**NOTES FOR TRANSLATOR**

**Please write the translation directly at the end of each section (see note after section 1.1)**

**Please keep the italics and underlining the same (they indicate the correct answer(s))**

**Please keep code the same and do not translate**

**Do not worry about other formatting. (Each question and its separate answers are copied and pasted individually by us into another program. This is the reason these questions in this document are not identically formatted since they didn’t come from the same standard document.)**

**Unit 1.1**

1. Grenzen Sie die Begriffe Spezifikation und Requirements Engineering voneinander ab.

*Ziel von Aktivitäten der Spezifikation ist die Erstellung einer technischen Dokumentation der nach außen relevanten Anforderungen, nach denen ein Softwaresystem produziert werden soll. Aufbauend auf den Erkenntnissen des fachlichen Requirements Engineerings, bei der fachliche Anforderungen ermittelt, dokumentiert, geprüft und abgestimmt wurden, wird im Rahmen der Spezifikation eine stark technisch ausgeprägte Dokumentation des zu erstellenden Systems erstellt.*

*Dabei handelt es sich bezogen auf die Aktivitäten des Requirements Engineerings um eine Erweiterung und Detaillierung der Dokumentation von Anforderungen. Hinsichtlich Ermittlungstechniken oder Prüftechniken unterscheiden sich RE und Spezifikation nicht.*

1. Nennen Sie die Aktivitäten eines Softwareprozesses, in denen Spezifikationsdokumente verwendet werden.

*Die Spezifikation ist die Grundlage für die Umsetzung des Systems in Aktivitäten zum Design und der Implementierung als auch für die Formulierung von Testfällen in allen Teststufen sowie die Durchführung und Auswertung von Tests.*

1. Grenzen Sie die Begriffe Spezifikation und Design voneinander ab.

*Die Spezifikation eines Systems gibt aus fachlicher Sicht den technisch detaillierten Rahmen für Designentscheidungen vor. Dabei trifft eine Spezifikation keine Entscheidung darüber, wie das System intern konstruiert werden muss, sondern beschreibt stets nur die nach außen sichtbaren Systemeigenschaften. Erst beim Design werden konkrete Elemente des Systems gestaltet. Dabei muss sich an die Vorgaben der Spezifikation gehalten werden.*

**Unit 1.2**

1. Nennen und beschreiben Sie die zur Spezifikation von GUIs relevanten Aspekte.
* *Inhalte und Aufbau von einzelnen Dialogmasken:*

*Detaillierte Vorgabe zu Art, Größe, Position, Farbe und Inhalt von Elementen einer Bildschirmseite, z. B. von Eingabefeldern, Texten, Schaltflächen, Bildern*

* *Validierung von Daten:*

*Spezifikation der Regeln, um Eingabefelder auf fachliche Plausibilität zu prüfen*

* *Dialogfluss:*

*Spezifikation der Führung des Anwenders durch die Oberfläche in Abhängigkeit von eingegebenen Daten und Aktionen des Anwenders*

1. Nennen und beschreiben Sie die zur Spezifikation von technischen Systemschnittstellen relevanten Aspekte.
* *Der Zweck der Schnittstelle auf einer fachlichen Ebene. Beispiel: Übertragung der Fondskurse in die Fondsverwaltung, Validierung von Adressdaten auf Gültigkeit.*
* *Das detaillierte Verhalten bzw. das technische Protokoll der Kommunikation sowie die Regeln, nach denen das System mit seinem Umfeld kommuniziert. Beispiel: HTTP, FTP.*
* *Die Datenstruktur der Nachrichten, die an der Schnittstelle ausgetauscht werden. Beispiel: XML, CSV*

**Unit 1.3**

1. Erläutern Sie den Zusammenhang zwischen Klassendiagrammen und Objektdiagrammen und nennen Sie je eine Situation, in der Sie diese Diagrammtypen einsetzen würden.

