**ISPE01\_Unit01\_Question01**

The term software requirements specification (SRS) ...

* refers to activities for documenting detailed technical requirements. (0.5 pts)



* refers to the outcome of activities for documenting detailed technical requirements. (0.5 pts)



* involves completely different activities to conceptual requirements engineering. (-1 pts)



* refers to the results obtained by a software architect from design-related activities. (-1 pts)



**ISPE01\_Unit01\_Question02**

Errors in the SRS

* impact the architecture and implementation. (0.25 pts)



* may extend to the test cases because these are prepared on the basis of the SRS. (0.25 pts)



* can be identified using test techniques from requirements engineering. (0.25 pts)



* may also originate from errors in requirements engineering. (0.25 pts)



**ISPE01\_Unit01\_Question03**

A software requirements specification ...

* always describes a system’s internal structure as a white box. (-1 pts)



* specifies guidelines that all design decisions must follow. (0.33 pts)



* always denotes an external view of a system or component. (0.33 pts)



* should contain unique, measurable system characteristics wherever possible. (0.33 pts)



**ISPE01\_Unit01\_Question04**

The elements to be taken into account in SRS are ...

* interfaces between human and technical actors and the system. (0.5 pts)



* all business-related functions but excluding use cases. (-1 pts)



* identifiable from the rendering of the system context. (0.5 pts)



* all outside of the system context. (-1 pts)



**ISPE01\_Unit01\_Question05**

The specification of use cases ...

* contains conceptual objects and conceptual procedures which the system must support. (0.5 pts)



* already contains design decisions for the specific implementation of requirements. (-1 pts)



* contains specified conceptual system components assigned to technical functions but no conceptual functions. (-1 pts)



* contains business object rules which the system must guarantee. (0.5 pts)



**ISPE01\_Unit01\_Question06**

The specification of user interfaces ...

* includes individual dialog boxes but not their correlations. (-1 pts)



* can be achieved using detailed GUI prototypes. (0.5 pts)



* comprises both conceptual and technical validations. (0.5 pts)



* should enable the development team to prepare conceptual GUI tests. (-1 pts)



**ISPE01\_Unit01\_Question07**

The specification of quality characteristics ...

* has a major influence on the set of conceptual requirements. (-1 pts)



* is particularly important for enabling architects to design an appropriate architecture. (0.5 pts)



* must give due consideration to “testability”. (0.5 pts)



* can be omitted from a project if there is a tight deadline because no conceptual functions are specified. (-1 pts)



**ISPE01\_Unit01\_Question08**

Which of the following statements is/are correct?

* The specific structure should be tailored to the current project and the individuals and groups involved. (0.5 pts)



* As a general rule, every company has just one binding template for the specification of all software systems. (-1 pts)



* In addition to the specified system properties, the SRS should also contain descriptive information about itself. (0.5 pts)



* The introduction to the SRS should also include a system overview. (-1 pts)



**ISPE01\_Unit01\_Question09**

For system components ...

* Technical interfaces are described in detail. (0.5 pts)



* Business objects are described, but not their lifecycle. (-1 pts)



* Quality characteristics are only described if they refer to the component, not the entire system. (0.5 pts)



* The technical data model in the form of XML descriptions is always included in the Appendix. (-1 pts)



**ISPE01\_Unit01\_Question10**

Which of the following statements regarding the use of UML models in SRS is/are correct?

* Conceptual procedures are always specified by a combination of class and activity diagrams. (-1 pts)



* The use case diagram is not suitable for use in SRS. (-1 pts)



* Sequence diagrams always contain a specific procedure for both activity and state diagrams. (-1 pts)



* Object diagrams and sequence diagrams can be used to model examples of certain facts. (1 pts)



**ISPE01\_Unit02\_Question01**

For a dialog mask ...

* ... it is only necessary to specify validation rules if information is displayed but not entered. (-1 pts)



* the position, nature, and size of the dialog elements must be specified. (0.5 pts)



* depending on requirements, rules on the activation and deactivation of input elements must be specified. (0.5 pts)



* it is important to specify which information about which conceptual object is saved in the database and how. (-1 pts)



**ISPE01\_Unit02\_Question02**

A dialog flow ...

