

## A1 Scrum: Akzeptanzkriterien vs. Definition of Done

Erläutern Sie mit eigenen Worten den Unterschied zwischen den Akzeptanzkriterien einer User Story und der Definition of Done bei Scrum.

### Lösung 1

## A2 Continuous Deployment

Nennen Sie mögliche Risiken einer Continuous-Deployment-Pipeline, die ein Softwaresystem über mehrere Schritte hinweg bis ins Produktivsystem einspielt.

### Lösung 2

## Klassisch oder agil?

Eine medizinische Gemeinschaftspraxis mit mehreren Ärzten:innen und Arzthelfern:innen möchte Sie als Softwaredienstleister beauftragen, ein Online-Reservierungssystem für Impfungen zu entwickeln. Es soll ein Festpreis für die Lieferung der Software vereinbart werden. Der Auftraggeber hat bisher kaum Erfahrung mit Softwareprojekten gemacht. Außerdem stellen Sie schon im ersten Gespräch fest, dass es verschiedene Auffassungen über den Funktionsumfang und die Details der fachlichen Anforderungen gibt.

Geben Sie an, welche Vorgehensweise Sie als Projektleitung für dieses Projekt empfehlen würden. Diskutieren Sie die Chancen und Risiken des Einsatzes von Scrum für dieses Projekt.

### Lösung 3

## A4 Online-Reservierungssystem für Impfungen

Sie erheben nun die Anforderungen für das Online-Reservierungssystem für Impfungen, siehe Aufgabe A3. Sie wollen zunächst auf fachlicher Ebene den Nachrichtenfluss zwischen Patienten und Web-Anwendung modellieren. Sie haben bereits die folgenden Komponenten identifiziert:

- Der Patient als menschlicher Akteur
- Die Web-Anwendung als GUI
- Ein Identitätsmanagement als Hintergrunddienst
- Ein Termin-Verwaltungssystem als Hintergrunddienst

Der reguläre Nachrichtenfluss zwischen diesen Komponenten gestaltet sich wie folgt:

- Der Patient besucht die Web-Anwendung und meldet sich mit seinem Benutzernamen und Passwort an (Sie können davon ausgehen, dass der Patient bereits registriert ist).
- Die Webseite prüft die Gültigkeit der Anmeldung im Identitätsmanagement.
- Bei ungültigen Anmeldedaten wird dem Patienten durch die Web-Anwendung angeboten, sein Passwort zurückzusetzen. Dazu wird eine entsprechende Operation im Identitätsmanagement aufgerufen. Der Nachrichtenfluss endet dann.
- Ansonsten ruft die Web-Anwendung die möglichen Wochentage für eine Impfung vom Termin-Verwaltungssystem ab und zeigt diese an.
- Der Patient wählt einen der möglichen Tage aus. Die Web-Anwendung ruft dann die möglichen Uhrzeiten vom Termin-Verwaltungssystem ab und zeigt diese an.
- Der Patient wählt eine Uhrzeit aus und bestätigt die Buchung. Dadurch wird der Termin im Termin-Verwaltungssystem reserviert und im persönlichen Kalender im Identitätsmanagement eingetragen.
- Falls bei der Buchung angegeben, erhält der Patient optional eine E-Mail mit dem gebuchten Termin. Diese wird vom Identitätsmanagement gesendet.

Modellieren Sie den Nachrichtenfluss zwischen Patienten und den beteiligten Systemen durch ein UML-Sequenzdiagramm.

## Lösung 4

### A5 Auto starten

Im Zip-Archiv »sequence.zip« finden Sie eine Java-Konsolenanwendung für den Start eines Autos, siehe ILIAS. Die Anwendung besteht aus den Klassen App, Battery, Car, Engine und OilSensor.

Modellieren Sie, ausgehend vom Aufruf der Methode main in der Klasse App, den Nachrichtenfluss der Anwendung als UML-Sequenzdiagramm. Berücksichtigen Sie dabei folgende Hinweise:

- Der Aufruf von main soll als »gefundene Nachricht« ohne Parameter modelliert werden.
- Die korrekten, verschachtelten Sequenznummern der Aufrufe sollen angegeben werden.
- Alle Datentypen, Parameter und Rückgaben sollen angegeben werden.

- Antwort-Nachrichten können weggelassen werden, soweit sie sich eindeutig aus den Aufrufen ergeben.
- Die Konsolenausgabe der App als Reaktion auf die Rückgabe von `myCar.prepareStart()` brauchen Sie nicht zu modellieren.

## **Lösung 5**