

### Aufgabe 12-1: Verständnis- und Vertiefungsfragen

Beantworten Sie die folgenden Fragen.

- a) Grenzen Sie folgende Begriffspaare gegeneinander ab:
1. konstruktive Qualitätssicherung – analytische Qualitätssicherung
  2. (Softwareentwicklungs-)Prozess – Prozessmodell
  3. Projektmanagement – Prozessmanagement
- b) Nennen und erklären Sie (kurz) die wichtigsten Konzepte mit denen ein Softwareentwicklungsprozess (oder auch -modell) beschrieben wird?
- c) Beantworten Sie folgende Fragen zum CMM:
1. Ist CMM ein Prozessmodell? Begründen Sie.
  2. Auf welcher Reifestufe erfolgt der Wechsel von Projektmanagement zu Prozessmanagement?
  3. Ab welcher Reifestufe wird der Softwareentwicklungsprozess kontinuierlich angepasst und verbessert?
- d) Worin besteht der wesentliche Unterschied zwischen einem Prozess mit einer Abnahmekontrolle und Total Quality Management (TQM)?
- e) Allgemein: Was ist die Idee iterativer Modelle? Welche Probleme versucht man hiermit zu lösen? Nennen Sie mindestens vier Vorteile, die sich durch die Einführung von Iterationen ergeben.  
Konkreter: Welchen Zweck verfolgt man mit dem Bau von Prototypen?
- f) Welche Erweiterung nehmen agile Modelle zusätzlich zur Einführung von Iterationen vor? Welche Vorteile ergeben sich hieraus *zusätzlich* zu den von Ihnen in **e)** genannten?
- g) Warum wird Extreme Programming (XP) als ein besonders diszipliniertes Prozessmodell bezeichnet?

### Aufgabe 12-2: Auswahl von Prozessmodellen

- a) Fassen Sie noch einmal kurz die Unterschiede zwischen den drei Prozessmodellarten *wasserfallartig*, *evolutionär*, und *inkrementell* zusammen.
- b) Wählen Sie für jedes der folgenden zu bauenden Systeme die am besten geeignete Prozessmodellart (aus den in **a)** genannten) und begründen Sie Ihre Wahl.
1. Ein Terminal als digitaler, interaktiver Ersatz für Papierfahrpläne in größeren Bahnhöfen (für Ankunft- und Abfahrtszeiten).
  2. Eine Steuerungseinheit für das Antiblockiersystem (ABS) in PKWs.
  3. Ein System zur Verwaltung von Lehrveranstaltungen, wie etwa unser KVV.

### **Aufgabe 12-3: Extreme Programming**

Füllen Sie den folgenden Lückentext mit passenden Begriffen. Falls nötig, nutzen Sie dazu die auf der Vorlesungswebseite angegebenen **Quellen**.

Zeitlich ist ein Projekt, das mit Extreme Programming (Abk. \_\_\_\_\_) durchgeführt wird, in \_\_\_\_\_ eingeteilt. Diese enden jeweils mit einer neuen Version des Softwaresystems. Am \_\_\_\_\_ einer jeden Iteration besprechen der Kunde und die Entwickler gemeinsam, welche Funktionalitäten realisiert werden sollen. Der Kunde formuliert dabei seine Wünsche auf den \_\_\_\_\_ (engl.). Während der gesamten Entwicklung ist der Kunde \_\_\_\_\_. Er definiert zudem \_\_\_\_\_, um am \_\_\_\_\_ jeder Iteration die Funktionalität testen zu können. Für die Entwickler gibt Extreme Programming zusätzlich eine Reihe von Praktiken vor, wie etwa \_\_\_\_\_, \_\_\_\_\_ oder die gemeinsame Verantwortung.

### **Aufgabe 12-4: Wasserfallkritik**

In den letzten fünfzehn Jahren wurde das Wasserfallmodell stark kritisiert und mehr „Agilität“ gefordert. Zuvor war jedoch ein wasserfallartiges Vorgehen (fast) immer als das ideale Vorgehen beschrieben worden.

- a)** Erklärung Sie diese Entwicklung. Warum kann man das Wasserfallmodell einerseits durchaus als *ideal* bezeichnen?
- b)** Und warum hat man sich andererseits dennoch davon gelöst? Nennen Sie mindestens zwei Punkte, an denen Wasserfall-Projekte scheitern können, und die bei agilen Projekten zumindest stark abgedeckt werden.