* includes the dialog flow control as well as the dialog masks, depending on the functional status of the modeled business object. (-1 pts)



* includes dialog flow conditions that automatically control the sequence of dialog masks. (0.5 pts)



* should not include the specification of manual navigation options except in exceptional cases. (-1 pts)



* can be specified using GUI prototypes and UML state diagrams. (0.5 pts)



**ISPE01\_Unit02\_Question03**

Which of the following statements is/are correct?

* GUI elements can be classified into different categories depending on the size and/or complexity of the individual elements. (0.5 pts)



* Atomic GUI elements cannot be broken down any further, unless they are combined to form complex GUI elements. (-1 pts)



* The table shows an example of composite elements made up of atomic GUI elements. (0.5 pts)



* Elements for inputting and outputting complex data structures are known as complex GUI elements. They do not generally contain any validation logic. (-1 pts)



**ISPE01\_Unit02\_Question04**

The specification of GUI elements for enumeration types ...

* is no different from the specification of simple text boxes. (-1 pts)



* must distinguish between the conceptual and technical descriptions of selection options. (0.33 pts)



* must define the sequence in which the available selection options are displayed. (0.33 pts)



* with auto-generated selection options, must ensure that all the required information (e.g. functional label in different languages) is available. (0.33 pts)



**ISPE01\_Unit02\_Question05**

The specification of conversion ...

* defines how the technical data model is depicted in the GUI in a conceptually appropriate manner. (0.33 pts)



* must enable the development team to implement the logic for auto-converting formatting differences between the data model and the GUI. (0.33 pts)



* may result in a situation where individual elements in the data model are made up of the values of multiple GUI elements. (0.33 pts)



* is basically only relevant for composite elements. (-1 pts)



**ISPE01\_Unit02\_Question06**

When specifying a validation rule ...

* the number of possible validation times depends on the GUI technology used. (0.5 pts)



* it is important to consider the influence of messages on the automatic dialog flow control. (-1 pts)



* it is not always possible to specify the precise rendering of failed rules. (-1 pts)



* you should also specify the timing of evaluation, type of validation, and representation of the validation message as well as the rule itself. (0.5 pts)



**ISPE01\_Unit02\_Question7**

Which of the following statements is/are correct?

* The conversion check supplements the required field check but is not required in all cases. (-1 pts)



* The plausibility check must be carried out after every page change. (-1 pts)



* Different validation types are used depending on the requirements placed on the dialog flow. (1 pts)



* The required field check verifies the conceptual accuracy of entered values. (-1 pts)



**ISPE01\_Unit02\_Question08**

When using UML state diagrams to specify dialog flows ...

* Navigation options are modeled by the interim state. (-1 pts)



* The display of a screen mask equates to a state in the diagram. (0.5 pts)



* A possible landing page is modeled by the initial state. (0.5 pts)



* Transitions may have arrowheads at both ends to indicate the option of cyclical navigation. (-1 pts)



**ISPE01\_Unit02\_Question09**

When specifying dialog flows to a transition in the UML state diagram...

* Triggers are used to model navigation activators. (0.5 pts)



* A function is modeled as an action if it is called when moving from one page to another in the system’s business logic. (0.5 pts)



* A condition is modeled if the transition is possible in all cases. (-1 pts)



* A decision node is modeled if the actual dialog flow is not influenced by data stored in the system. (-1 pts)



**ISPE01\_Unit02\_Question10**

When specifying dialog flows ...

* it is important to ensure an appropriate documentation format for the current project situation. For example, GUI prototypes may only be used to illustrate complex GUI dialogs. (-1 pts)



* you can reduce the modeling complexity of state diagrams by generalizing standard navigations. (1 pts)



* it is important to provide a precise, clear, machine-readable description of all potential operations through the dialog. (-1 pts)



* automatically activated navigations should only be considered in exceptional cases. (-1 pts)



**ISPE01\_Unit03\_Question01**

Software components ...