*Das UML-Objektdiagramm wird zur Darstellung eines ganz bestimmten Datensatzes auf Basis der in einem Klassendiagramm vorgegebenen Struktur erstellt. Vergleichbar mit dem Sequenzdiagramm können mit einem Objektdiagramm Sachverhalte exemplarisch dargestellt und veranschaulicht werden.*

*Einsatz: z. B. zur Veranschaulichung konkreter Daten aus einem Klassendiagramm zur Abstimmung mit Fachexperten ODER zur Überprüfung der Eignung des Klassendiagramms anhand von konkreten Datensätzen.*

1. Nennen Sie drei weitere für eine Spezifikation typische Dokumentationsformen, die es neben Text und UML-Diagrammen noch gibt.
* GUI-Prototypen zur Spezifikation von Benutzerschnittstellen,
* XML-Sprachen zur Spezifikation von Datenstrukturen an technischen Systemschnittstellen,
* Entscheidungstabellen zur Spezifikation von Systemverhalten sowie Geschäftsregeln, die je nach Situation zur Spezifikation von Datenstrukturen oder Systemverhalten eingesetzt werden können.

**Unit 2.1**

1. Erläutern Sie kurz, welchen Zweck GUIs für industrielle Informationssysteme erfüllen.

*Industrielle Informationssysteme stellen in der Regel eine GUI, d. h. grafische Benutzeroberfläche, zur Verfügung. Das bedeutet die Information wird visuell dargestellt und vom Nutzer mit den Augen wahrgenommen. Die GUIs industrieller Informationssysteme werden genutzt, um durch den fachlichen Prozess zu führen bzw. zu navigieren. Dabei werden den Nutzern Eigenschaften von Fachobjekten angezeigt und Fachobjekte von Nutzern bearbeitet. Außerdem werden von Nutzern Entscheidungen in das System eingegeben, die zur Ablaufsteuerung durch den Geschäftsprozess erforderlich sind.*

1. Nennen und beschreiben Sie kurz zwei Elemente von GUI-Spezifikationen.
* *Bildschirmdialog*

*Seitenabfolge (auch: Dialog), die den Nutzer durch das System führt und ihn bei der Bearbeitung seiner Aufgaben unterstützt. Ein Bildschirmdialog besteht aus einer oder mehreren Dialogmasken sowie einem festgelegten Dialogfluss.*

* *Dialogmaske*

*Menge angemessener Eingabe- und Ausgabeelemente, die zu einem Interaktionsschritt bezogen auf eine vom System unterstützte Fachfunktion benötigt werden.*

* *Konvertierung*

*Umwandlung von Datensätzen der Datenbank in ein lesbares Format und zurück.*

* *Validierung*

*Regel zur Prüfung von Nutzereingaben, die sicherstellen, dass die eingegebenen Daten im System korrekt gespeichert werden.*

* *Dialogfluss*

*Die Navigationsmöglichkeiten zwischen Dialogmasken. Ein Dialogfluss wird verwendet, um in einem Informationssystem einen Arbeitsablauf abzubilden, dessen Abarbeitung mehrere Masken nacheinander erfordert.*

1. Nennen Sie drei Dokumentationsformen für GUI-Spezifikationen.
* *Beispiele für visuelle Dokumentationsformen:*
	+ *Skizzen,*
	+ *Screenshots von GUI-Prototypen,*
	+ *Diagramme mit der Reihenfolge der Dialogmasken*
* *Beispiele für Dokumentationsformen mit Text:*
	+ *Regeln zur Aktivierung und Deaktivierung von GUI‐Elementen,*
	+ *Validierungsregeln,*
	+ *Dialogflussbedingungen.*

**Unit 2.2**

1. Nennen Sie je ein GUI-Element und einen Anwendungsfall für atomare GUI-Elemente, Komposit-Elemente und komplexe GUI-Elemente.
* *Atomares GUI-Element: z. B. Textfeld, Eingabe einer Email-Adresse;*
* *Komposit-Element: z. B. Optionsfelder, Auswahl von 1 aus N vorgegebenen Optionen, z. B. Wahl ob Hin- und Rückfahrt oder nur Hinfahrt bei der Ticketbuchung;*
* *Komplexes GUI-Element: Datumseingabe per grafischen Kalender, Eingabe des Reisedatums bei einer Ticketbuchung.*
1. Beschreiben Sie kurz anhand eines selbstgewählten Anwendungsbeispiels (nicht das KZF-Kennzeichen), wozu eine Konvertierung benötigt wird.

