Mock Exam

iSAQB® Certified Professional for Software Architecture – Foundation Level (CPSA-F®)

Document version: 2020.1-EN-rev6
Based on curriculum - version V5.1-EN; January 2, 2020

Explanations to the mock exam Certified Professional for Software Architecture - Foundation Level (CPSA-F®)

This examination is a mock exam, which is based on the certification exam of the Certified Professional for Software Architecture - Foundation Level (CPSA-F®) in form and scope. It serves to illustrate the real iSAQB® CPSA® examination as well as to prepare for the corresponding exam.

The mock exam consists of 39 multiple-choice questions, which can be evaluated with 1 or 2 points depending on the level of difficulty. At least 60 percent must be achieved to pass the exam. 50.0 points can be achieved in this mock examination, you would need 30.0 points to pass.

The following general rules apply: Correct answers result in plus points, incorrect answers result in a deduction of points, but only with regard to the respective question. If the wrong answer to a question leads to a negative score, this question is evaluated with a total of 0 points.

The multiple-choice questions of the mock exam are divided into three types of questions:

A-Questions (Single Choice, Single Correct Answer):
Select the only correct answer to a question from the list of possible answers. There is only one correct answer. You receive the specified score for selecting the correct answer. Depending on the level of difficulty, you can achieve a score of 1 or 2 points.

P-Questions (Pick-from-many, Pick Multiple):
Select the number of correct answers given in the text from the list of possible answers to a question. Select just as many answers as are required in the introductory text. You receive 1/n of the total points for each correct answer. For each incorrect cross, 1/n of the points are deducted. The score is 1 or 2 points depending on the level of difficulty.

K-Questions (Allocation Questions, Choose Category):
For a question, select the correct of the two options for each answer choice ("correct" or "incorrect" or "applicable" or "not applicable"). You will receive 1/n of the points for each correctly placed cross. Incorrectly placed crosses result in the deduction of 1/n of the points. If NO answer is selected in a line, there are neither points nor deductions. The score is 1 or 2 points depending on the level of difficulty.

For a more detailed explanation of the question types and scoring system, further information is available in the CPSA-F examination guide.

The processing time is 75 minutes for native speakers and 90 minutes for non-native speakers. In order to ensure that the preparation for the exam is as authentic as possible, the processing time should be adhered to and any aids (such as seminar materials, books, internet, etc.) should not be used.

The exam can subsequently be evaluated using the solution for this mock exam.

Given that the iSAQB® e.V. is indicated as source and copyright holder, the present mock exam may be used in the context of training courses, for exam preparation or it may be passed on free of charge.

However, it is explicitly prohibited to use these exam questions in a real examination.

Question 1 *A-Question: Select one option.* *1 point*

ID: Q-20-04-01

How many definitions of “software architecture” exist?

☐ (a) Exactly one for all kinds of systems.

☐ (b) One for every kind of software system (e.g. “embedded”, “real-time”, “decision support”, “web”, “batch”, …).

☐ (c) A dozen or more different definitions.

Question 2 *P-Question: Choose the three best aspects.* *1 point*

ID: Q-20-04-02

Which **THREE** of the following aspects are covered by the term “software architecture”?

☐ (a) Components.

☐ (b) Cross cutting concepts.

☐ (c) (internal and external) interfaces.

☐ (d) Database schemata.

☐ (e) Hardware Sizing.

Question 3 *P-Question: Choose the four best answers.* *2 points*

ID: Q-17-13-01

Which **FOUR** of the following statements about (crosscutting) concepts are most appropriate?

☐ (a) Uniform usage of concepts reduces coupling between building blocks.

☐ (b) The definition of appropriate concepts ensures the pattern compliance of the architecture.

☐ (c) Uniform exception handling is most easily achieved when architects agree with developers upon a suitable concept prior to implementation.

☐ (d) For each quality goal there should be an explicitly documented concept.

☐ (e) Concepts are a means to increase consistency.

☐ (f) A concept can define constraints for the implementation of many building blocks.

☐ (g) A concept might be implemented by a single building block.

Question 4

K-Question: Select "Appropriate" or "Not appropriate" for each line.

2 points

ID: Q-17-13-02

In your project, three architects and seven developers are working on the documentation of the software architecture. Which methods are appropriate in order to achieve a consistent and adequate documentation, and which are not?

