Einführung in Software-Engineering – Übung 7

Aufgabe 1)

Die nachfolgende Schätzung der Komplexität soll mit Hilfe von Story-Points beschrieben werden. Dabei dienen die fünf nachfolgenden Beispiele auch dafür die Höhe der Story-Points mit zu bestimmen, da man diese in Abhängigkeit zueinander sehen kann und so die Implementierung, welche am wenigsten Zeit in Anspruch nehmen wird, den geringsten Wert bekommt und alle weiteren relativ davon gesehen mehr Aufwand benötigen. Dabei entspricht der Wert eines Story-Points **30 Minuten**.

User-Story 1: Es soll eine Stoppuhr laufen und angezeigt werden, während der Lernende die Frageseite der Flashcards betrachtet. Die Schätzung beläuft sich hier auf zwei Story-Points für die Implementierung, da wir diese als nicht so aufwändig einschätzen. Für das Testen werden wiederum 2 Story-Points dazugezählt, da auch dieses nicht sehr komplex sein sollte. Insgesamt ergibt sich eine Schätzung von **vier Story-Points**.

User-Story 2: Es ist eine Statistik für jede Flashcard zu pflegen, die speichert, wann die Karte gelernt wurde (0,5 Story-Points), ob die Karte positiv gelernt wurde (0,5 Story-Points), wie oft die Karte gelernt wurde (0,5 Story-Points), wie lange der Lernende die Karte betrachtet hat (0,5 Story-Points) und wie oft hintereinander die Karteikarte positiv gelernt wurde (0,5 Story-Points). Die Statistik soll permanent angezeigt und aktualisiert werden (0,5 Story-Point) und gespeichert werden (0,5 Story-Points). Für die Angabe, wie häufig eine Antwort zu einer Karte gewusst wurde, soll es auch eine farbliche Auswertung geben (2 Story-Points). Zudem soll ein Hinweis erscheinen, wenn der Lernende den Vorgang abbrechen möchte, ohne vorher die neue Statistik gespeichert zu haben (0,5 Story-Points). Insgesamt ergeben sich also 6 Story-Points für die Implementierung. Für das Testen erwarten wir wiederum einen Aufwand von 4 Story-Points, da dies relativ aufwendig zu sein scheint. Insgesamt ergeben sich so **10 Story-Points.** Dies erscheint uns realistisch, wenn man den erwarteten Aufwand im Verhältnis zur ersten User-Story sieht.

User-Story 3: Das Einstellen der Anzahl der zu lernenden Karten soll möglich werden (1 Story-Points), genauso wie das Lernen einer zufälligen Karte mit (2 Story-Points) oder ohne Zurücklegen (1 Story-Point) der Flashcards. Zudem sollen die Karten nach dem Vorbild der physischen Karten in fünf verschiedene „Fächer“ unterteilt werden können, die jeweils danach sortiert werden, wann sie das letzte Mal positiv gelernt wurden (3 Story-Points). Neu erstellte Karten werden in das erste Fach einsortiert (1 Story-Point) und bei jedem erfolgreichen Beantworten der Frage wird die Flashcard in das nächste Fach einsortiert (1 Story-Point).Für diese User-Story sollten somit 9 Story-Points für die bisher wohl aufwändigste Implementierung als Komplexitätsschätzung herangezogen werden. Für das Testen erwarten wir zusätzlich noch 3 Story-Points, was insgesamt betrachtet zu einem Ergebnis von **12 Storys-Points** führt.

User-Story 4: Es kann hier insgesamt von **9 Story-Points** ausgegangen werden. Die Implementierung der Suchfunktion soll das Suchen nach einem Begriff auf den Vorderseiten der Flashcards möglich machen (4 Story-Points), sodass danach alle Karteikarten angezeigt werden, die den gesuchten Begriff beinhalten. Also müssen alle Vorderseiten der Karten einzeln durchgegangen werden und mit Hilfe von Stringvergleichen können die Ergebnisse gefiltert werden. Dies schätzen wir als relativ komplex ein. Des Weiteren soll eine Sortierung der Karteikarten nach drei verschiedenen Kriterien ermöglicht werden. Demnach kann die Sortierung nach den einzelnen Kriterien jeweils für sich als

eigene Methode aufgefasst werden (3 * 1 = 3 Story-Points). Insgesamt ergeben sich hier also grob vier Teilaufgaben, die jeweils unterschiedlich aufwändige Zeitintervalle benötigen. Für das Testen kommen zusätzlich 2 Story-Points hinzu.

