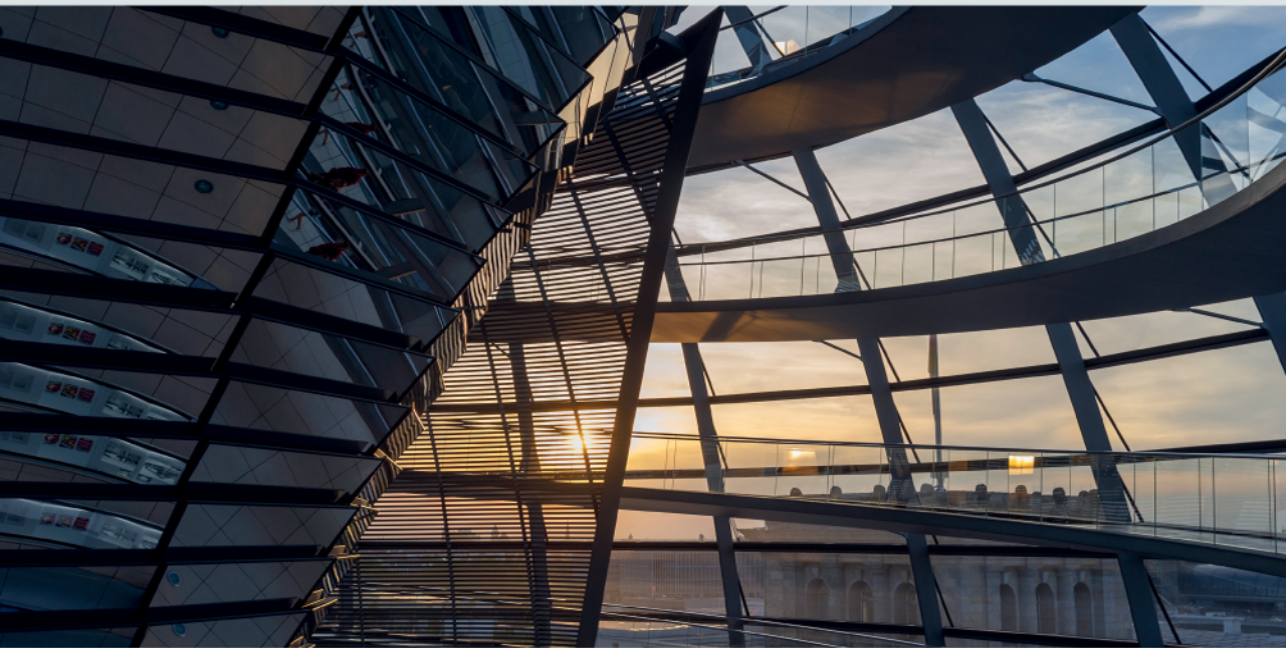


SOFTWARE ARCHITECTURE FOUNDATION

2nd EDITION

GERNOT STARKE & ALEXANDER LORZ



CPSA-F[®] EXAM PREPARATION

Software Architecture Foundation - 2nd Edition

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: 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
ISM
ISO/IEC 20000
ISO/IEC 27001/27002
ISPL
IT4IT®
IT-CMF™
IT Service CMM
ITIL®
MOF
MSF
SABSA
SAF
SIAM™
TRIM
VeriSM™
XLA®

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 1042 5
ISBN eBook (pdf):	978 94 018 1043 2
ISBN ePUB:	978 94 018 1044 9
Edition:	Second edition, first impression, June 2023
Layout and Design:	Coco Bookmedia, Amersfoort-NL
Copyright:	© Van Haren Publishing, 2021, 2023

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	IX
-----------------------------	-----------

FOREWORD BY MIRKO HILLERT	X
--	----------

FOREWORD BY STEFAN TILKOV	XI
--	-----------

FOREWORD BY PETER HRUSCHKA	XII
---	------------

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 approaches 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
LG 2-10: Principles of software deployment	105
CPSA-F CHAPTER 3: SPECIFICATION AND COMMUNICATION OF SOFTWARE ARCHITECTURES	107
LG 3-1: Requirements of technical documentation	107
LG 3-2: Describe and communicate software architectures	110
LG 3-3: Models and notations to describe software architecture	114
LG 3-4: Architectural views.....	121
LG 3-5: Context view	126
LG 3-6: Document cross-cutting concepts.....	129
LG 3-7: Describe interfaces	132
LG 3-8: Document architectural decisions	134
LG 3-9: Resources and tools for documentation.....	136
CPSA-F CHAPTER 4: SOFTWARE ARCHITECTURE AND QUALITY	139
LG 4-1: Quality models and quality characteristics.....	139
LG 4-2: Clarify quality requirements.....	143
LG 4-3: Qualitative analysis	147
LG 4-4: Quantitative evaluation	150
CPSA-F CHAPTER 5: EXAMPLES OF SOFTWARE ARCHITECTURES	153
LG 5-1: From requirements to solutions	153
LG 5-2: Rationale of a technical implementation	154
PART III: BACKGROUND.	155
Additional patterns	155
Appropriateness: it always depends.	161

PART IV: MOVING ON 163

 Advanced Level modules 163

 Advanced Level certification 168

PART V: APPENDICES 171

APPENDIX A: ABOUT THE AUTHORS 172

APPENDIX B: GLOSSARY OF TERMS 174

APPENDIX C: REFERENCES AND FURTHER READING 203

INDEX 209

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 in the current 2023 version 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.