* are independent software units which interact with one another via precisely one interface. (-1 pts)



* are the elements which make up complex software systems and are therefore included in the specification. (0.5 pts)



* are viewed in the specification as a black box. Only their externally visible behavior must be specified. (0.5 pts)



* cannot contain any further software components themselves. (-1 pts)



**ISPE01\_Unit03\_Question02**

Which of the following statements is/are correct?

* The interconnection of multiple components requires clearly defined component interfaces. (0.25 pts)



* Components which implement the same interfaces are usually interchangeable. (0.25 pts)



* The functions implemented by a component are specified by the interface description for that component. (0.25 pts)



* In a specification, it is sometimes only necessary to specify individual system components, not the whole system. (0.25 pts)



**ISPE01\_Unit03\_Question03**

The UML component diagram ...

* has interfaces and human actors as its principal notation elements. (-1 pts)



* is suitable for modeling conceptual dependencies. (0.5 pts)



* is suitable for denoting operations between different components. (-1 pts)



* is a UML structural diagram suitable for modeling the internal structure of systems. (0.5 pts)



**ISPE01\_Unit03\_Question04**

In the UML component diagram ...

* components are denoted by the stereotype. (-1 pts)



* components should always be linked together via interfaces, never directly. (0.33 pts)



* the interface(s) provided by each component is/are displayed. (0.33 pts)



* the interface(s) required by each component is/are displayed. (0.33 pts)



**ISPE01\_Unit03\_Question05**

Interfaces in the UML component diagram ...

* are always grouped together using ports, depending on the specific interface technology. (-1 pts)



* are denoted by the stereotype <<component>>. (-1 pts)



* that are required by components can be modeled with a small, closed circle. (-1 pts)



* that are requested by components can be modeled with an open semi-circle. (1 pts)



**ISPE01\_Unit03\_Question06**

When modeling component behavior with UML diagrams ...

* key component functions can be modeled as use cases in use case diagrams. (0.33 pts)



* conceptual operations are modeled in UML activity diagrams. (0.33 pts)



* the component’s use case diagram indicates which technical interfaces are required for which functions. (0.33 pts)



* use case diagrams are used to refine activity diagrams. (-1 pts)



**ISPE01\_Unit03\_Question07**

Partitions in the UML activity diagram ...

* indicate which actions or activities are implemented by which actors. (0.25 pts)



* indicate which decisions are made by which actors. (0.25 pts)



* help to identify interfaces to systems or components because each intersection between the control flow and the partition boundary must be implemented by an interface. (0.25 pts)



* embellish the diagram to include responsibilities. (0.25 pts)



**ISPE01\_Unit03\_Question08**

Business rules for the specification of components ...

* always contain guidelines that influence the behavior of business processes. (-1 pts)



* consist of a context, at least one condition, and at least one action in the case of operational business rules. (0.33 pts)



* should be comprehensively and consistently specified. (0.33 pts)



* can be documented with decision tables and/or state transition tables. (0.33 pts)



**ISPE01\_Unit03\_Question09**

Decision tables for specifying component behavior ...

* enable the clear representation of complex business rules. (0.5 pts)



* comprise four zones: one for the conditions, one for their evaluations, one for the actions, and one for the allocation of evaluations to actions. (0.5 pts)



* can verify that a set of rules is free from contradictions, but do not check for completeness. (-1 pts)



* may alternatively be modeled by UML state diagrams. (-1 pts)



**ISPE01\_Unit03\_Question10**

State transition tables for specifying component behavior ...

* contain evaluations for rules in complex functional operations. (-1 pts)



* document which actions are possible in which states and which subsequent actions are achieved. (-1 pts)



* illustrate which actions are not possible in which states. (0.5 pts)



* are an alternative to UML state diagrams. (0.5 pts)



**ISPE01\_Unit04\_Question01**

The UML activity diagram ...

