

Mock Exam

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

Presentation 2021.2-rev4-EN-20210606



Question 1

ID: Q-20-04-01

A-Question:

Select one option

1 point

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 1

ID: Q-20-04-01

A-Question:

Select one option

1 point

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

ID: Q-20-04-02

P-Question: Choose the three best aspects.

1 point

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 schema
- (e) Hardware sizing

Question 2

ID: Q-20-04-02

P-Question: Choose the three best aspects.

1 point

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 schema
- (e) Hardware sizing

Question 3

ID: Q-17-13-01

P-Question: Select the **four** best fitting answers

2 points

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 can be 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. Concepts are a means to increase consistency.
- (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 3

ID: Q-17-13-01

P-Question: Select the **four** best fitting answers

2 points

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 can be 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. Concepts are a means to increase consistency.
- (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

ID: Q-17-13-02

K-Question: Select “appropriate” or “not appropriate” for every line.

2 points

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 lead architect coordinates the creation of the documentation.

(b) Identical templates are used for the documentation.

(c) All parts of the documentation are automatically extracted from the source code.

Question 4

ID: Q-17-13-02

K-Question: Select “appropriate” or “not appropriate” for every line.

2 points

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 lead architect coordinates the creation of the documentation.

(b) Identical templates are used for the documentation.

(c) All parts of the documentation are automatically extracted from the source code.

Explanation

Things like *reasoning* or *alternatives* won't be contained in code, but need to be included in documentation, therefore not **all** parts of documentation can be extracted from source code.

Question 5

ID: Q-17-13-03

P-Question: Select the **four** best fitting answers

1 point

Which FOUR of the following techniques are best suited to illustrate the workflow or behavior of the system at runtime?

- (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 5

ID: Q-17-13-03

P-Question: Select the **four** best fitting answers

1 point

Which FOUR of the following techniques are best suited to illustrate the workflow or behavior of the system at runtime?

- (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

ID: Q-17-13-04

P-Question: Select the **three** best fitting answers

1 point

Which THREE of the following principles apply to testing?

- (a) In general, it is not possible to discover all errors in the system.
- (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 shows the existence of errors rather than the absence of errors.
- (e) Functional programming does not allow automated testing.

Question 6

ID: Q-17-13-04

P-Question: Select the **three** best fitting answers

1 point

Which THREE of the following principles apply to testing?

- (a) In general, it is not possible to discover all errors in the system.
- (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 shows the existence of errors rather than the absence of errors.
- (e) Functional programming does not allow automated testing.

Question 7

ID: Q-17-03-05

K-Question: Select "true" or "false" for every line.

1 point

Which of the following statements regarding the information hiding principle 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 work bottom-up.

(d) Information hiding is a derivative of the approach of incremental refinement along the control flow.

Question 7

ID: Q-17-03-05

K-Question: Select "true" or "false" for every line.

1 point

Which of the following statements regarding the information hiding principle 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 work bottom-up.

(d) Information hiding is a derivative of the approach of incremental refinement along the control flow.

Question 8

ID: Q-20-04-03

P-Question: Choose the **two** best options

1 point

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 8

ID: Q-20-04-03

P-Question: Choose the **two** best options

1 point

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

ID: Q-20-04-12

K-Question: Select "true" or "false" for every line.

1 point

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

true

false

(a) The architect collaborates with the stakeholders to determine where the requirements and constraints will change often (e.g., business processes, technologies), and designs the architecture such that changes can occur without requiring extensive restructuring of the software architecture.

(b) Required product qualities should drive your architectural decisions.

(c) The software architecture can be designed completely independent of the hardware and infrastructure.

Question 9

ID: Q-20-04-12

K-Question: Select "true" or "false" for every line.

1 point

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

true

false

(a)

The architect collaborates with the stakeholders to determine where the requirements and constraints will change often (e.g., business processes, technologies), and designs the architecture such that changes can occur without requiring extensive restructuring of the software architecture.

(b)

Required product qualities should drive your architectural decisions.

(c)

The software architecture can be designed completely independent of the hardware and infrastructure.

