Curriculum for

Certified Professional for Software Architecture (CPSA)®
Advanced Level

Module
FLEX

Flexible Architecture Models - Microservices and Self-Contained Systems

Version 2020.1-EN; September 5, 2020

Table of Contents

List of Learning Goals

- LG 1-1: The is the first learning goal, in category xy
- LG 2-1: TBD
- LG 2-2: TBD
- LG 3-1: TBD
- LG 3-2: TBD
- LG 4-1: TBD
- LG 4-2: TBD
- LG 5-1: TBD
- LG 5-2: TBD
- LG 7-1: Consistency Models
- LG 7-2: Resilience Patterns

Introduction: General information about the iSAQB Advanced Level

What is taught in an Advanced Level module?

- The iSAQB Advanced Level offers modular training in three areas of competence with flexibly designable training paths. It takes individual inclinations and priorities into account.
- The certification is done as an assignment. The assessment and oral exam is conducted by experts appointed by the iSAQB.

What can Advanced Level (CPSA-A) graduates do?

CPSA-A graduates can:

- Independently and methodically design medium to large IT systems
- In IT systems of medium to high criticality, assume technical and content-related responsibility
- Conceptualize, design, and document actions to achieve quality requirements and support development teams in the implementation of these actions
- Control and execute architecture-relevant communication in medium to large development teams

Requirements for CPSA-A certification

- Successful training and certification as a Certified Professional for Software Architecture, Foundation Level® (CPSA-F)
- At least three years of full-time professional experience in the IT sector; collaboration on the design and development of at least two different IT systems
  - Exceptions are allowed on application (e.g., collaboration on open source projects)
- Training and further education within the scope of iSAQB Advanced Level training courses with a minimum of 70 credit points from at least three different areas of competence
  - existing certifications (for example: Sun/Oracle Java architect, Microsoft CSA) can be credited upon application
- Successful completion of the CPSA-A certification exam

Essentials

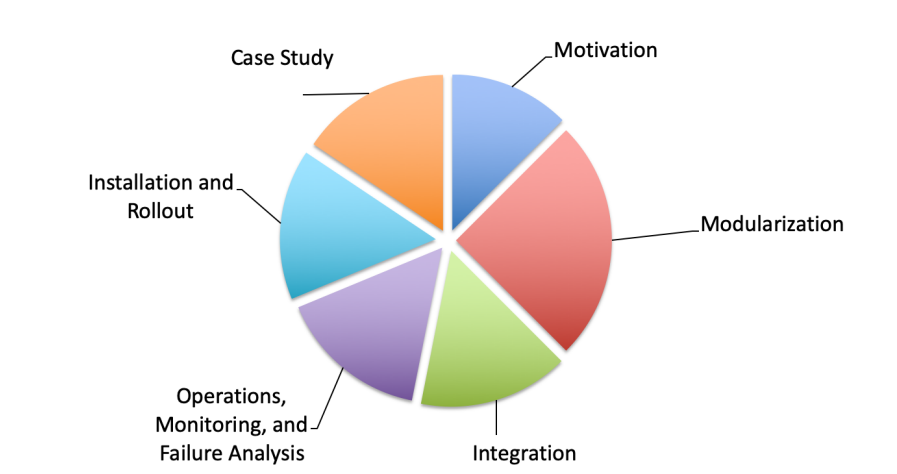
What does the module “FLEX” convey?

The module presents FLEX to the participants … At the end of the module, the participants know … and are able to …

Curriculum Structure and Recommended Durations

Content	Recommended minimum duration (minutes)
1. Motivation	120
2. Modularisation	150
3. Integration	120
4. Installation and Roll Out	120
5. Operations, Monitoring, and Failure Analysis	120
6. Case Study	150
7. Outlook	120
Total	900 (15h)

Allocation of time for the topic areas



Duration, Teaching Method and Further Details

The times stated below are recommendations. The duration of a training course on the FLEX module should be at least 3 days, but may be longer. Providers may differ in terms of duration, teaching method, type and structure of the exercises and the detailed course structure. In particular, the curriculum provides no specifications on the nature of the examples and exercises.

Licensed training courses for the FLEX module contribute the following credit points towards admission to the final Advanced Level certification exam:

Methodical Competence:	**10** Points
Technical Competence:	**20** Points

Communicative Competence:	**00** Points

Prerequisites

TODO Participants **should** have the following prerequisite knowledge:

- Prerequisite 1
- Prerequisite 2, etc.

Knowledge in the following areas may be **helpful** for understanding some concepts:

- Area 1:
  - Knowledge 1
  - Experience 2
  - Knowledge 3
  - Experience 4
  - Understanding 5

Structure of the Curriculum

The individual sections of the curriculum are described according to the following structure:

- **Terms/principles**: Essential core terms of this topic.
- **Teaching/practice time**: Defines the minimum amount of teaching and practice time that must be spent on this topic or its practice in an accredited training course.
- **Learning goals**: Describes the content to be conveyed including its core terms and principles.

This section therefore also outlines the skills to be acquired in corresponding training courses.

Supplementary Information, Terms, Translations

To the extent necessary for understanding the curriculum, we have added definitions of technical terms to the iSAQB glossary and complemented them by references to (translated) literature.

1. Motivation

Duration: 120 min	Practice time: 0 min

1.1. Terms and Principles

Availability, Resilience, Time-to-Market, Flexibility, predictability, reproducibility, homogenization of stages, internet/web scale, distributed systems, parallelizability of feature development, evolution of architecture (build for replacement), heterogenity, automation capability.

1.2. Learning Goals

LG 1-1: The is the first learning goal, in category xy

tbd.