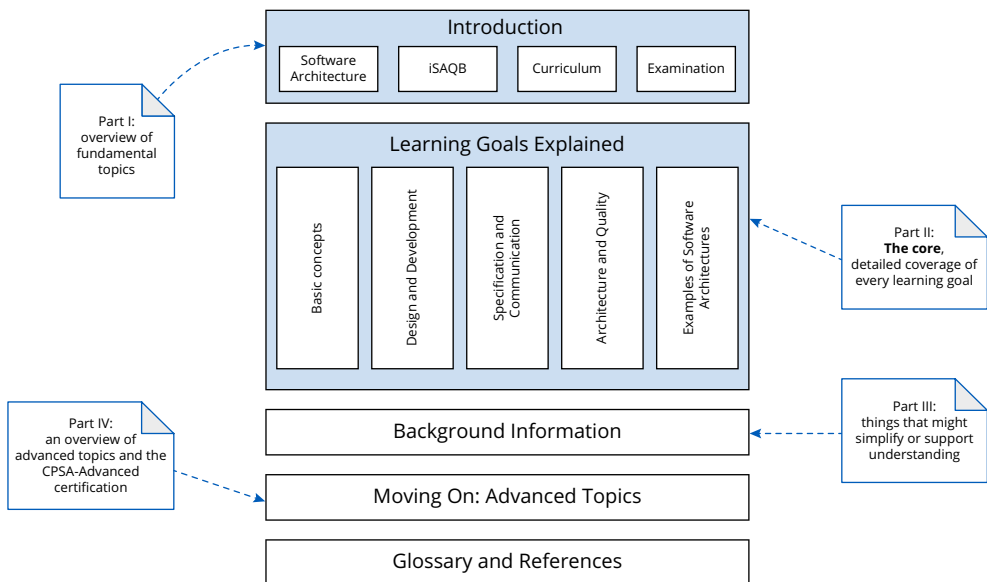
Special information: Sometimes we want to focus your attention - that's where we use information boxes like this.

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

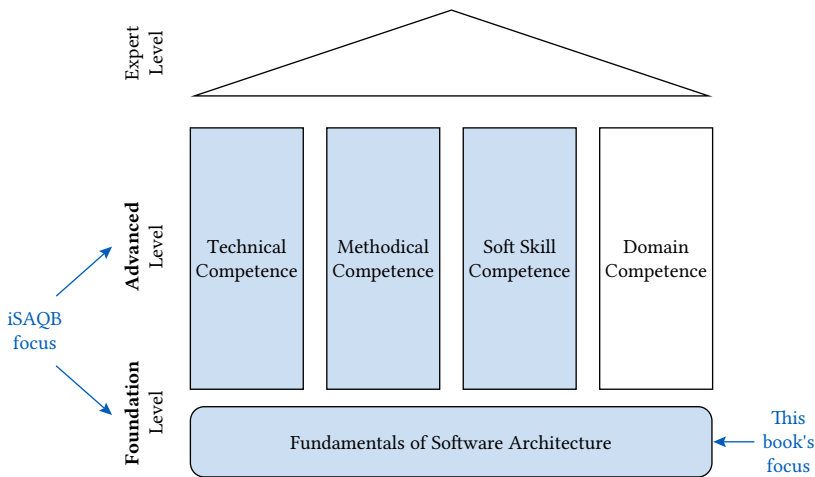


Figure 0.1 Levels of iSAQB Education and Certification

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

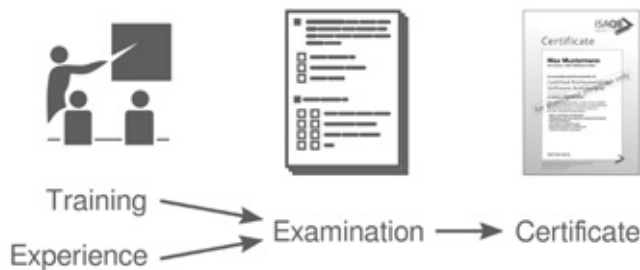
Although the content of the iSAQB curriculum is copyrighted, all work related to the curriculum is carried out in *open source* manner:

- 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.