Appropriate	Not appropriate		
☐	☐	(a)	The chief architect creates the documentation.
☐	☐	(b)	Identical templates are used for the documentation.
☐	☐	(c)	All parts of the documentation are automatically extracted from the source code.

Question 5

P-Question: Choose the four best options.

1 point

ID: Q-17-13-03

Which **FOUR** of the following techniques are best suited to illustrate the interaction of runtime building blocks?

- ☐ (a) Flowcharts.
- ☐ (b) Activity Diagrams.
- ☐ (c) Depiction of screen flows (sequence of user interactions).
- ☐ (d) Sequence diagram.
- ☐ (e) Linear Venn diagram.
- ☐ (f) Numbered list of sequential steps.
- ☐ (g) Tabular description of interfaces.
- ☐ (h) Class diagrams.

Question 6

P-Question: Choose the three best options. *1 point*

ID: Q-17-13-04

Which **THREE** of the following principles apply to testing?

- ☐ (a) In general, exhaustive testing is not possible.
- ☐ (b) In components with many known previous errors, the chances for additional errors are high.
- ☐ (c) Sufficient testing can show that a program is free of errors.
- ☐ (d) Testing can only show the existence of errors.
- ☐ (e) Functional programming does not allow automated testing.

Question 7

K-Question: Select "True" or "False" for each line. *2 points*

ID: Q-17-13-05

Which of the following statements regarding the design principle 'information hiding' are true and which are false?

True	False		
☐	☐	(a)	Adhering to the "information hiding principle" increases flexibility for modifications.
☐	☐	(b)	Information hiding involves deliberately hiding information from callers or consumers of the building block.
☐	☐	(c)	Information hiding makes it harder to distinguish between interface and implementation.
☐	☐	(d)	Information hiding is a derivative of the approach of incremental refinement along the control flow.
☐	☐	(e)	In object-oriented development, information hiding is primarily relevant at class level.

Question 8

P-Question: Choose the two best options. *1 point*

ID: Q-20-04-03

What are the **TWO** most important goals of software architecture?

- ☐ (a) Improve accuracy of patterns in structure and implementation.
- ☐ (b) Achieve quality requirements in a comprehensible way.
- ☐ (c) Enable cost-effective integration and acceptance tests of the system.
- ☐ (d) Enable a basic understanding of structures and concepts for the development team and other stakeholders.

Question 9

K-Question: Select "True" or "False" for each line. *2 points*

ID: Q-20-04-12

Put yourself in the position of a software architect for a large business application in the banking or insurance domain. Which of the following statements is true and which is false?

True	False		
☐	☐	(a)	Your architecture should be structured in a way that allows changes to corresponding business processes without requiring extensive restructuring of the software architecture.
☐	☐	(b)	Required product qualities should drive your architectural decisions.
☐	☐	(c)	To be independent of the infrastructure you should decide your key software architecture structure before infrastructure architects select the hardware or infrastructure for a product.
☐	☐	(d)	Your software architecture should foresee changes in technology (i.e. new UI-frameworks, different deployment strategies, new peripheral devices) and only require local adaptation when it happens.

Question 10

P-Question: Choose the three most important responsibilities. *2 points*

ID: Q-20-04-06

What are your **THREE** most important responsibilities as a software architect with respect to requirements?

☐	(a)	Help the business people to express quality requirements in a way that can be tested.
☐	(b)	Help to identify new business opportunities based on your technology know-how.
☐	(c)	Reject business requirements that contain technical risks.
☐	(d)	Rewrite business requirements in a terminology that can be understood by your development team.
☐	(e)	Check requirements for technological feasibility.

Question 11

P-Question: Choose the three most important action items. *1 point*

ID: Q-20-04-07

You are responsible as an architect for keeping a legacy system up and running according to the ongoing requirements of your business. What are the **THREE** most important action items on your agenda?

- ☐ (a) Negotiating the maintenance budget for your team.
- ☐ (b) Assuring up-to-date documentation of the deployed system.
- ☐ (c) Analyzing the impact of new requirements on the current system.
- ☐ (d) Encouraging the team members to learn new programming languages.
- ☐ (e) Suggesting technology updates in addition to the business requirements to your management.

Question 12

K-Question: Select "True" or "False" for each line. *2 points*

ID: Q-20-04-08

You are the responsible architect for one product in a product family. The product family has an overall product-family architect. Select which of the following statements is true or false.

