

**Softwaretechnik SoSe 2012 Übungsblatt 11: Konstruktive QS und Prozessmodelle**  
 Bearbeitung zum Tutorium in KW 26

Bereiten Sie Ihre **Lösungen** grundsätzlich so vor, dass Sie diese in der Übung Ihren Kommilitonen in geeigneter Form **zeigen** und **diskutieren** können. Geben Sie bitte stets Ihre verwendeten **Quellen** an.

**Aufgabe 11-1: (Verständnis- und Vertiefungsfragen)**

Beantworten Sie die folgenden Fragen kurz (pro Stichwort ein Satz).

1. Was ist ein Prozess, was ist ein Prozessmodell und was ist ein Prozessmanagement?
2. Ab welcher CMM-Reifestufe wird der Softwareentwicklungsprozess kontinuierlich angepasst und verbessert?
3. Was ist das Besondere an Total Quality Management (TQM)?
4. Was ist der Unterschied zwischen einem evolutionären Prozess und einem inkrementellen Prozess?
5. Wozu werden Prototypen gebaut?
6. Welche Besonderheiten hat ein Rational Unified Process (RUP), der wasserfallentartet ist?
7. Was verdeutlicht die V-Form der Darstellung des V-Modells?
8. Welche Vorgehensweisen verhindern bei agilen Prozessen (im Gegensatz zu wasserfallartigen Prozessen), dass man am Ende ganz ohne lauffähige Software da steht?
9. Warum wird Extreme Programming als besonders diszipliniert bezeichnet?

**Aufgabe 11-2: (Auswahl von Prozessmodellen)**

Im Folgenden betrachten wir drei grundsätzliche Arten von Prozessmodellen: Wasserfallartig, evolutionär und inkrementell. Wählen Sie für jedes der folgenden zu bauenden Systeme die Ihrer Meinung nach am besten geeignete Prozessmodellart und begründen Sie Ihre Wahl.

1. Die Steuerung für ein Antiblockiersystem (ABS) für Autos
2. Ein Learning Management System
3. Eine interaktive Selbstbedienungskonsole für große Bahnhöfe, an dem man die aktuellen Ankunfts- und Abfahrtszeiten von Zügen einsehen kann.

**Aufgabe 11-3: (Extreme Programming)**

Füllen Sie den folgenden Lückentext mit passenden Begriffen und Wörtern. Nutzen Sie dazu falls nötig die auf der Vorlesungswebsite angegebenen Quellen.

*Ein Projekt, das mit Extreme Programming (Abk. \_\_\_\_\_), durchgeführt wird, läuft grundsätzlich in Iterationen ab. Jede \_\_\_\_\_ endet mit einer neuen Softwareversion, in der einige neue Funktionalitäten zu dem System hinzugefügt wurden. Welche dies sind, schreibt der Kunde auf so genannte \_\_\_\_\_ (engl.). Am \_\_\_\_\_ einer jeden Iteration*

besprechen der \_\_\_\_\_ und die Entwickler, welche Funktionalitäten in ihr realisiert werden sollen. Auch während der Entwicklung ist der Kunde immer \_\_\_\_\_. Er definiert zudem \_\_\_\_\_, um am \_\_\_\_\_ jeder Iteration die Funktionalität zu testen. Auf Entwicklerseite gibt es auch einige besondere Praktiken, z.B. \_\_\_\_\_, \_\_\_\_\_ oder die gemeinsame Verantwortung.

#### **Aufgabe 11-4: (Wasserfallkritik)**

In den letzten fünf bis zehn Jahren wurde das Wasserfallmodell zunehmend kritisiert und mehr „Agilität“ gefordert. Vorher jedoch war ein wasserfallartiges Vorgehen immer als das ideale Vorgehen beschrieben worden, an das man sich möglichst halten sollte. Wie erklären Sie sich diese Entwicklung, d.h. wie kam es vermutlich zu dem Wasserfall als Ideal und warum löste man sich vermutlich später davon?