* enables the conceptual modeling of messages exchanged between components via object nodes. (0.33 pts)



* enables the modeling of messages required by actions for execution. (0.33 pts)



* enables the modeling of messages which are output as the outcome of executing an action. (0.33 pts)



* can be used to denote technical details about a message exchange. (-1 pts)



**ISPE01\_Unit04\_Question02**

The UML sequence diagram ...

* relates to the activity diagram in the same way as the class diagram relates to the object diagram. (-1 pts)



* is used to illustrate examples of particular operations. (1 pts)



* primarily denotes the precise processing time for an inquiry. (-1 pts)



* is a UML structural diagram for the detailed technical representation of exchanged messages between actors. (-1 pts)



**ISPE01\_Unit04\_Question3**

The lifelines in a sequence diagram ...

* do not contain any information about the precise duration of processing. (0.33 pts)



* may include an activation bar to indicate that an actor is active in relation to the current operation. (0.33 pts)



* are always clearly assigned to one actor (0.33 pts)



* represent the start and end points of interaction partners. (-1 pts)



**ISPE01\_Unit04\_Question04**

The messages in a sequence diagram ...

* may only contain specific message objects in exceptional cases. (-1 pts)



* are displayed in sequence from top to bottom. (0.33 pts)



* should not be modeled in parallel on a horizontal line other than in exceptional cases. (0.33 pts)



* in the form of a call indicate that one interaction partner is activating a function from another interaction partner. (0.33 pts)



**ISPE01\_Unit04\_Question05**

Which of the following statements is/are correct?

* The specification of behavior at technical component interfaces usually defines the use of precisely one type of UML behavioral diagram. (-1 pts)



* The UML sequence diagram is used for highly technical renderings but is unsuitable for modeling possible alternatives. (0.33 pts)



* The UML activity diagram is ideal for the conceptual representation of communication relationships at a functional level but not for the representation of specific technical operations. (0.33 pts)



* Each UML behavioral diagram has its own strengths and weaknesses, so before deciding on a particular diagram type it is important to specify the particular focus of modeling. (0.33 pts)



**ISPE01\_Unit04\_Question06**

In order to identify component interfaces that require detailed specification ...

* the activity diagram may be used but only if it contains partitions. (0.25 pts)



* the sequence diagram may be used. (0.25 pts)



* the use case diagram may be used. (0.25 pts)



* the component diagram may be used. (0.25 pts)



**ISPE01\_Unit04\_Question07**

For a component interface ...

* it is necessary to identify and specify the actual functions offered. (0.5 pts)



* the set of parameters required for each function must be specified. (0.5 pts)



* precisely one outcome type may be specified for each function. (-1 pts)



* a unique name must be given to the interface. (-1 pts)



**ISPE01\_Unit04\_Question08**

The data structures of a component interface ...

* are always denoted in a UML class diagram. (-1 pts)



* cannot always be influenced when accessing the interface of an existing system. (0.33 pts)



* can be modeled in a UML class diagram with classes, attributes, and relationships between classes. (0.33 pts)



* must be specified in detail for every parameter and every potential outcome type. (0.33 pts)



**ISPE01\_Unit04\_Question09**

Which of the following statements is/are correct in relation to specified component interfaces?

* Every call in a sequence diagram can be allocated to precisely one function of an interface. (0.33 pts)



* Every object node in an activity diagram can be allocated to an interface function, either as a parameter or as a return type. (0.33 pts)



* Each identified data type for a given interface must be allocated to a separate function. (-1 pts)



* Inconsistencies between different UML diagrams should be prevented wherever possible in the specification. (0.33 pts)



**ISPE01\_Unit04\_Question10**

Which of the following statements is/are correct?

* It is important to specify precisely how the functions of a component interface are implemented. (-1 pts)



* If a component has more than one interface, all functions should nevertheless be combined into a single interface. (-1 pts)



* Ascribing meaningful names to interfaces will help you to understand and familiarize yourself with the functional role of the interface. (1 pts)



* Deriving specific interface formats from class diagrams in downstream development phases is very difficult. (-1 pts)



**ISPE01\_Unit05\_Question01**

Detailed conceptual data models ...