True	False		
☐	☐	(a)	You have to accept constraints that apply to the whole product family also for your product.
☐	☐	(b)	Since parts of this product family are separately sellable products, your product is not bound to the constraints of the suite.
☐	☐	(c)	You should have regular meetings with your fellow product architects and the family architect to negotiate common quality requirements and constraints.
☐	☐	(d)	You can negotiate deviations from quality requirements that have been defined for the overall suite with the suite architect.

Question 13

K-Question: Select "True" or "False" for each line. *1 point*

ID: Q-20-04-09

Decide for each of the following statements whether it is true or false.

True	False		
☐	☐	(a)	Architectural cornerstones might be decided during iterative development of features.
☐	☐	(b)	The total effort spent on architectural work is much higher in iterative projects compared to waterfall projects.
☐	☐	(c)	Agile projects do not need architecture documents since the development team uses daily standup-meetings to communicate decisions.
☐	☐	(d)	If your systems consist of a set of microservices there is no need for a central architecture document since each service is free to choose its technologies.

Question 14

K-Question: Select "True" or "False" for each line. *2 points*

ID: Q-20-04-10

Discuss which of the following statements regarding project goals and architectural goals is true and which is false.

True	False		
☐	☐	(a)	Project Goals can include functional requirements as well as quality requirements.
☐	☐	(b)	Architectural goals are a derived from the quality requirements for the system or product.
☐	☐	(c)	Business stakeholders should concentrate on business goals and not interfere with architectural goals.
☐	☐	(d)	To avoid conflicts business goals and architectural goals should be non-overlapping sets.

Question 15

P-Question: Choose the two best-fitting answers. *1 point*

ID: Q-20-04-11

What does the rule „explicit, not implicit“ mean for architecture work? Choose the **TWO** best-fitting answers.

- ☐ (a) Architects should avoid recursive structures and replace them by explicit loops.
- ☐ (b) Architects should make the assumptions leading to decisions explicit.
- ☐ (c) Architects should explicitly insist on natural language explanations (i.e. comments) for each building block.
- ☐ (d) Architects should explicitly insist on written or at least oral justifications for development effort estimates from their team.
- ☐ (e) Architects should make prerequisites for their decisions explicit.

Question 16

P-Question: Choose the three most appropriate answers. *1 point*

ID: Q-20-04-19

Identify the **THREE** most appropriate examples for typical categories of software systems.

- ☐ (a) Batch system.
- ☐ (b) Interactive online system.
- ☐ (c) Linnés system.
- ☐ (d) Embedded real-time system.
- ☐ (e) Integration test system.

Question 17

P-Question: Choose the three most often found in practice. *1 point*

ID: Q-20-04-32

There are many approaches that lead to a software architecture. Which of the following are the **THREE** most often found in practice?

- ☐ (a) User-Interface Driven Design.
- ☐ (b) Domain Driven Design.
- ☐ (c) View-based Architecture Development.
- ☐ (d) Bottom-up Design.
- ☐ (e) Majority Voting.

Question 18

P-Question: Choose the three most often used views. *1 point*

ID: Q-20-04-38

Many architecture development methods suggest a view-based approach. Which of the following views are the **THREE** most often used?

- ☐ (a) Physical database view.
- ☐ (b) Context view.
- ☐ (c) Building Block/Component view.
- ☐ (d) Test-driven view.
- ☐ (e) Configuration view.
- ☐ (f) Runtime view.

Question 19

K-Question: Select "Contained" or "Avoided" for each line. *1 point*

ID: Q-20-04-22

You document a component of your software architecture. Which information should be contained in your black box description and which information should be avoided?

Contained	Avoided		
☐	☐	(a)	Interfaces.
☐	☐	(b)	Responsibility.
☐	☐	(c)	Internal structure.
☐	☐	(d)	Hints for the implementation.

Question 20

P-Question: Choose the two most appropriate answers. *1 point*

ID: Q-20-04-17

Which prerequisites have to be fulfilled before developing a software architecture? Pick the **TWO** most appropriate answers.

- ☐ (a) The requirements specification for the system is complete, detailed and consistent.
- ☐ (b) The most important qualities for the system are known.
- ☐ (c) Organizational constraints are known.
- ☐ (d) The programming language has been selected.
- ☐ (e) Hardware for the development team is available.

Question 21

P-Question: Choose the three most appropriate answers. *1 point*

ID: Q-20-04-18

Which factors can influence the design of a software architecture? Pick the **THREE** most appropriate answers.