*Beispiel: Eingabe einer IP-Adresse der Form XXX.XXX.XXX.XXX, wobei XXX jeweils einer Zahl von 0–255 entspricht. In der Datenbank wird nur ein String gespeichert, die Eingabe soll jedoch in 4 Textfeldern erfolgen, die nebeneinander angezeigt werden.*

**Unit 2.3**

1. Nennen Sie vier verschiedene Transaction Levels von GUI-Validierungen und geben Sie zu jedem Transaction Level ein selbstgewähltes Beispiel für eine Validierung an.
* *Verlassen des GUI-Elements: Prüfen des richtigen Datentyps bei der Eingabe einer ganzen Zahl;*
* *Verlassen der Bildschirmseite: Prüfung, ob alle Elemente einer Adresse eingeben wurden;*
* *Zwischenspeichern der Daten, z. B. um die Bearbeitung zu einem späteren Zeitpunkt fortzusetzen: Prüfen, ob alle bereits eingegebenen Daten im richtigen Format eingegeben wurden;*
* *Senden der Daten an das System und Abschließen des Eingabevorgangs: Prüfen, ob das Ende einer Vertragslaufzeit ein Zeitpunkt nach Beginn der Vertragslaufzeit ist.*
1. Beschreiben Sie kurz, auf welche Weise dem Nutzer eine fehlgeschlagene Validierungsregel mitgeteilt werden kann.
	* *Anzeigen einer Fehlernachricht;*
	* *Umranden der betroffenen Eingabefelder;*
	* *Einfärben der Beschriftung der Eingabeelemente;*
	* *Einschränken von Navigationsmöglichkeiten.*

**Unit 2.4**

1. Nennen Sie die Elemente, die bei der Spezifikation von Dialogflüssen berücksichtigt werden müssen.
	* + *Jede im Arbeitsablauf verwendete Maske;*
		+ *Die Festlegung der Reihenfolge in der die Masken durchlaufen werden;*
		+ *Manuelle Navigationsmöglichkeiten zwischen den Masken (zurück, weiter); sowie*
		+ *Dialogflussbedingungen zur Steuerung des Durchlaufs (Überspringen von Masken).*
2. Wie können solche Navigationsmöglichkeiten übersichtlich dokumentiert werden, die es von bzw. zu allen Bildschirmmasken gleichermaßen gibt?

*Häufige oder für alle Dialogmasken mögliche Navigationen können exemplarisch für eine Dialogmaske modelliert und dann im beschreibenden Text für alle weiteren Masken verallgemeinert werden.*

**Unit 3.1**

1. Beschreiben Sie kurz, was eine Komponente ist. Gehen Sie dabei auf den Zusammenhang zwischen System und Komponente ein.

*Komplexe Softwaresysteme werden in Komponenten aufgeteilt. Eine Komponente ist eine unabhängige Softwareeinheit, die aufgrund vereinbarter Schnittstellen mit anderen Komponenten zu einem Softwaresystem zusammengestellt werden kann.*

1. Nennen Sie drei Qualitätsziele, die durch den Einsatz von Komponenten erreicht werden sollen.
* *Einfache Erweiterbarkeit,*
* *bessere Testbarkeit sowie*
* *bessere Wartbarkeit.*
1. Nennen und beschreiben Sie die Elemente, die zu einer Komponente spezifiziert werden müssen.
* *Internes Verhalten, beispielsweise Laufzeitverlängerung des Vertrags, Berechnung der Gesamtsumme;*
* *Abläufe und logische Struktur der ausgetauschten Nachrichten an Schnittstellen von Komponenten.*