* do not influence the GUI, because a GUI is not specified with data models. (-1 pts)



* have a decisive influence on the specification of technical interfaces. (0.5 pts)



* represent the functional sequence of information systems. (-1 pts)



* can be denoted by graphical and text models. (0.5 pts)



**ISPE01\_Unit05\_Question02**

ID attributes in data models ...

* are used to distinguish objects technically but not conceptually from one another. (-1 pts)



* are used to distinguish objects conceptually but not technically from one another. (-1 pts)



* are used to distinguish objects both conceptually and technically from one another. (1 pts)



* are only needed to distinguish objects but are not used to identify them. (-1 pts)



**ISPE01\_Unit05\_Question03**

Checking data models for completeness ...

* can be omitted if the project is already behind schedule. (-1 pts)



* is a pre-requisite for specifying detailed conceptual operations with the UML use case diagram. (-1 pts)



* is far more important for GUI specification than for the specification of technical interfaces. (-1 pts)



* can be supported with the use of UML object diagrams. (1 pts)



**ISPE01\_Unit05\_Question04**

For attributes in data models ...

* the admissible value range is restricted by specifying the data type. Wherever possible, opt for data types with minimal storage space requirements. (-1 pts)



* either the default value or multiplicities may be specified, but not both. (-1 pts)



* in addition to the data type, constraints may also be modeled to restrict the possible value range. (0.5 pts)



* unlike object-oriented designs, visibilities are not usually defined in the specification. (0.5 pts)



**ISPE01\_Unit05\_Question05**

The set of admissible attribute values in the data model

* can be significantly extended by using enumerations rather than standard data types. (-1 pts)



* can be selectively extended or restricted for all affected attributes in the data model by using enumerations and making adjustments at precisely one point. (0.5 pts)



* can be asynchronously restricted by using structural business rules. (-1 pts)



* cannot always be modeled in detail. If necessary, they can be documented using natural language. (0.5 pts)



**ISPE01\_Unit05\_Question06**

Which of the following statements relating to detailed conceptual data models is/are correct?

* A conceptual data model is only used for key conceptual functions or technical help functions. (-1 Pts)



* The parameters of functions are specified but not their result type. (-1 pts)



* Structural business rules can often be specified with the UML class diagram. (1 pts)



* Dependencies between classes can be described with so-called associative elements. (-1 pts)



**ISPE01\_Unit05\_Question07**

The UML object diagram ...

* is a type of diagram always used in preference over class diagrams. (-1 pts)



* allows the selective representation of class instances but not the relationships between them. (-1 pts)



* can be used to denote specific attribute values in classes. (1 pts)



* is a behavioral diagram whose structure is based on the UML class diagram. (-1 pts)



**ISPE01\_Unit05\_Question08**

The object diagram helps you to check data models

* by ascribing real values to functions in classes. (-1 pts)



* using intelligible notation elements that are readily understood even by specialist stakeholders. (0.5 pts)



* by breaking down the relationships between classes in the form of structural behavioral rules. (-1 pts)



* by specifically illustrating inherited attributes as part of an object. (0.5 pts)



**ISPE01\_Unit05\_Question09**

Classes in data models...

* can often be allocated to the specific categories of entity, ID, and service. (-1 pts)



* can modeled as value objects if they do not have their own conceptual identity. (0.5 pts)



* are assigned to a conceptual category using the stereotypes <<value>>, <<ID>> and <<service>>. (-1 pts)



* can be used to denote stateless specialist functions. (0.5 pts)



**ISPE01\_Unit05\_Question10**

Entities in data models ...

* can be traced and identified on the basis of their conceptual identity. (0.25 pts)



* are assigned to value objects where information about entities is stored. (0.25 pts)



* unlike value objects, cannot simply be cloned, otherwise conflicts with the conceptual ID attributes may arise. (0.25 pts)



* are frequently subject to a lifecycle, over the course of which their characteristics may change. (0.25 pts)



**ISPE01\_Unit06\_Question01**

The exchange of messages between systems ...

