project management there are several organizations with a slightly different focus, like PMI⁴, and for testers the ISTQB⁵ (International Software Testing Qualification Board) sets the standards. For software architecture, this gap is bridged by the iSAQB.

³ <https://www.ireb.org/>

⁴ <https://www.pmi.org/>

⁵ <https://www.istqb.org/>

Levels of iSAQB education and certification

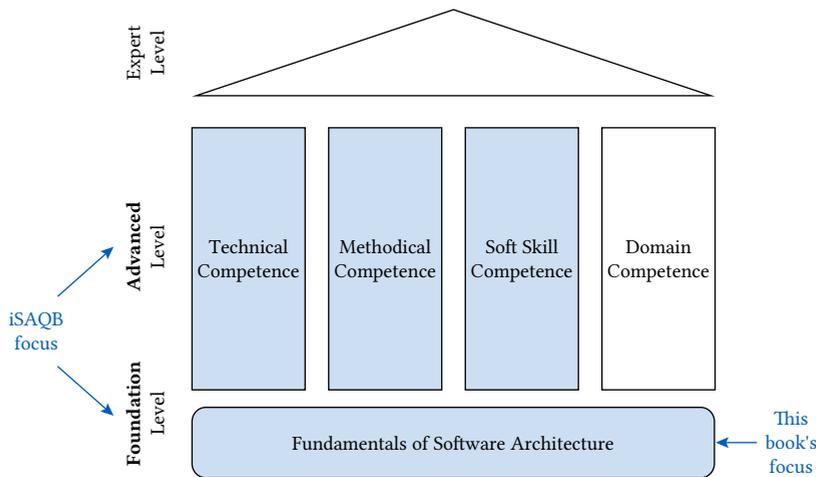


Figure 0.1 Levels of iSAQB Education and Certification

How iSAQB works

Please refer to the iSAQB website for detailed information on how the iSAQB association works, how to apply for membership, etc. For the sake of this book, it is sufficient to note that membership is open to anyone interested in software architecture, and that the heterogeneous structure of iSAQB members allows for well-balanced professional discussions.

The publicly available curricula provide full transparency on all learning goals.

The various curricula are maintained by distinct working groups and curators. For the Foundation Level, the following stakeholders are of particular importance:

Foundation Level Working Group (FLWG)

An iSAQB working group, consisting of volunteer software architecture experts from various domains. This group maintains both the curriculum and the corresponding examination questions to ensure strict compliance between those two artifacts. The curriculum is maintained in an open manner on GitHub <https://github.com/isaqb-org/>. Everybody is free to propose changes and report errors or omissions. The FLWG will publish a new and updated version of the curriculum once every two years, balancing the need for regular updates, and the need for stability in trainings and examinations.

On the other hand, examination questions are strictly confidential, and therefore are managed privately within this working group.

All decisions regarding the content of the CPSA-Foundation Level curriculum are handled by this working group. Major changes have to be approved by the iSAQB executive board.

Certification bodies

Commercial enterprises that organize and conduct the actual examinations. iSAQB ensures the strict independence of education and examination. No certification body is allowed to conduct training, and no training provider is allowed to conduct or organize examinations.

Audit Group

An iSAQB working group that actively monitors the quality of education and examination through inspections and reviews. Trainers, training providers and certification bodies are all subject to such audits.

Open and transparent

Although the content of the iSAQB curriculum is copyrighted, all work related to the curriculum is carried out in *open source* manner:

- The curricula are maintained in public repositories on GitHub, <https://github.com/isaqb-org>.
- Comments and change requests are maintained as GitHub-issues.
- Current versions of the curriculum are available on <https://isaqb-org.github.io/>.

The iSAQB distinguishes between numerous topics, with the Foundation Level being just one of these. Every topic in the advanced level has its own curriculum, see section *iSAQB examination overview*, for details.

About certification

Before we dive into the details of certification and examination, let's clarify the terminology:



- First of all, you should expand your knowledge of the subject area: Enhance your own professional experience with training, reading, and other means of education. This will provide you with the required knowledge and skills.
- In the examination you have to demonstrate your capabilities and level of comprehension regarding software architecture. Your examination will be scored, and you will pass the exam if you achieve at least 60% of the maximum possible score.
- After successfully passing the examination, you will receive a certificate.

Why certification?

With the iSAQB certification you gain some compelling advantages. In particular it:

1. Helps you gain competitive advantage and can positively differentiate you from other professionals;
2. Significantly increases your financial earning potential;
3. Improves your professional credibility;
4. Connects you with the growing community of software architecture experts;
5. Demonstrates your commitment to continuous learning and improvement;
6. Provides a methodical foundation for day-to-day architecture decisions;
7. Exposes you to methodical approaches beyond your daily development and project work;
8. Proves that you have dealt intensively with relevant topics around software architectures.

As the iSAQB is strictly vendor-independent and technology-neutral, their certifications are not limited to certain domains.

