



Prüfungsklausur Softwaretechnik I

A

Bewertung

Aufgabe 1 (2 Punkte):

Für die phasenbezogene Software-Entwicklung (Problemdefinition und Erprobung ausgenommen) seien die anteiligen Aufwände bekannt (Anforderungsanalyse 10%, Spezifikation 30%, Entwurf 20% und Implementation 40%). Bestimmen Sie den Zeitaufwand für die Implementation nach der *Analogiemethode* in Personenmonaten, wenn in einem konkreten Projekt für die ersten beiden Phasen (Anforderungsanalyse und Spezifikation) bereits 6 Monate benötigt wurden.

$$t(A + S = 40\%) = 6 \text{ Monate}, \quad t(I = 40\%) = 6 \text{ Monate} \quad (2 \text{ P.})$$

Aufgabe 2 (2 Punkte):

Welche *Eigenschaftstransformation einer Anforderung* wird durch die jeweilige Methode der Anforderungsanalyse erreicht

a) Interview,

b) Begriffsanalyse.

implizit \rightarrow *explizit* (1 P.) *unverträglich* \rightarrow *verträglich* (1 P.)
(auch andere)

Aufgabe 3 (3 Punkte):

Welche Anforderungsart wird hauptsächlich beim *Entwurf* umgesetzt? Nennen Sie zwei Ergebnisse der Entwurfsphase.

Systembezogene Anforderungen (1 P.)

Ergebnisse: Architektur, Programmbeschreibung (2 P.)

(oder: *White-Box-Testplan, eingeworbene COTS*)

Aufgabe 4 (4 Punkte):

Mit welcher Komponente wird beim Entwurf auf der Grundlage einer Komponentenhierarchie begonnen bei

a) einem *Top-Down-Entwurf*?

b) einem *Hardest-First-Entwurf*?

*oberste Komponente
der Hierarchie* (1 P.)

*die „schwierigste“ Komponente
aus empirischer Sicht* (1 P.)

Wann ist der jeweilige Ansatz zweckmäßig?

bei einer Neuentwicklung (1 P.)

bei erkennbaren Problemkomponenten (1 P.)

Aufgabe 5 (6 Punkte):

Charakterisieren Sie kurz die Begriffe der *Programmierungstechnik*, der *Akquisition* und der *Konfiguration*.

Programmierungstechnik: Art und Weise, Quellcode zusammenzustellen, zu übersetzen und zu verarbeiten (2 P.)

Akquisition: Einwerbung von COTS (2 P.)

Konfiguration: spezielle Ausprägung einer Software-Architektur (2 P.)

Aufgabe 6 (4 Punkte):

Was besagen die *C-Abdeckungsmaße* beim White-Box-Test? Nennen Sie drei *typische Äquivalenzklassen* von Testdaten beim Black-Box-Test.

*C-Maße: Grad der Abdeckung der Knoten und Kanten eines Flussgraphen
beim White-Box-Test (1 P.)*

Äquivalenzklassen: Normal-, Falsch-, Extremwerte (3 P.)

Aufgabe 7 (2 Punkte):

Beschreiben Sie ganz kurz den Unterschied zwischen einem *Alphatest* und einem *Betatest*.

Alphatest (Systemtest) schließt mit dem Integrationstest ab, (1 P.)

Betatest (Erprobung) betrifft den Akzeptanztest eines Softwaresystems durch den Auftraggeber (1 P.)

Aufgabe 8 (2 Punkte):

Welche der Personen aus dem SE-Personal (Systemanalytiker, Auftraggeber, Akquisiteur, Programmierer, Tester) gehört zu einem *agilen Software-Entwicklungsteam*?

zwei Programmierer (mit Rollenwechsel zum Tester), (1 P.)

ein (Vetreter des) Auftraggeber (1 P.)

Aufgabe 9 (2 Punkte):

Welcher Anforderungsart sind jeweils die Ursachen für die folgende *Wartungsaktivität* zuzuordnen

a) Erweiterung,

b) Verbesserung.

funktionale Anforderungen (1 P.)

qualitative Anforderungen (1 P.)