* must be specified as part of the SRS up to and including the level of the specific exchange format. (0.5 pts)



* is detailed in every aspect by a UML class diagram. (-1 pts)



* usually takes the form of binary messages so that they cannot easily be read by humans. (-1 pts)



* in the form of binary messages has the advantage that they are very efficient to store and read. (0.5 pts)



**ISPE01\_Unit06\_Question02**

The exchange of structured messages between systems ...

* has the advantage there is no need to specify the message structure. (-1 pts)



* has the advantage that messages are readable by systems and humans alike. (0.33 pts)



* allow the message content to be checked prior to processing. (0.33 pts)



* simplifies the preparation of technical data models by deriving them directly from the specified message structure. (0.33 pts)



**ISPE01\_Unit06\_Question03**

XML messages ...

* are considered well-formed if they follow the structure specified in the XML schema. (-1 pts)



* are valid if all tags are closed with a corresponding end tag. (-1 pts)



* usually have a root element. In justified exceptional cases, however, this element may be omitted. (-1 pts)



* are comprised of simple and complex elements, each of which must be opened with a <tag> and closed with a </tag>. (1 pts)



**ISPE01\_Unit06\_Question04**

Elements in XML messages ...

* may themselves contain elements. (0.25 pts)



* may contain attributes as well as elements. (0.25 pts)



* can usually be freely named, provided you observe certain requirements regarding the characters used. (0.25 pts)



* unlike attributes are relatively easy to extend and include additional information, for example via additional child elements. (0.25 pts)



**ISPE01\_Unit06\_Question05**

Simple elements in XML messages ...

* are specified with their name and data type in XML schemas as a minimum requirement. (0.5 pts)



* can be extended in their value range but not limited. (-1 pts)



* contain no or precisely one child element. (-1 pts)



* can be used in enumerations. (0.5 pts)



**ISPE01\_Unit06\_Question06**

Complex elements in XML messages ...

* may contain multiple complex child elements but no simple child elements. (-1 pts)



* may contain multiple complex child elements and multiple simple child elements. (1 pts)



* may contain either only complex or only simple child elements. (-1 pts)



* may contain multiple simple child elements but no complex child elements. (-1 pts)



**ISPE01\_Unit06\_Question07**

Which of the following statements on compositors in XML schemas is/are correct?

* Compositors always precisely define how often and in which sequence child elements must occur in XML messages. (-1 pts)



* With <xs:choice>, no or precisely one child element is selected from a set of pre-defined options. (-1 pts)



* With <xs:all>, all elements can occur in any given sequence or quantity. (-1 pts)



* <xs:sequence> defines a fixed sequence of elements in the XML message. (1 pts)



**ISPE01\_Unit06\_Question08**

The definition of element types in the XML schema ...

* makes it easier to reuse elements with the same structure. (1 pts)



* is the only option for reusing element structures. (-1 pts)



* always precedes definition of the root element to ensure that the type definition is available for all subsequent elements. (-1 pts)



* is only possible with the compositor. (-1 pts)



**ISPE01\_Unit06\_Question09**

UML class diagrams...

* can be generated from XML schemas to provide a graphical overview of the structure and content of technical messages. (0.5 pts)



* can be derived from XML schemas, but XML schemas cannot be derived from class diagrams. (-1 pts)



* can only be derived from XML schemas in isolated cases. (-1 pts)



* can be derived from XML schemas; however, fixed element sequences in the class diagram must be modeled with constraints. (0.5 pts)



**ISPE01\_Unit06\_Question10**

The specification of Web services ...

* using an XML language has the advantage that technical systems can communicate with one another regardless of their programming language or technical platform. (0.5 pts)



* with WSDL does not contain any further information about deployment except input and output messages. (-1 pts)



* can be implemented with WSDL, an XML language that was purpose-developed for the specification of Web services. (0.5 pts)



* with WSDL includes the actual messages but not the functions offered by the Web service. (-1 pts)



**ISPE01\_Unit07\_Question01**

Quality characteristics of software systems ...