**Unit 3.2**

1. Welcher Diagrammkategorie ist das Komponentendiagramm zugeordnet und was sind dessen Hauptelemente?

*Das UML-Komponentendiagramm (engl.: component diagram, Abkürzung: cod) ist ein Strukturdiagramm. Die Hauptelemente sind Komponenten, Schnittstellen sowie Beziehungen von Komponenten zu Schnittstellen.*

1. Wofür werden UML-Komponentendiagramme schwerpunktmäßig eingesetzt?

*Der Schwerpunkt der Modellierung liegt auf dem inneren Aufbau und den Elementen eines Systems. Die modellierten Komponenten können fachliche Komponenten sein. Deren Beziehungen stellen dann fachliche Abhängigkeiten dar, die grundsätzlich auch ohne die Unterstützung von IT-Systemen bestehen. Darüber hinaus kann das Komponentendiagramm auch zur Spezifikation von technischen Komponenten eingesetzt werden. So können beispielsweise Softwarearchitekten die konkreten Komponenten eines Systems sowie deren Schnittstellen festlegen.*

**Unit 3.3**

1. Beschreiben Sie den Einsatz von Partitionen im UML-Aktivitätsdiagramm.

*Eine Partition im Aktivitätsdiagramm grenzt den Einflussbereich der an dem Ablauf dargestellten Akteuren ab. Als Akteur können sowohl menschliche Nutzer als auch technische Systeme eingesetzt werden. Partitionen dienen der Spezifikation und der Darstellung der Verantwortlichkeit für fachliche Funktionen und Entscheidungen. Aus diesen Verantwortlichkeiten leiten sich die konkreten Anforderungen an Schnittstellen und benötigten Geschäftsobjekten ab.*

1. Grenzen Sie die Modellierungstechniken Entscheidungstabellen und Zustandstabellen voneinander ab.

*Entscheidungstabellen sind eine Technik zur Spezifikation von Komponentenverhalten. Sie bieten eine Möglichkeit komplexe Geschäftsregeln übersichtlich zu dokumentieren. Die Entscheidungstabelle eignet sich gut zur Veranschaulichung von komplexeren Regelzusammenhängen. Sie hilft außerdem eine Menge von Geschäftsregeln auf Vollständigkeit zu prüfen.*

*Eine Zustandstabelle wird eingesetzt, um mögliche Aktionen oder Funktionen in Abhängigkeit des aktuellen Zustands von Geschäftsobjekten, Komponenten oder Prozessen zu ermöglichen. Sie zeigt an, welche Aktionen in welchen Zuständen erlaubt bzw. nicht erlaubt sind.*

**Unit 4.1**

1. Beschreiben Sie kurz, wie der Objektknoten im UML-Aktivitätsdiagramm die Spezifikation von Verhalten an Systemschnittstellen unterstützt.

*Ein Objektknoten im Aktivitätsdiagramm wird eingesetzt, um darzustellen, welche Art von Information eine Aktion benötigt, um ausgeführt zu werden, und welche Art Information nach der Ausführung einer Aktion ausgegeben wird. Mit Hilfe von Partitionen und Objektknoten im Aktivitätsdiagramm können auf eine einfache Art und Weise komplexe Abläufe unter Einbeziehung der zwischen Systemen ausgetauschten Nachrichten modelliert werden. Somit lässt sich ein einfacher Überblick über die benötigten Nachrichtentypen darstellen.*

1. Nennen Sie Stärken und Schwächen des UML-Sequenzdiagramms bei der Spezifikation von Systemschnittstellen.
* Stärken:
	+ Darstellung komplizierter Interaktionen und technischer Interaktionskaskaden in ihrer genauen Reihenfolge;
	+ Verbindung zwischen Ablauf und Datenstruktur.
* Schwächen:
	+ Fokus liegt auf konkreten Abläufen;
	+ Ablauf isoliert vom fachlichen Anwendungsfall;
	+ Wird sehr schnell sehr komplex und technisch;
	+ Nur mit fortgeschrittenen UML-Kenntnissen lesbar.