Aufgabe 10 (8 Punkte):

Nennen Sie jeweils zwei Beispiele für ein *CASE*-, *CARE*- und *CAME-Tool*. Geben Sie zwei Komponenten einer Entwicklungsinfrastruktur an, die über eine Programmierumgebung hinausgehen.

CASE: Rational, Textverarbeitung (usw.) (2 P.)

CARE: Together, Disassembler (2 P.)

CAME: Logiscope, Together (2 P.)

Entwicklungsinfrastruktur: Community, Erfahrungshintergrund (oder auch Standards) (2 P.)

Aufgabe 11 (2 Punkte):

Geben Sie jeweils ein Beispiel einer *Software-Metrik* bzw. eines *Software-Maßes* an für

a) eine nominale Skalierung,

b) eine Verhältnisskalierung.

ISO9000 (1 P.)

Antwortzeit (1 P.)

Aufgabe 12 (4 Punkte):

In welchem Bereich (Produkt, Prozess, Ressource) sind die folgenden *Qualitätsbewertungsmethoden* ISO9000, CMM, ISO9126 und PSP anzuwenden.

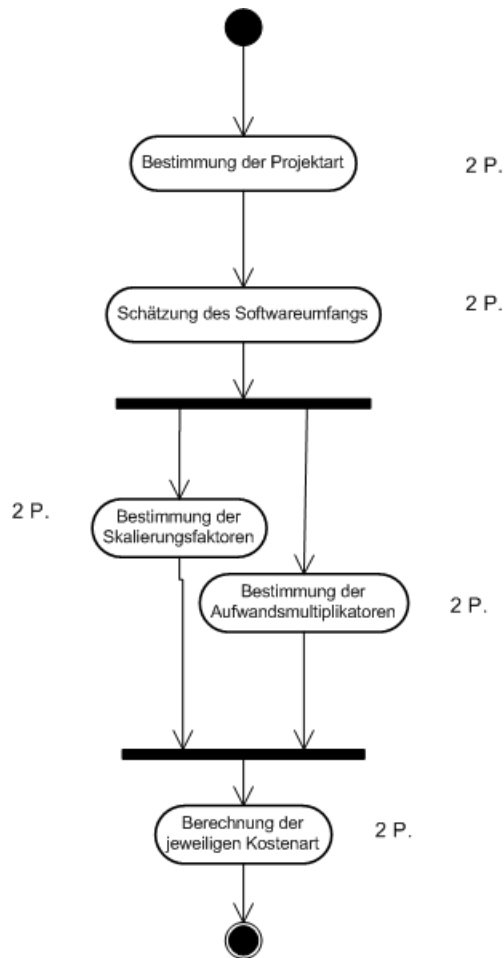
ISO9126 (Produkt), (1 P.)

ISO9000, CMM (Prozess) (2 P.)

PSP (Ressource) (1 P.)

Aufgabe 13 (10 Punkte):

Zeichnen Sie für die Aufwandschätzung nach der COCOMO-Methode ein *Aktivitätsdiagramm*.



Aufgabe 14 (4 Punkte):

Erläutern Sie die folgenden Begriffe zur *komponentenbasierten Software-Entwicklung*:

- a) COTS,
- b) Black-Box-Wiederverwendung,
- c) implizite Wiederverwendung,
- d) interne Wiederverwendung.

COTS: kommerzielle Komponenten (1 P.)

Black-Box: Übernahme ohne Veränderung (1 P.)

implizite: „eingebettete“ Komponenten (1 P.)

interne: Komponenten aus der eigenen Firma bzw. im eigenen Team (1 P.)

Aufgabe 15 (5 Punkte):

Nennen Sie 5 *UML-Diagramme*, die sich vorwiegend auf das objektorientierte Entwicklungsparadigma beziehen.

Klassendiagramm, Kollaborationsdiagramm, Objektdiagramm, Komponentendiagramm, Verteilungsdiagramm (oder auch Paketdiagramm und Sequenzdiagramm) (5 P.)