Question 10

ID: Q-20-04-03

P-Question: Choose the **three** best options

2 points

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

- (a) Support the business people to specify explicit and concrete quality requirements.
- (b) Help to identify new business opportunities based on your technology know-how.
- (c) Reject business requirements that contain technical risks.
- (d) Capture all business requirements in a terminology that can be understood by your development team.
- (e) Check requirements for technological feasibility.

Question 10

ID: Q-20-04-03

P-Question: Choose the **three** best options

2 points

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

- (a) Support the business people to specify explicit and concrete quality requirements.
- (b) Help to identify new business opportunities based on your technology know-how.
- (c) Reject business requirements that contain technical risks.
- (d) Capture all business requirements in a terminology that can be understood by your development team.
- (e) Check requirements for technological feasibility.

Explanation

Concerning option (c): It's **not** our task to *reject* requirements just because they contain risks. We should identify and communicate those risks, but not reject such requirements.

Question 11

ID: Q-20-04-07

P-Question: Choose the **three** best options

2 points

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 11

ID: Q-20-04-07

P-Question: Choose the **three** best options

2 points

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

ID: Q-21-05-01

K-Question: Select "true" or "false" for every option.

1 point

Which of the following statements regarding architecture decisions are true, which are false?

true

false

(a) Architecture decisions never need to be written down because they are already known to the development team.

(b) An architecture decision record helps to make the decision's context understood.

(c) Once a decision has been made on a central or fundamental framework (e.g. persistence framework), that decision must not be changed.

(d) Quality requirements help significantly with architecture decisions.

Question 12

ID: Q-21-05-01

K-Question: Select "true" or "false" for every option.

1 point

Which of the following statements regarding architecture decisions are true, which are false?

true

false

(a) Architecture decisions never need to be written down because they are already known to the development team.

(b) An architecture decision record helps to make the decision's context understood.

(c) Once a decision has been made on a central or fundamental framework (e.g. persistence framework), that decision must not be changed.

(d) Quality requirements help significantly with architecture decisions.

Question 13

ID: Q-20-04-09

K-Question: Select "true" or "false" for every line.

1 point

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

appropriate

not appropriate

(a) Each iteration of an agile development approach could have a impact on the fundamental architecture decisions.

(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 13

ID: Q-20-04-09

K-Question: Select "true" or "false" for every line.

1 point

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

appropriate

not appropriate

(a) Each iteration of an agile development approach could have a impact on the fundamental architecture decisions.

(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

ID: Q-20-04-10

K-Question: Select "true" or "false" for every line.

2 points

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 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 14

ID: Q-20-04-10

K-Question: Select "true" or "false" for every line.

2 points

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

true	false		
<input checked="" type="checkbox"/>	<input type="checkbox"/>	(a)	Project Goals can include functional requirements as well as quality requirements.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	(b)	Architectural goals are a derived from the quality requirements for the system or product.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	(c)	Business stakeholders should concentrate on business goals and not interfere with architectural goals.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	(d)	To avoid conflicts business goals and architectural goals should be non-overlapping sets.

Explanation

Business stakeholder might very well have goals like performance, flexibility or security, which are considered "architecture goals".

Question 15

ID: Q-20-04-11

P-Question: Select the **two** best fitting answers

1 point

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 verbal justifications for development effort estimates from their team.
- (e) Architects should make prerequisites for their decisions explicit.

Question 15

ID: Q-20-04-11

P-Question: Select the **two** best fitting answers

1 point

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 verbal justifications for development effort estimates from their team.
- (e) Architects should make prerequisites for their decisions explicit.

Question 16

ID: Q-20-04-19

P-Question: Select the **three** best fitting answers

1 point

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 16

ID: Q-20-04-19

P-Question: Select the **three** best fitting answers

1 point

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

ID: Q-20-04-32

P-Question: Select the **three** best fitting answers

1 point

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 17

ID: Q-20-04-32

P-Question: Select the **three** best fitting answers

1 point

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

ID: Q-20-04-38

P-Question: Select the **three** most often used architecture views

1 point

Several architecture development methods suggest a view-based approach. Which three of the following views are 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 18

ID: Q-20-04-38

P-Question: Select the **three** most often used architecture views

1 point

Several architecture development methods suggest a view-based approach. Which three of the following views are 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

ID: Q-20-04-22

P-Question: Select the **two** best fitting answers

1 point

When documenting a building block of your software architecture, which two information should the black-box description contain?