**Unit 4.2**

1. Nennen Sie drei mögliche Startpunkte für die Identifikation konkreter Funktionen von Systemschnittstellen.
* Jeder Aufrufbeziehung von Komponenten zu Schnittstellen im Komponentendiagramm muss eine oder mehrere Funktionen der Schnittstelle zugeordnet werden können.
* Jedem Aufruf im Sequenzdiagramm muss eine entsprechende Funktion der Schnittstelle zugeordnet werden können.
* Jeder Schnittpunkt von Kontrollfluss und Partitionsgrenze im Aktivitätsdiagramm ist ein Indiz für eine Funktion einer Schnittstelle
1. Erläutern Sie, wie Sie prüfen können, ob Sie alle benötigten Nachrichtentypen an einer Schnittstelle identifiziert haben.

*Alle benötigten und gelieferten Nachrichten, die als Objektknoten im Aktivitätsdiagramm modelliert werden, lassen sich auch in der Schnittstellen identifizieren. Es dürfen in der Schnittstellenbeschreibung nicht weniger Parameter und Ergebnisse spezifiziert werden, als aus der Summe der bereits verfügbaren Aktivitätsdiagramme ablesbar sind.*

*Darüber hinaus muss geprüft werden, ob jeder Nachricht im Sequenzdiagramm ein Nachrichtentyp der Schnittstellenspezifikation zugeordnet werden kann.*

1. Nennen Sie vier Quellen für detaillierte Attribute, die für Datenstrukturen in Systemschnittstellen benötigt werden.
	* *Bereits dokumentierte Anforderungen;*
	* *Schnittstellenbeschreibungen bereits existierender Komponenten;*
	* *Industriestandards, Normen;*
	* *Unternehmensrichtlinien.*

**Unit 5.1**

1. Nennen Sie drei Bereiche einer Spezifikation, in denen fachlichen Datenmodelle beschrieben werden.

*Fachlichen Entitäten zur Darstellung, welche Informationen das System intern verarbeiten beziehungsweise unterstützen muss.*

*Fachliches Modell eines Geschäftsobjektes, dessen konkrete Daten über einen Bildschirmdialog vom Nutzer eingegeben werden müssen, z. B. der Antrag auf eine KFZ-Haftpflichtversicherung.*

*Fachliches Modell einer Nachricht, die über eine technische Systemschnittstelle ausgetauscht wird, z. B. der Datensatz einer gewünschten Flugreise zur Abfrage an alle Buchungssysteme für Fluglinien.*

1. Beschreiben Sie kurz, wie das fachliche Datenmodell zur Prüfung der spezifizierten Anforderungen eingesetzt werden kann.

*Das fachliche Datenmodell kann bei der Ermittlung von Anforderungen als eine Art Checkliste helfen: Jedes relevante Element des Datenmodells muss durch das Softwaresystem unterstützt werden.*

**Unit 5.2**

1. Erläutern Sie, welche Probleme es bei der eindeutigen Identifikation von Geschäftsobjekten geben kann und wie diese bei der Spezifikation von Datenmodellen berücksichtigt werden können.

*Attribute von Geschäftsobjekten können sich verändern, bzw. können fachliche Werte von Attributen identisch sein, jedoch verschiedene Objekte bezeichnen. Daher lassen sich Geschäftsobjekte nicht immer auf Grundlage der fachlichen Datensätze zuverlässig unterscheiden. Mit der Ergänzung eines ID-Attributes wird das Datenmodell um eine Eigenschaft ergänzt, mit der sich Datensätze auch noch dann unterscheiden bzw. identifizieren lassen, wenn die fachlichen Attribute eines Geschäftsobjektes dazu nicht ausreichen.*