On the other hand: Successful certification does not automatically enable you to develop and deploy better software architectures. But a university degree in computer science does not guarantee that either. Nevertheless, most students try to achieve a formal completion of their efforts, as organizations and enterprises commonly pay attention to such formalities.

iSAQB certifications have lifetime validity

In alignment with higher educational degrees (like bachelor, master, PhD and others), iSAQB certifications *are valid forever* and do not need to be renewed.

Other certification bodies require graduates to regularly renew and to regularly pay additional fees for this renewal. This is not the case here

iSAQB Foundation

The title of this section (iSAQB Foundation) is a shortened version of the *official* name “iSAQB Certified Professional for Software Architecture - Foundation Level”. As that 8-word monster is a little difficult to pronounce, we’ve cut it down to “iSAQB Foundation”. This section provides a brief overview of the curriculum for this certification level and helps you to plan your personal preparation for the examination.

iSAQB Foundation Level curriculum

The Foundation Level curriculum [iSAQB-FLC] consists of the following chapters:

1. Basic concepts of software architecture: Roles and tasks of software architects, important results, cooperation with other stakeholders.
2. Design and development of software architectures:

- How can you design structures and concepts systematically? This is the most important part of the CPSA-F curriculum and also the part that has the greatest significance for the practical work of software architects.
 - Design principles and the most important architecture and design patterns.
3. Specification and communication of software architecture:
 - How and what should you document?
 - How do you communicate your ideas, concepts, and decisions with other stakeholders?
 4. Software architecture and quality:
 - How can you systematically achieve the required qualities of your system?
 - How can you analyze, evaluate and diagnose your design?
 5. Examples of software architectures: Software architects should be aware of good and bad examples of architectures and should discuss solutions with peers - beyond the horizon of their own experience.

What do you learn in the CPSA-Foundation Level training?

Graduates of the iSAQB CPSA-Foundation Level examination will have the knowledge and skills required for designing and documenting a specific software architecture based on a sufficiently detailed requirements specification for small and medium-sized systems. They will be equipped with the methodical tools and abilities to enable them to make problem-specific design decisions on the basis of their previously acquired practical experience.

What is NOT covered in the CPSA-F training?

The iSAQB Foundation Level curriculum reflects the contents currently considered by iSAQB members to be necessary and useful for achieving the CPSA-F learning goals.

It is not a comprehensive description of the entire domain of *software architecture*.

The following topics or concepts are *not* part of the CPSA-F curriculum:

- Concrete implementation technologies, frameworks or libraries;
- Programming or programming languages;
- Fundamentals or notations of modeling languages (such as UML, SysML, BPMN or BPEL);
- System analysis and requirements engineering (please refer to the education and certification program of IREB, the International Requirements Engineering Board);
- Software testing (please refer to the education and certification program of ISTQB, the International Software Testing Board);
- Project or product management;
- Specific software or development tools.

iSAQB examination overview

Prerequisites for CPSA-F certification

Candidates aiming to achieve the iSAQB CPSA-Foundation certification require prior knowledge and experience. In particular, substantial practical experience of software development in a team environment is required for understanding the learning materials and considerably increases your prospects of successfully passing the examination.

We consider the following prerequisites to be essential:

- More than 18 months of practical experience with software development, gained through team-based development of several systems outside of formal education.
- Knowledge of, and practical experience with, at least one higher programming language, especially:
 - concepts of modularization (packages, namespaces etc.), parameter passing (call-by value, call-by-reference) and scope (type- and variable declaration and definition);
 - basics of type systems (static vs. dynamic typing, generic data types);
 - error and exception handling in software;
 - potential problems of global state and global variables.
- Basics knowledge of:
 - modeling and abstraction;
 - algorithms and data structures (i.e. Lists, Trees, Hash tables, Dictionary, Map);
 - UML (class, package, component and sequence diagrams) and their relation to source code.

Furthermore, the following will be useful for understanding several concepts:

- Basics and differences of imperative, declarative, object-oriented and functional programming.
- Practical experience in:
 - a higher level programming language;
 - designing and implementing distributed applications, such as client-server systems or web applications;
 - technical documentation, especially documenting source code, system design or technical concepts.

- In CPSA-Foundation examinations, there might be questions covering these prerequisites. None
- of these questions concern specific programming languages or programming constructs.

Relevance of topics for the examination

The curriculum contains prioritized learning goals. For each learning goal, the relevance for the examination of this learning goal or its sub-elements is explicitly stated through the use of a relevance classification as described in the table below.

Relevance class	Learning goal category	Interpretation
R1	Being able to. . .	Will be part of the examination. These are the contents that graduates are expected to be able to apply themselves.
R2	Understanding. . .	May be part of the examination. Graduates are expected to understand these contents in principle, but not necessarily to apply them themselves.
R3	Knowing. . .	This will not be part of the examination. Graduates should know about the topic.

Examination process

The examination is a written multiple-choice test, which can currently be taken *paper-based* or as an electronic assessment, either online or in a test center. That choice depends on the certification body⁶ you choose.

You can find an overview, including contact information, on the iSAQB website <https://isaqb.org>.

The iSAQB e.V. strictly enforces a clear organizational separation of education (training, coaching) and the examination itself:

- Trainers and coaches are never allowed to conduct examinations.
- Certification bodies are never allowed to conduct trainings or perform coaching.