- (a) Public interfaces.
- (b) Responsibility of the building block.
- (c) Internal structure of the building block.
- (d) Specification of the implementation details.

Question 19

ID: Q-20-04-22

P-Question: Select the **two** best fitting answers

1 point

When documenting a building block of your software architecture, which two information should the black-box description contain?

- (a) Public interfaces.
- (b) Responsibility of the building block.
- (c) Internal structure of the building block.
- (d) Specification of the implementation details.

Question 20

ID: Q-20-04-17

P-Question: Select the **two** best fitting answers

1 point

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 20

ID: Q-20-04-17

P-Question: Select the **two** best fitting answers

1 point

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.

Explanation

In most cases it is unrealistic to have *complete* requirements specification. Often it is enough to have an overview and know certain details (e.g. quality requirements).

Question 21

ID: Q-20-04-18

P-Question: Select the **three** best fitting answers

1 point

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 21

ID: Q-20-04-18

P-Question: Select the **three** best fitting answers

1 point

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

ID: Q-20-04-18

A-Question:

Select one option

1 point

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).
- (c) Non-repudiability.

Question 22

ID: Q-20-04-18

A-Question:

Select one option

1 point

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).
- (c) Non-repudiability.

Question 23

ID: Q-20-04-33

A-Question:

Select one option

1 point

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

- (a) Hard real-time systems
- (b) Rule-based systems
- (c) Linnés systems
- (c) Safety critical systems

Question 23

ID: Q-20-04-33

A-Question:

Select one option

1 point

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

- (a) Hard real-time systems
- (b) Rule-based systems
- (c) Linnés systems
- (c) Safety critical systems

Question 24

ID: Q-20-04-20

A-Question:

Select one option

1 point

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 24

ID: Q-20-04-20

A-Question:

Select one option

1 point

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

ID: Q-20-04-21

K-Question: Select “tight coupling” or “loose coupling” for each line.

1 point

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 using indirect calls via interfaces or abstractions.

(b) Building blocks use shared complex data structures.

(c) Building blocks use a shared table (for read- and write operations) within a relational database.

(d) When designing building blocks, you have consistently applied the dependency inversion principle.

Question 25

ID: Q-20-04-21

K-Question: Select “tight coupling” or “loose coupling” for each line.

1 point

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 using indirect calls via interfaces or abstractions.

(b) Building blocks use shared complex data structures.

(c) Building blocks use a shared table (for read- and write operations) within a relational database.

(d) When designing building blocks, you have consistently applied the dependency inversion principle.

Question 26

ID: Q-20-04-14

P-Question: Select the **two** best fitting answers

2 points

Which two statements about the principle “Don’t repeat yourself” (DRY) fit best? 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 additional attack vectors in IT security.
- (e) Applying the Layer patterns allows a consistent application of the DRY principle.

Question 26

ID: Q-20-04-14

P-Question: Select the **two** best fitting answers

2 points

Which two statements about the principle “Don’t repeat yourself” (DRY) fit best? 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 additional attack vectors in IT security.
- (e) Applying the Layer patterns allows a consistent application of the DRY principle.

Question 27

ID: Q-20-04-15

K-Question: Select "true" or "false" for every line.

2 points

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 always be done in writing to ensure traceability.

(c) Written documentation should always precede verbal communication.

(d) Architects should pick one variant (verbal or written) and stick to this choice during the whole development.

Question 27

ID: Q-20-04-15

K-Question: Select "true" or "false" for every line.

2 points

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		
<input checked="" type="checkbox"/>	<input type="checkbox"/>	(a)	Verbal communication should supplement written documentation.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	(b)	Feedback to architecture decisions should always be done in writing to ensure traceability.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	(c)	Written documentation should always precede verbal communication.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	(d)	Architects should pick one variant (verbal or written) and stick to this choice during the whole development.