**Unit 5.3**

1. Begründen Sie kurz, warum sich das Objektdiagramm zur Prüfung von fachlichen Datenmodellen eignet.

*Mit Hilfe von Objektdiagrammen lassen sich bestimmte komplexe Systemzustände darstellen. Mit dieser Darstellung lassen sich zum einen die fachliche Korrektheit von Klassendiagramm prüfen und zum anderen der aktuelle Zustand von Datensätzen bewerten. Insbesondere Mengenangaben lassen sich mit dem Objektdiagramm gut darstellen. Durch diese Darstellung von Datensätzen auf Instanzebene, und nicht wie im Klassendiagramm auf Typebene, ist für das Verständnis des Objektdiagramms kein so großes Abstraktionsvermögen erforderlich, wie für das Verständnis von Klassendiagrammen.*

1. Begründen Sie kurz, warum ein Objektdiagramm mit möglichst echten Datensätzen modelliert werden sollte.

*Kann für jeden relevanten Datensatz ein entsprechendes Objektdiagramm erstellt werden, so ist das ein Indiz für die Eignung des spezifizierten Datenmodells. Häufig werden bei jedoch erst bei der Befüllung von Objektdiagrammen mit echten Daten Lücken und Schwachstellen im Datenmodell identifiziert, die anschließend noch behoben werden müssen.*

1. Begründen Sie kurz, warum es sinnvoll sein kann, im Objektdiagramm nicht alle Attribute von Klassen zu modellieren.

*Oft wird ein Objektdiagramm eingesetzt, um ganz konkrete Wertekonstellationen oder Zustände zu veranschaulichen oder zu diskutieren. Der Ersteller eines Objektdiagramms sollte daher nur die aktuell für das Verständnis relevanten Werte angeben. Alle übrigen Werte müssen nicht belegt werden.*

**Unit 5.4**

1. Erläutern Sie kurz, was Entitäten sind und wozu sie eingesetzt werden.

*Entitäten sind Elemente des Datenmodells, die eine fachliche Identität haben. Entitäten unterliegen häufig einem Lebenszyklus, der mit einem Zustandsdiagramm beschrieben werden kann. Die Eigenschaften einer Entität können sich über den Lebenszyklus ändern.*

1. Erläutern Sie kurz, was Werteobjekte sind und wozu sie eingesetzt werden.

*Werteobjekte (engl.: value objects) sind Datenobjekte ohne eigene Identität. Sie haben keinen eigenen Lebenszyklus und auch kein fachliches ID-Attribut. Wertobjekte werden unter anderem eingesetzt um zusätzliche Informationen zu Entitäten zu speichern.*

1. Erläutern Sie kurz, was Dienste sind und wozu sie eingesetzt werden.

*Als Dienste (auch: Service) werden zustandslose Fachfunktionen bezeichnet, die nicht direkt Entitäten oder Werteobjekten zugeordnet werden können. Zustandslos bedeutet, dass ein Dienst selber keine Attribute hat und damit über keinen inneren Zustand verfügt. Zustandslose Funktionen liefern bei gleichen Parametern immer das gleiche Ergebnis zurück.*

**Unit 6.1**

1. Welche Anforderungen sind bei der Übersetzung einer internen Datenstruktur in ein Austauschformat zu beachten?
* *Erhaltung der Datenstruktur;*
* *Einfache Lesbarkeit von Menschen;*
* *Effiziente Speicherung der Information.*
1. Nennen Sie die Vor- und Nachteile von binären Nachrichten.

*Vorteile:*

*Die technische Datenstruktur bleibt dabei erhalten und kann direkt rekonstruiert werden. Die Speicherung der Daten kann in der Regel sehr effizient erfolgen.*

*Nachteile:*

*Struktur und Inhalte von Binärnachrichten sind nicht direkt von Menschen lesbar und schreibbar.*

1. Nennen Sie die Vor- und Nachteile von strukturierten Textnachrichten.

*Vorteile:*

*Strukturierte Textnachrichten können unter Beachtung der festgelegten Struktur von Menschen gelesen und auch geschrieben werden. Mögliche Syntaxfehler werden sehr einfach erkannt, da jede Nachricht automatisch vor der Verarbeitung auf korrekte Form hin geprüft wird.*