☐ (a) Political.

☐ (b) Organizational.

☐ (c) Technical.

☐ (d) Virtual.

Question 22

A-Question: Select one answer. *1 point*

ID: Q-20-04-28

Which of the following qualities can most likely be improved by using a layered architecture?

☐ (a) Runtime efficiency (performance).

☐ (b) Flexibility in modifying or changing the system.

☐ (c) Flexibility at runtime (configurability).

☐ (d) Non-repudiability.

Question 23

A-Question: Select one answer. *1 point*

ID: Q-20-04-33

For which kind of system can the Blackboard Architecture pattern be used?

☐ (a) Hard real-time systems.

☐ (b) Rule-based systems.

☐ (c) Linnés systems.

☐ (d) Safety critical systems.

Question 24

A-Question: Select one answer. *1 point*

ID: Q-20-04-20

Which goals are you trying to achieve with the dependency inversion principle?

☐ (a) Big building blocks shall not depend on small building blocks.

☐ (b) Components shall be able to create dependent components more easily.

☐ (c) Building blocks shall only depend on each other via abstractions.

Question 25

K-Question: Select "Tight coupling" or "Loose coupling" for each line. *1 point*

ID: Q-20-04-21

What are characteristics of tight (high) or loose (low) coupling?

Tight coupling	Loose coupling		
☐	☐	(a)	Building blocks directly call dependent building blocks, i.e. without detours via interfaces or abstractions.
☐	☐	(b)	Building blocks use common data types.
☐	☐	(c)	Building blocks use a common table within a relational database.
☐	☐	(d)	When designing building blocks, you have consistently applied the dependency inversion principle.

Question 26

P-Question: Choose the two best answers. *2 points*

ID: Q-20-04-14

Which **TWO** statements about the principle „Don't repeat yourself" (DRY) are correct? (In other words: What could happen, if parts of the source code or configuration do exist in multiple copies in the system?)

☐ (a) DRY reduces security.

☐ (b) Strict adherence to DRY could lead to higher coupling.

☐ (c) The components of the system that contain redundant code can be improved independently of each other.

☐ (d) Adherence to DRY leads to a reduction of attack vectors in IT security.

☐ (e) Applying the Layer patterns allows a consistent application of the DRY principle.

Question 27

K-Question: Select "True" or "False" for each line. *2 points*

ID: Q-20-04-15

You can communicate aspects of your software architecture verbally and/or in writing. How do these variants correlate? Decide for each of the following statements whether it is true or false.

True	False		
☐	☐	(a)	Verbal communication should supplement written documentation.
☐	☐	(b)	Feedback to architecture decisions should be done in writing to ensure traceability.
☐	☐	(c)	Written documentation should always precede oral communication.
☐	☐	(d)	Architects should pick one variant (oral or written) and stick to this choice during the whole development.

Question 28

K-Question: Select "True" or "False" for each line. *2 points*

ID: Q-20-04-37

Which of the following statements about notations for architectural views is true and which is false?

True	False		
☐	☐	(a)	Business Process Model & Notation (BPMN) should only be used by Business Analysts and not for architecture documentation.
☐	☐	(b)	UML deployment models are the only way to document the mapping of software components to infrastructure.
☐	☐	(c)	UML Package Diagrams can be used to capture the building-block view of software architectures.
☐	☐	(d)	As long as the notation is explained (e.g. by a legend), any notation can be sufficient to describe building block structures and collaboration.

Question 29

P-Question: Choose the two best answers. *1 point*

ID: Q-20-04-13

Which architectural views do have practical application for developing software architectures?

- ☐ (a) Pattern View.
- ☐ (b) Observer View.
- ☐ (c) Building-Block (or Component) View.
- ☐ (d) Deployment View.

Question 30

P-Question: Choose the two most appropriate answers. *1 point*

ID: Q-20-04-23

Parts of the context view are a business context and a technical context. Pick the **TWO** most appropriate answers that apply to the technical context.

- ☐ (a) The technical context contains the physical channels between your system and its environment.
- ☐ (b) The technical context contains all the infrastructure on which the components of your system are deployed.
- ☐ (c) The technical context should include hardware pricing or pricing of cloud services used as infrastructure for your architecture.
- ☐ (d) The technical context contains information about the chosen programming language as well as all frameworks used to implement your software architecture.
- ☐ (e) The technical context might contain different elements than the business context.