* describe functions that precisely reflect the actual needs of stakeholders. (-1 pts)



* in their entirety describe the system’s aspired quality goal. (1 pts)



* should be measurable, but there is no need to agree with the test method with stakeholders. (-1 pts)



* must only be compatible with the project’s framework conditions in exceptional cases. (-1 pts)



**ISPE01\_Unit07\_Question02**

Quality models ...

* help to formulate a detailed description of quality characteristics by breaking sub-features down into features. (-1 pts)



* structure the abstract concept of quality into various sub-features. (0.5 pts)



* should be prepared from scratch for each individual software project. (-1 pts)



* are a tool for specifying quality characteristics. (0.5 pts)



**ISPE01\_Unit07\_Question03**

An FCM model ...

* refers to a specific type of quality characteristic. (-1 pts)



* includes metrics for measuring the actual quality achieved as well as sub-features. (0.5 pts)



* contains sub-features to specialize factors. (0.5 pts)



* is comprised of fractures, sub-features, and metrics. (-1 pts)



**ISPE01\_Unit07\_Question04**

The GQM method ...

* is a technique for specifying quality characteristics and for measuring the quality attained. (1 pts)



* only needs to be used once for each organization, as the quality characteristics of software systems tend to be identical. (-1 pts)



* is particularly useful for a comprehensive specification of quality. (-1 pts)



* can only be used for certain system components but not for GUIs. (-1 pts)



**ISPE01\_Unit07\_Question05**

Implementation of the GQM method ...

* is rarely supported by a quality model. (-1 pts)



* comprises 10 steps that are implemented in a pre-defined sequence. (-1 pts)



* is strictly sequential, which means that findings from late stages have no influence on the stages already completed. (-1 pts)



* begins with the definition of quality targets and ends with the formulation of specific quality characteristics. (1 pts)



**ISPE01\_Unit07\_Question06**

Step 1 of the GQM method ...

* includes limiting the scope of quality to selected system components. (0.33 pts)



* forces the consideration of quality from a very particular perspective. (0.33 pts)



* defines the framework which will guide all subsequent steps. (0.33 pts)



* can be skipped if you are aiming to achieve a high-quality overall system, rather than just individual components. (-1 pts)



**ISPE01\_Unit07\_Question07**

Formulation of the question ...

* produces a question that can be answered with Yes or No in exceptional cases only. (-1 pts)



* may be implemented on the basis of a refined quality model. (1 pts)



* is not a creative process but is implemented strictly according to the pre-defined structure of the quality model. (-1 pts)



* occurs independently of the perspective defined in step 1. (-1 pts)



**ISPE01\_Unit07\_Question08**

A definition of the indicators to be measured (metrics) ...

* produces a quantitative, objective measurement in every case. (-1 pts)



* can sometimes lead to subjective indicators. (0.5 pts)



* with subjective metrics tends to be less time-consuming and easier than using objective metrics. (-1 pts)



* answers the question formulated in step 3. (0.5 pts)



**ISPE01\_Unit07\_Question09**

The specification of TARGET values for measured indicators...

* is always the most important step in the GQM method. (-1 pts)



* is a binding requirement for the architecture and development team because this ultimately subjectifies quality characteristics. (-1 pts)



* sets out a quantitative framework which the system must later satisfy. (0.5 pts)



* for subjective indicators should always be defined alongside the measurement technique. (0.5 pts)



**ISPE01\_Unit07\_Question10**

Which of the following statements is/are correct?

* Defining the measurement technique ensures that the implementation of formulated quality characteristics can be verified. (0.33 pts)



* Because the preparation of a detailed specification of quality is very time-consuming, the GQM method should initially be confined to key elements only. (0.33 pts)



* A company-specific quality model may also be used to specify quality characteristics. (0.33 pts)



* When formulating quality characteristics, it is important to ensure that all sub-features of a quality model are covered. (-1 pts)