*Nachteile:*

*Umwandlung der technischen Datenstruktur in die Nachricht erforderlich. In der Regel größerer Speicherbedarf als binäre Daten.*

**Unit 6.2**

1. Worin unterscheiden sich Attribute und Elemente in XML-Dateien?

*Die Namen der XML-Elemente werden zwischen spitzen Klammern „<“ und „>“ dargestellt und Tag genannt. Der Inhalt eines XML-Elements steht zwischen dem öffnenden Tag und dem schließenden Tag. Attribute sind dazu geeignet, XML-Elemente mit Eigenschaften zu versehen. Sie dürfen ausschließlich im Anfangs-Tag des Elements stehen.*

1. Welche verschiedenen Elementtypen gibt es in XML? Geben Sie zu jedem ein selbstgewähltes Beispiel in XML an.

***Einfache Elemente*** *enthalten nur Zeichenketten. Im unten aufgeführten Beispiel ist die Bestellnummer ein einfaches Element:*

<bestellnr>48729</bestellnr>

***Strukturierte Elemente*** *enthalten weitere Elemente. Im Beispiel trifft das auf jede Position der Bestellung zu:*

 <position>

 <id>1</id>

 <artikelnr>57823566</artikelnr>

 <menge>4</menge>

 <preisEUR>15,90</preisEUR>

 </position>

***Leere Elemente*** *besitzen keinen Inhalt. Die Information steckt also lediglich in der Präsenz des Elements und – falls vorhanden – in dessen Attributen. Das HTML-Element für Zeilenumbrüche ist hierfür ein Beispiel:*

<br />

1. Worin unterscheiden sich die Eigenschaften Gültigkeit und Wohlgeformtheit von XML-Dokumenten?

*Wohlgeformtheit bedeutet, dass alle allgemeinen Syntaxregeln eingehalten wurden. Gültigkeit bedeutet, dass auch die Regeln eines zu der XML-Datei angegebenen Schemas erfüllt werden.*

**Unit 6.3**

1. Spezifizieren Sie das folgende einfache XML-Element mit XML-Schema:

**<lieferdatum>2014-05-23</lieferdatum >**

*Lösung:*

<xs:element name="lieferdatum" type="xs:date" />

**Unit 6.4**

1. Erläutern Sie, wie Kompositoren aus einem XML-Schema in ein UML-Klassendiagramm übertragen werden können.

*Alle Kompositoren sowie das mehrfache Auftreten von Elementen werden durch Assoziationen ausgedrückt. Die Richtung der Assoziation ergibt sich dabei aus der Eltern-Kind-Abhängigkeit der Elemente. Die maxOccurs- und minOccurs-Attribute werden hierbei mit berücksichtigt, auch wenn deren Standardwert gilt. Sie werden durch Multiplizitäten im Klassendiagramm dargestellt. Die Abbildung des Kompositors von xs:choice in das Klassendiagramm ist nur bedingt möglich. Für den Fall, dass sich die einzelnen Elemente innerhalb von xs:choice sinnvoll unter einem Oberbegriff verallgemeinern lassen, kann im Klassendiagramm die Vererbung eingesetzt werden. Geht das nicht, ist eine Darstellung in UML nicht direkt möglich. Für diesen Fall sollte eine Notiz im Diagramm vermerkt werden, dass jeweils nur eines aus der Menge der Elemente auftreten kann.*

1. Erläutern Sie, warum die XML-Datentypen mit in das Klassendiagramm übertragen werden müssen.

*Die tatsächlichen technischen Datentypen der Attribute werden dabei übernommen, damit im späteren Verlauf des Projektes die Typsicherheit gewährleistet werden kann.*

**Unit 6.5**

1. Nennen Sie die XML-Sprache mit der Webservices typischerweise spezifiziert werden und nennen Sie die typischen Elemente der Spezifikation von Webservices.