All examinations consist of approximately 40-45 multiple choice questions.

The complete examination has a maximum duration of 75 minutes, which can be prolonged by 15 minutes for those candidates where the language used in the examination is not their native language, though this requires a prior request. For example, if you are a native Spanish speaker and the exam is conducted in English, you will be granted an additional 15 minutes.

Earlier submission is possible.

Each question will earn you from 1 to 3 points, depending on the level of difficulty and the length of the question.

The maximum achievable score is always shown in the question header. As a general principle, the points for correct answers are totaled up and candidates need at least 60% of the achievable maximum score to pass the examination.

⁶ As of April 2021 the iSAQB cooperates with about a dozen licensed certification bodies.

Some additional facts to keep in mind:

- No aids, tools, books or notes of any kind are allowed in the examination.
- Remote proctored online examinations may require you to scan your surroundings with your webcam and you might not be allowed to have additional monitors in your field of view. Some exam providers also prohibit the use of head-sets.
- You may leave the room⁷ as soon as you're finished, but you are not allowed to return. You may stay until the time is over.
- You are required to present an official identification document (passport etc.) with a photo of yourself prior to the examination.
- Taking notes on additional sheets of paper besides the official exam paper is allowed, but that paper must be submitted together with the examination.
- Examinees will be notified of their examination result in writing by the certification body. The printed certificate will be shipped by the certification body to the address given by the participant at the beginning of the examination form, unless a certification body decides to issue digitally signed documents instead.
- Double-check the information in the header of the examination form to ensure that your name will be spelled correctly on the certificate.
- If applicable, make sure that the certifier knows in advance that you are a non-native speaker in order to get extra time.

■ **Examination material is strictly confidential.**

- Every participant of iSAQB examinations is personally required to sign a confidentiality agreement. They may not disclose, make public or otherwise pass on any examination questions or corresponding answers.

Types of examination questions

The iSAQB uses three different types of examination questions, all of which are described below.

■ Visual layout will be different in original examination.

- The visual layout of the original questions will most likely differ from the layout shown here, but the types of questions are exactly identical!

Single choice questions (marked with "A" for Assortment)

You shall select one answer from a list of possibilities.

Only one answer is correct.

Depending on the question, you have to mark the only correct or the only wrong option. Example of an A-question:

7 Online exams are conducted in a supervised manner. In this case there is also an upper limit for the time available, but no lower limit.

A-question: Which of the following is an animal: (1 point)

- Carrot
- Cat
- Computer
- Chaos

Pick from many (marked with “P”)

You shall select two or more from a (larger) number of options. The expected number of answers will always be clearly stated. Example of a P-question:

P-question: From the following list of tools, select the two (2) most important tools for software development: (1 point)

- Compiler
- Editor
- Hammer
- Nail
- Coffee machine

Remarks:

- Every correct answer yields a fraction of the available points (in our example, 0.5 points).
- For every wrong answer, the same fraction is subtracted from this question’s score. In the example, one correct and one wrong answer yields zero points in total.
- Leaving out a correct option is neutral: Marking fewer options than expected will NOT lead to any subtraction of points. In the example, if you only select only one (correct) option, you get a score of 0.5 points.
- In P-questions, more options than you are supposed to select might seem to be applicable. Here you’re expected, based upon your experience and knowledge, to judge which options are better suited than others. A coffee machine is certainly useful for software developers, but other items from this list are more important.

Allocation questions (marked with “K” from German “Klärungsfrage”⁸)

For each option given you are supposed to decide between one of two choices, e.g. between “right/wrong”, “applicable/non applicable”, “static/dynamic”. For every option given, there will always be exactly one correct choice. Example of a K-question:

⁸ Yes, we know. This is weird.

K-question: For every animal given below, choose if it lives in water or on land: (1 point)

water	land	
[x]	<input type="checkbox"/>	Dolphin
<input type="checkbox"/>	[x]	Horse
<input type="checkbox"/>	[x]	Cat
[x]	<input type="checkbox"/>	Octopus

Remarks:

- Every correct answer gives a fraction of the available points (in our example, 0.25 point).
- For every wrong answer, a fraction is subtracted from this question's score. In the example, two correct and two wrong answers yields zero points in sum.
- Leaving out an answer is neutral (zero points). In the example, giving only three (correct) answers yields 0.75 point.

Important note for all types of questions

- The type of questions and the number of answers to be selected are always clearly indicated in the examination form or in the corresponding online system.
- Pay attention to the maximum number of options to be selected. Giving more answers than asked for always results in 0 points for the corresponding question.
- Giving fewer answers than asked for is not punished. In case of doubt, you are better leaving out an answer to avoid getting points subtracted for picking a wrong answer. Please consider this advice seriously and do not take unnecessary risks. A bird in the hand is worth two in the bush.
- The minimum score you can receive for a single question is 0 points. Even if you have ticked more wrong options than correct ones, the penalty applied will not propagate beyond the scope of a question. So guessing is still an option exactly when (and only when) you don't know any of the answers at all.