Explanation

- Sometimes verbal communication needs to come first, there is no general rule.
- Feedback should not be restricted to written statements.

Question 28

ID: Q-20-04-37

K-Question: Select "true" or "false" for every line.

2 points

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.

(c) 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 28

ID: Q-20-04-37

K-Question: Select "true" or "false" for every line.

2 points

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.

(c)

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

ID: Q-20-04-13

P-Question: Select the **two** best fitting answers

1 point

Which architectural views have the most practical application for developing software architectures? 1 point

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

Question 29

ID: Q-20-04-13

P-Question: Select the **two** best fitting answers

1 point

Which architectural views have the most practical application for developing software architectures? 1 point

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

Question 30

ID: Q-20-04-23

P-Question: Select the **two** most appropriate answers

1 point

The context view might contain a business context and a technical context, or both. 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 30

ID: Q-20-04-23

P-Question: Select the **two** most appropriate answers

1 point

The context view might contain a business context and a technical context, or both. 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

ID: Q-20-04-24

P-Question: Select the **two** best reasons

1 point

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 31

ID: Q-20-04-24

P-Question: Select the **two** best reasons

1 point

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

ID: Q-20-04-25

K-Question: Select "true" or "false" for every line.

2 points

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

true

false

(a) Use of interfaces should be easy to learn.

(b) The client code should be reasonably easy to understand in relation to the functional complexity.

(c) An interface should provide access to a comprehensive set of implementation details.

(d) Interface specifications should contain functional and non-functional aspects.

(e) Local and remote calls to an interface should behave identically in all aspects.

Question 32

ID: Q-20-04-25

K-Question: Select "true" or "false" for every line.

2 points

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

true	false		
<input checked="" type="checkbox"/>	<input type="checkbox"/>	(a)	Use of interfaces should be easy to learn.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	(b)	The client code should be reasonably easy to understand in relation to the functional complexity.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	(c)	An interface should provide access to a comprehensive set of implementation details.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	(d)	Interface specifications should contain functional and non-functional aspects.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	(e)	Local and remote calls to an interface should behave identically in all aspects.

Explanation

Regarding option (e), "identical behavior in all aspects": It's technically not feasible to have *identical* behavior, at least concerning latency, and response time.

A more detailed explanation can be found in the (rather famous) [Fallacies_of_distributed_computing](#)

Question 33

ID: Q-20-04-26

K-Question: Select "true" or "false" for every line.

1 point

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 impact the structure of the building block or components.

(b) Software architects shall justify all design decisions in writing.

(c) Architectural decisions can have interdependencies between each other.

(d) Tradeoffs between conflicting quality requirements should be explicit decisions.

Question 33

ID: Q-20-04-26

K-Question: Select "true" or "false" for every line.

1 point

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		
<input checked="" type="checkbox"/>	<input type="checkbox"/>	(a)	Architectural decisions can impact the structure of the building block or components.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	(b)	Software architects shall justify all design decisions in writing.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	(c)	Architectural decisions can have interdependencies between each other.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	(d)	Tradeoffs between conflicting quality requirements should be explicit decisions.

Explanation

Not *all* decisions need to be justified in writing - as the requirement for *written* documentation depends on the situation, the team, the system and other factors.

Question 34

ID: Q-20-04-31

K-Question: Select "typical" or "not typical" for every line.

2 points

Which of the following statements are typical reasons for maintaining adequate architecture documentation and which are not typical reasons?

- | typical | not typical | | |
|--------------------------|--------------------------|-----|--|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | To support onboarding of new developers. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | To support the automated testing approach of the system. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | To support the work of distributed teams. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | To assist in future enhancements of the product. |
| <input type="checkbox"/> | <input type="checkbox"/> | (e) | To conform to regulatory or legal constraints. |
| <input type="checkbox"/> | <input type="checkbox"/> | (f) | To ensure that developers have enough work to do. |

Question 34

ID: Q-20-04-31

K-Question: Select "typical" or "not typical" for every line.