Prüfungsklausur Softwaretechnik I

B

Bewertung

Aufgabe 1 (2 Punkte):

Für die phasenbezogene Software-Entwicklung (Problemdefinition und Erprobung ausgenommen) seien die anteiligen Aufwände bekannt (Anforderungsanalyse 10%, Spezifikation 25%, Entwurf 30% und Implementation 35%). Bestimmen Sie den Zeitaufwand für die Implementation nach der *Analogiemethode* in Personenmonaten, wenn in einem konkreten Projekt für die ersten beiden Phasen (Anforderungsanalyse und Spezifikation) bereits 6 Monate benötigt wurden.

$$t(A + S = 35\%) = 6 \text{ Monate}, \quad t(I = 35\%) = 6 \text{ Monate} \quad (2 P.)$$

Aufgabe 2 (2 Punkte):

Welche *Eigenschaftstransformation einer Anforderung* wird durch die jeweilige Methode der Anforderungsanalyse erreicht

- a) Experiment, b) Analogieschluss.
subjektiv → *objektiv* (1 P.) *nicht messbar* → *messbar* (1 P.)
(auch andere)

Aufgabe 3 (3 Punkte):

Welche Anforderungsart wird hauptsächlich in der *Implementation* umgesetzt? Nennen Sie zwei Ergebnisse der Implementationsphase.

qualitative Anforderungen (1 P.)
Ergebnisse: Programmcode, Testdokumentation (2 P.)
(oder: *Entwicklerdok., Installationstool*)

Aufgabe 4 (4 Punkte):

Mit welcher Komponente wird beim Entwurf auf der Grundlage einer Komponentenhierarchie begonnen bei

- a) *einem Trial-and-Error-Entwurf?* b) *einem Bottom-Up-Entwurf?*
alle Komponente in *die unteren Komponenten*
erster „Näherung“ (2 P.) *einer Hierarchie* (2 P.)

Wann ist der jeweilige Ansatz zweckmäßig?

- bei geringen System- oder Techno-* (1 P.) *bei bereits vorhandenen Komponenten* (1 P.)
logiekenntnissen

Aufgabe 5 (6 Punkte):

Charakterisieren Sie kurz die Begriffe der *Architektur*, des *Paradigmas* und der *Programmkonvention*.

Architektur: Zusammensetzung der Systemstruktur aus Komponenten (2 P.)
Paradigma: Art und Weise der Modellierung und Realisierung (2 P.)
Programmierkonvention: spezielle Vorschriften zur Form und Gestaltung der Programme (2 P.)

Aufgabe 6 (4 Punkte):

Wo werden die Testdaten für den *Black-Box-Test* definiert? Nennen Sie drei *Visualisierungsformen* von Programmen beim *White-Box-Test*.

Black-Box-Testdaten werden in der Spezifikationsphase definiert (1 P.)

*Programmvisualisierungen für White-Box-Test: Flussgraph, Callgraph, Datenflussgraph (3 P.)
(ggf. auch Funktionsbaum, Struktogramm, Aktivitäts-, Sequenz- und Klassendiagramm)*

Aufgabe 7 (2 Punkte):

Geben jeweils eine andere Bezeichnung für den *Alphatest* und den *Betatest* an.

Alphatest = Systemtest bzw. Integrationstest (1 P.)

Betatest = Erprobung bzw. Akzeptanztest (1 P.)

Aufgabe 8 (2 Punkte):

Charakterisieren Sie die Personalstruktur einer *agilen Software-Entwicklung*.

zwei Programmierer, (1 P.)

ein (Vetreter des) Auftraggeber (1 P.)

Aufgabe 9 (2 Punkte):

Welcher Anforderungsart sind jeweils die Ursachen für die folgende *Wartungsaktivität* zuzuordnen

a) Anpassung,

b) Erweiterung.

systembezogene Anforderungen (1 P.)

funktionale Anforderungen (1 P.)

Aufgabe 10 (8 Punkte):

Nennen Sie jeweils zwei Beispiele für ein *CASE-*, *CARE-* und *CAME-Tool*. Geben Sie zwei Komponenten einer Entwicklungsinfrastruktur an, die über eine Programmierumgebung hinausgehen.