User-Story 5: Hier soll es möglich werden das Erstellen von Karteikarten, das Löschen von diesen und die Editierungen der Karten rückgängig zu machen(4 Story-Points). Das Wiederherstellen von den rückgängig gemachten Änderungen soll ebenfalls unterstützt werden (2 Story-Points). Zudem soll es keine Begrenzung auf eine Anzahl der Änderungen, die rückgängig gemacht werden sollen, geben. Insgesamt ergibt sich hier eine Schätzung der Komplexität auf 6 Story-Points allein für die Implementierung, da jeweils für drei verschiedene Anwendungen auf Karteikarten Änderungen gespeichert werden müssen und ebenfalls deren Wiederherstellung in Bezug auf diese Veränderungen ermöglicht werden. Zudem kommen wieder zwei Story-Points für das Testen hinzu. Insgesamt ergeben sich so **8 Story-Points**. Dies erscheint uns realistisch, da es wohl in etwa mit dem Aufwand der User-Story 4 verglichen werden kann.

Aufgabe 2)

a) Testplan:

Beschreibung des Testfalles	**Erwartetes Ergebnis**
„0 seconds“ beim ersten Start.	Der Timer zeigt beim Start des Lernens 0 Seconds an.
„0 seconds“ beim Weiterklicken der Karteikarte	Der Timer zeigt auch bei den weiteren Karteikarten (beim Weiterklicken) zu Beginn 0 Seconds an.
„0 seconds“ bei erneutem Lernen	Nach Abbruch des Lernens und erneutem Starten des Lernprozesses zeigt der Timer einen Startwert von 0 an.
Timer stoppt nach Antwort	Nachdem die Antwort angezeigt wird und die eigentliche Lernphase beendet wurde, stoppt die Zeit und beginnt auch erst wieder, nachdem eine neue Frage angezeigt wird.

b) In Aufgabenteil 1a haben wir als Schätzung der Komplexität angegeben, dass diese zusammen mit dem Testen auf vier Story-Points käme. Nach der Implementierung derselben Aufgabe erscheint es nun ein größerer Aufwand zu sein. Als neue Schätzung legen wir nun 6 Story-Points zu Grunde, da wir ca. drei Stunden an der Implementierung gearbeitet haben.

Die **Velocity** berechnet sich mit der folgenden Formel:

Velocity = benötigte Zeit / Anzahl der vergebenen Story-Points

Für die User Story 1ergibt sich so: 180 Min / 4 Story-Points = 45 Min/Story-Points

Daraus kann nun eine **verbesserte Schätzung** der anderen User-Storys resultieren (hier in Minuten angegeben):

User Story 2: Statt den geplanten 300 Minuten ergeben sich hier 10 Story-Points * 45 Min/Story-Points = 450 Minuten

User Story 3: Statt den geplanten 360 Minuten ergeben sich hier 12 Story-Points * 45 Min/Story-Points = 540 Minuten

User Story 4: Statt den geplanten 270 Minuten ergeben sich hier 9 Story-Points * 45 Min/Story-Points = 405 Minuten

User Story 5: Statt den geplanten 240 Minuten ergeben sich hier 8 Story-Points * 45 Min/Story-Points = 360 Minuten

Aufgabe 3)

a) User-Story:

„Die Flashcards-Anwendung soll nun auch das Austauschen von Karten mit andern Lernenden unterstützen. Dafür sollen einzelne Karten oder ganze Sets von Karteikarten als Datei gespeichert werden können und solche ebenfalls eingebunden werden können. Weiterhin soll es möglich sein, diese neuen Karten in bestehende Themen zu integrieren und dabei pro Set Duplikate, also Karten mit gleichen Fragestellungen, zu entfernen. Bevorzugt sind dabei die eigenen Flashcards zu behalten."

Für die Implementierung dieser Funktionalität gehen wir von einem Aufwand von **10 Story-Points** aus, da zum einen die Funktion zum Speichern der Karten und –Sets ermöglicht werden soll (2 Story-Points) sowie das Einbinden von fremden Karten in die Flashcards-Anwendung (2 Story-Points). Des Weiteren müssen diese Karten in Themen integriert werden können (2 Story-Points) und danach mögliche Duplikate gelöscht werden (2 Story-Points). Zudem werden Tests benötigt, die jedoch nicht allzu kompliziert ausfallen, sodass diese ebenfalls mit 2 Story-Points angesetzt werden können. Auch hier entspricht der Wert eines Story-Points 30 Minuten.

b) Implementierung und Tests siehe Quellcode.