Question 31

P-Question: Choose the two best reasons. *1 point*

ID: Q-20-04-24

Software architecture documentation could contain descriptions of cross-cutting concerns. Pick the **TWO** best reasons why documentation of cross-cutting concerns is useful.

- ☐ (a) Cross-cutting concepts should focus on the domain and be free of technical information.
- ☐ (b) Aspects or concepts that are used in multiple parts of your software architecture should be described in a non-redundant way.
- ☐ (c) Cross-cutting concepts can be reused in more products within the same organization.
- ☐ (d) Cross-cutting concepts should be implemented by specialists. Therefore, separate documentation is useful.

Question 32

K-Question: Select "True" or "False" for each line. *1 point*

ID: Q-20-04-25

What are guidelines for good interface design? Check which of the following statements is true and which is false.

True	False		
☐	☐	(a)	Use of interfaces should be easy to learn.
☐	☐	(b)	The client code should be easy to understand.
☐	☐	(c)	An interface should be defined by the provider of the appropriate service.
☐	☐	(d)	Interfaces specifications should contain functional and non-functional aspects.

Question 33

K-Question: Select "True" or "False" for each line. *1 point*

ID: Q-20-04-26

One definition says: "Software architecture is the sum of all the decisions you have taken during development. Check which of the following statements about architectural/design decision is true and which is false.

True	False		
☐	☐	(a)	Architectural decisions can implicitly be contained in the structure of the building block/component view.
☐	☐	(b)	Software architects should justify all design decision in writing.
☐	☐	(c)	Architectural decisions can have interdependencies between each other.
☐	☐	(d)	Tradeoffs between conflicting quality requirements should be explicit decisions.

Question 34

K-Question: Select "Good reason" or "No good reason" for each line. *1 point*

ID: Q-20-04-31

Which of the following statements is a good reason for maintaining (adequate) architecture documentation and which is no good reason?

Good reason	No good reason		
☐	☐	(a)	To enable onboarding of new developers.
☐	☐	(b)	To conform to legal constraints.
☐	☐	(c)	To support the work of distributed teams.
☐	☐	(d)	To assist in future enhancements of the product.

Question 35

K-Question: Select "Conflicting" or "Not conflicting" for each line. *1 point*

ID: Q-20-04-30

Which of the following pairs of qualities are usually in conflict to each other, and which are not?

Conflicting	Not conflicting		
☐	☐	(a)	Understandability – Readability.
☐	☐	(b)	Usability – Security.
☐	☐	(c)	Runtime configurability – Robustness.
☐	☐	(d)	Security – Legal Compliance.

Question 36

P-Question: Choose the two best alternatives. *1 point*

ID: Q-20-04-27

ISO 25010 provides generic quality characteristics for software systems. How can quality requirements concerning these characteristics be made more concrete? Pick the **TWO** best alternatives.

- ☐ (a) By developing UI prototypes.
- ☐ (b) By defining explicit interfaces.
- ☐ (c) By discussing or writing scenarios.
- ☐ (d) By creating automatic tests.
- ☐ (e) By creating a quality tree.

Question 37

A-Question: Select one answer. *1 point*

ID: Q-20-04-28

Which of the following things does not help in qualitative analysis of your software architecture? Pick the only wrong answer.

- ☐ (a) Metrics.
- ☐ (b) Architecture models.
- ☐ (c) Quality scenarios.
- ☐ (d) Project status reports.
- ☐ (e) Log files.

Question 38

P-Question: Choose the two most appropriate indicators. *2 points*

ID: Q-20-04-29

You try to analyze your architecture quantitatively. Which are the **TWO** most appropriate indicators for architectural problem areas?

- ☐ (a) High coupling of components.
- ☐ (b) Inappropriate names of public methods.
- ☐ (c) Missing comments.
- ☐ (d) Error clusters.
- ☐ (e) Number of test cases per component.

Question 39

P-Question: Pick two answers. *1 point*

ID: Q-20-04-36

Which of the following alternatives are harder to measure in your software architecture? Pick **TWO** answers.

- ☐ (a) Size of building blocks (e.g. LOC).
- ☐ (b) Change rate of the source code of components.
- ☐ (c) Cohesion of the architectural components.
- ☐ (d) Security level of a component.
- ☐ (e) Number of the developers that contributed to a specific component.