1.3. References

[Humble, et al. 2010], [Wolff 2014], [Humble, et al. 2014]

2. Modularization

Duration: 120 min	Practice time: 30 min

2.1. Terms and Principles

Term 1, Term 2, Term 3

2.2. Learning Goals

LG 2-1: TBD

tbd.

LG 2-2: TBD

tbd.

2.3. References

[Eric Evans 2003], [Lewis, Fowler, et al. 2013], [12 Factor App 2012], [Wolff 2015], [Sam Newman 2015]

3. Integration

Duration: 120 min	Practice time: 30 min

3.1. Terms and Principles

Term 1, Term 2, Term 3

3.2. Learning Goals

LG 3-1: TBD

tbd.

LG 3-2: TBD

tbd.

3.3. References

[Eric Evans 2003], [Parecki et al. 2006], [Hohpe, Woolf 2003]

4. Installation and Roll Out

Duration: 90 min	Practice time: 30 min

4.1. Terms and Principles

Term 1, Term 2, Term 3

4.2. Learning Goals

LG 4-1: TBD

tbd.

LG 4-2: TBD

tbd.

4.3. References

[Vossen, Haselmann, Hoeren 2012], [Wolff, Müller, Löwenstein 2013], [Humble, et al. 2010], [Wolff 2014]

5. Operations, Monitoring, and Failure Analysis

Duration: 90 min	Practice time: 30 min

5.1. Terms and Principles

Term 1, Term 2, Term 3

5.2. Learning Goals

LG 5-1: TBD

tbd.

LG 5-2: TBD

tbd.

5.3. References

[Wolff 2014], [Nygard 2007]

6. Case Study

Duration: 90 min	Practice time: 60 min

During a curriculum-compliant training a case study has to be used to explain and practice the curriculum's concepts.

6.1. Terms and Principles

The Case Study does not introduce new terms, principles, or concepts

6.2. Learning Goals

The Case Study shall not introduce additional learning goals, but intesify the topics via practically relevant exercises.

6.3. References

None. Training providers are responsible for selecting and creating examples and exercises.

7. Outlook

Duration: 120 min	Practice time: 0 min

7.1. Terms and Principles

- Consistency models: ACID, BASE, Partitioning, CAP
- Resilience: Resilient Software Design, Stability, Availability, Graceful Degradation, Circuit Breaker, Bulkhead

7.2. Learning Goals

LG 7-1: Consistency Models

tbd.

LG 7-2: Resilience Patterns

tbd.

7.3. References

[Tanenbaum, van Steen 2006], [Lamport 1998], [Brewer 2000], [Takada 2013], [Nygard 2007], [Hanmer 2007], [Hamilton 2007]

References

This section contains references that are cited in the curriculum.

B

- [Brewer 2000] Eric Brewer, Towards Robust Distributed Systems, PODC Keynote, July-19-2000

E

- [Eric Evans 2003] Eric Evans: Domain-Driven Design: Tackling Complexity in the Heart of Software, Addison- Wesley Professional, 2003

H

- [Hamilton 2007] James Hamilton, On Designing and Deploying Internet-Scale Services, 21st LISA Conference 2007
- [Hanmer 2007] Robert S. Hanmer, Patterns for Fault Tolerant Software, Wiley, 2007
- [Hohpe, Woolf 2003] Gregor Hohpe, Bobby Woolf: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions, Addison-Wesley, 2003, ISBN 978-0-32120-068-6
- [Humble, et al. 2010] Jez Humble, David Farley: Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley, 2010, ISBN 978-0-32160-191-9
- [Humble, et al. 2014] Jez Humble, Barry O'Reilly, Joanne Molesky: Lean Enterprise: Adopting Continuous Delivery, DevOps, and Lean Startup at Scale, O'Reilly 2014, ISBN 978-1-44936-842-5

L

- [Lewis, Fowler, et al. 2013] James Lewis, Martin Fowler, et al.: Microservices - http://martinfowler.com/articles/microservices.html, 2013
- [Lamport 1998] Leslie Lamport, The Part-Time Parliament, ACM Transactions on Computer Systems 16, 2 (May 1998), 133-169

N

- [Sam Newman 2015] Sam Newmann: Building Microservices: Designung Fine-Grained Systems, O'Reilly Media, 2015
- [Nygard 2007] Michael T. Nygard, Release It!, Pragmatic Bookshelf, 2007

O

- [Parecki et al. 2006] Aaron Parecki et al., Explaining OAuth 2.0 http://oauth.net/

T

- [Takada 2013] Mikito Takada, Distributed Systems for Fun and Profit, http://book.mixu.net/distsys/ (Guter Einstieg und Überblick)
- [Tanenbaum, van Steen 2006] Andrew Tanenbaum, Marten van Steen, Distributed Systems – Principles and Paradigms, Pren- tice Hall, 2nd Edition, 2006

V

- [Vossen, Haselmann, Hoeren 2012] Gottfried Vossen, Till Haselmann, Thomas Hoeren: Cloud-Computing für Unternehmen: Technische, wirtschaftliche, rechtliche und organisatorische Aspekte, dpunkt, 2012, ISBN 978-3- 89864-808-0

W

- [Wolff 2014] Eberhard Wolff: Continuous Delivery: Continuous Delivery: Der pragmatische Einstieg, dpunkt, 2014, ISBN 978-3-86490-208-6
- [Wolff 2015] Eberhard Wolff: Microservices - Grundlagen flexibler Software Architekturen, dpunkt, 2015
- [Wolff, Müller, Löwenstein 2013] Eberhard Wolff, Stephan Müller, Bernhard Löwenstein: PaaS - Die wichtigsten Java Clouds auf einen Blick, entwickler.press, 2013

1

- [12 Factor App 2012] 12 Factor App, Guidelines for building apps on Heroku http://12factor.net/