*Eine XML-Sprache zur Spezifikation von Webservices ist WDSL. In einer WDSL-Datei werden zu einem Webservice folgende Elemente definiert:*

* *Funktionen des Webservices mit deren Ein- und Ausgabenachrichten;*
* *Adresse des Services (URL); sowie*
* *weitere technische Angaben, technischer Zugriff und Deployment.*
1. Welche Elemente müssen zu jeder Funktion eines Webservice spezifiziert werden?
* *Name*
* *Beschreibung*
* *Eingabenachricht*
* *Ausgabenachricht*
1. Beschreiben Sie den Ablauf der Kommunikation zwischen einem Client und einem Server, die über einen Webservice und per XML-Nachrichten miteinander kommunizieren.

*Der Client sendet eine XML-Nachricht an den Webservice. Dieser validiert zunächst, ob die Nachricht wohlgeformt und gültig ist. Anschließend werden die Parameter ausgelesen und auf deren Basis das Ergebnis ermittelt. Das Ergebnis wird in eine XML-Nachricht geschrieben, die dem Client als Antwort übermittelt wird. Der Client prüft nun die Nachricht auf Wohlgeformtheit und Gültigkeit. Besteht die XML-Nachricht beide Prüfungen, wird das Ergebnis verarbeitet.*

**Unit 7.1**

1. Nennen und erläutern Sie, was die Anwendung des SMART-Modells auf die Qualitätseigenschaften bedeutet.

*Die formulierten Qualitätseigenschaften für Softwaresysteme sollten nach dem Schema SMART beschrieben werden. SMART ist ein Akronym, das wie folgt aufgelöst werden kann:*

* + ***S*** *= spezifisch: Qualitätseigenschaften sollen deutlich beschrieben und definiert sein.*
	+ ***M*** *= messbar (qualitativ, quantitativ): Die Umsetzung der Qualitätseigenschaften muss geprüft werden können.*
	+ ***A*** *= akzeptabel: Die Qualitätseigenschaften müssen von den Stakeholdern des Projekts akzeptiert werden, dazu zählt auch die Akzeptanz der Prüfmethode.*
	+ ***R*** *= realistisch: Qualitätseigenschaften müssen unter Berücksichtigung der fachlichen und organisatorischen Rahmenbedingungen erreicht werden können.*
	+ ***T*** *= terminiert: Qualitätseigenschaften müssen innerhalb eines vorgesehenen Zeitrahmens, z. B. in einem bestimmten Release umgesetzt werden.*
1. Was ist die Grundidee eines Qualitätsmodells?

*Die Grundidee eines Qualitätsmodells ist die Zergliederung eines abstrakteren Oberbegriffs in verschiedene Unterbegriffe. Diese Unterbegriffe werden wiederum so lange zergliedert, bis eine konkret messbare Qualitätseigenschaft formuliert werden kann. Ein Qualitätsmodell besteht daher aus einer Hierarchie von Begriffen, die bei der Spezifikation konkreter Qualitätseigenschaften helfen können.*

**Unit 7.2**

1. Beschreiben Sie kurz was die GQM-Methode ist und wozu diese eingesetzt wird.

*Die GQM-Methoden ist ein Vorgehensmodell zur Spezifikation von prüfbaren Qualitätseigenschaften und der Durchführung ihrer Messungen. Ziel von GQM ist die Entwicklung von speziell auf ein spezifisches Softwaresystem angepasste Qualitätseigenschaften.*

1. Welche Schritte der GQM-Methode müssen zur Spezifikation von Qualitätseigenschaften durchgeführt werden?

*Die GQM-Methode lässt sich zur Spezifikation von Qualitätseigenschaften in die folgenden Schritte einteilen:*

1. *Definition der Auswertungsziele (engl.: Goals)*
2. *Verfeinerung der Auswertungsziele mit einem Qualitätsmodell*
3. *Ableitung der Fragestellungen (engl.: Questions)*
4. *Ableitung der Messewerte und Indikatoren (engl.: Metrics)*
5. *Festlegen von geforderten konkreten Messwerten*
6. *Vorgehen und Mechanismen zum Messen festlegen*
7. *Formulierung von prüfbaren Qualitätseigenschaften*