CASE: Rational, Textverarbeitung (usw.) (2 P.)

CARE: Together, Disassembler (2 P.)

CAME: Logiscope, Together (2 P.)

Entwicklungsinfrastruktur: Community, Erfahrungshintergrund (oder auch Standards) (2 P.)

Aufgabe 11 (2 Punkte):

Geben Sie jeweils ein Beispiel einer *Software-Metrik* bzw. eines *Software-Maßes* an für

a) eine Verhältnisskalierung,

b) eine ordinale Skalierung.

COTS-Preis (1 P.)

System-Ranking (oder CMM) (1 P.)

Aufgabe 12 (4 Punkte):

In welchem Bereich (Produkt, Prozess, Ressource) sind die folgenden *Qualitätsbewertungsmethoden* ISO9126, CMM, ISO9000 und PSP anzuwenden.

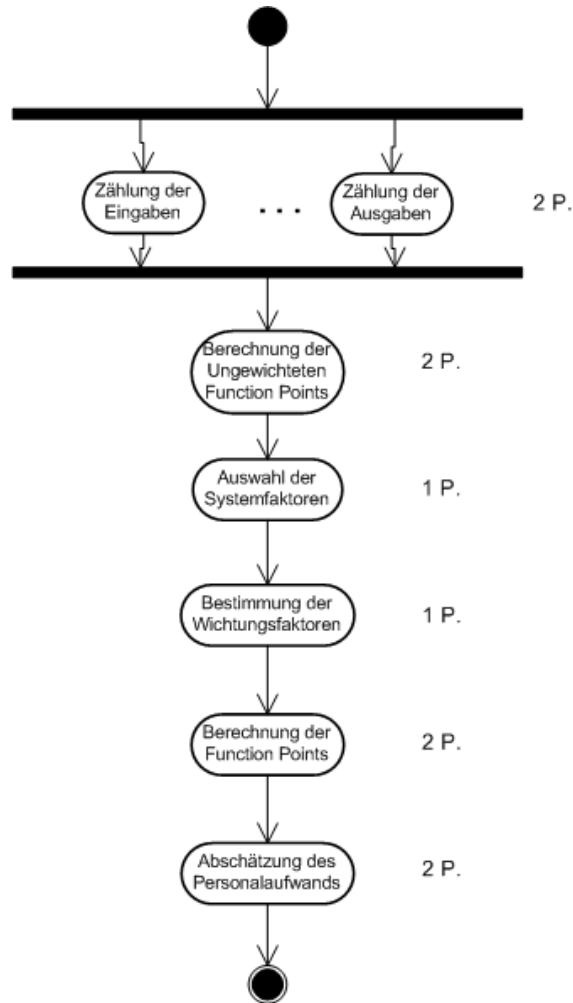
ISO9126 (Produkt), (1 P.)

ISO9000, CMM (Prozess) (2 P.)

PSP (Ressource) (1 P.)

Aufgabe 13 (10 Punkte):

Zeichnen Sie für die Aufwandschätzung nach der Function-Point-Methode ein *Aktivitätsdiagramm*.



Aufgabe 14 (4 Punkte):

Erläutern Sie die folgenden Begriffe zur *komponentenbasierten Software-Entwicklung*:

- a) COTS,
- b) White-Box-Wiederverwendung,
- c) explizite Wiederverwendung,
- d) externe Wiederverwendung.

COTS: kommerzielle Komponenten (1 P.)

White-Box: Verwendung mit Veränderung (1 P.)

explizite: im Produkt sichtbare Komponente (1 P.)

externe: Komponenten aus anderen Firmen oder Bereichen (1 P.)

Aufgabe 15 (5 Punkte):

Nennen Sie 5 *UML-Diagramme*, die sich nicht explizit auf das objektorientierte Entwicklungsparadigma beziehen.

Use-Case-Diagramm, Zustandsdiagramm, Aktivitätsdiagramm, Sequenzdiagramm, Verteilungsdiagramm (oder auch Komponentendiagramm) (5 P.)