2 points

Which of the following statements are typical reasons for maintaining adequate architecture documentation and which are not typical reasons?

- | typical | not typical | | |
|-------------------------------------|-------------------------------------|-----|--|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | (a) | To support onboarding of new developers. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | (b) | To support the automated testing approach of the system. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | (c) | To support the work of distributed teams. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | (d) | To assist in future enhancements of the product. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | (e) | To conform to regulatory or legal constraints. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | (f) | To ensure that developers have enough work to do. |

Question 35

ID: Q-20-04-30

K-Question: Select “conflicting” or “not conflicting” for every line.

1 point

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

conflict

no conflict

- | | | | |
|--------------------------|--------------------------|-----|---------------------------------------|
| <input type="checkbox"/> | <input type="checkbox"/> | (a) | Understandability – Readability. |
| <input type="checkbox"/> | <input type="checkbox"/> | (b) | Usability – Security. |
| <input type="checkbox"/> | <input type="checkbox"/> | (c) | Runtime configurability – Robustness. |
| <input type="checkbox"/> | <input type="checkbox"/> | (d) | Security – Legal Compliance. |

Question 35

ID: Q-20-04-30

K-Question: Select “conflicting” or “not conflicting” for every line.

1 point

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

conflict

no conflict

- | | | | |
|-------------------------------------|-------------------------------------|-----|---------------------------------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | (a) | Understandability – Readability. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | (b) | Usability – Security. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | (c) | Runtime configurability – Robustness. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | (d) | Security – Legal Compliance. |

Question 36

ID: Q-20-04-27

P-Question: Select the **two** best alternatives

1 point

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 automated tests.
- (e) By creating a quality tree.

Question 36

ID: Q-20-04-27

P-Question: Select the **two** best alternatives

1 point

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 automated tests.
- (e) By creating a quality tree.

Question 37

ID: Q-20-04-28

P-Question: Select the **four** best alternatives

1 point

Which of the following alternatives are most suitable for supporting a qualitative analysis of your software architecture? Pick the four best alternatives.

- (a) Quantitative dependency analysis.
- (b) Architecture models.
- (c) Quality scenarios.
- (d) Team size.
- (e) Log files.
- (f) Organizational structure.

Question 37

ID: Q-20-04-28

P-Question: Select the **four** best alternatives

1 point

Which of the following alternatives are most suitable for supporting a qualitative analysis of your software architecture? Pick the four best alternatives.

- (a) Quantitative dependency analysis.
- (b) Architecture models.
- (c) Quality scenarios.
- (d) Team size.
- (e) Log files.
- (f) Organizational structure.

Question 38

ID: Q-20-04-29

P-Question: Select the **two** best fitting answers

2 points

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

- (a) High coupling of components.
- (b) Names of public methods do not reflect their purpose.
- (c) Missing comments.
- (d) Clusters of errors in certain building blocks of the system.
- (e) Number of test cases per component.

Question 38

ID: Q-20-04-29

P-Question: Select the **two** best fitting answers

2 points

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

- (a) High coupling of components.
- (b) Names of public methods do not reflect their purpose.
- (c) Missing comments.
- (d) Clusters of errors in certain building blocks of the system.
- (e) Number of test cases per component.

Question 39

ID: Q-20-04-36

P-Question: Select the **three** best fitting answers

1 point

You try to quantitatively analyze your architecture. Which three of the following properties can you measure reliably in your software architecture? Pick the three best fitting 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.

Question 39

ID: Q-20-04-36

P-Question: Select the **three** best fitting answers

1 point

You try to quantitatively analyze your architecture. Which three of the following properties can you measure reliably in your software architecture? Pick the three best fitting 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.

Explanation

- Size can easily and reliably be measured when statically analyzing source code (lines-of-code metric is a reliable size metric)
- change-rate and number-of-developers-per-component can reliably be measured when taking the version control history into account, which is perfectly feasible with systems like git, subversion or similar tools that are widely used in development.