

# Software Engineering Übung 7:

## Gruppe:

1. Michael Scholz (matr.: 1576630, rbg: mi48azih)
2. Ulf Gebhardt (matr.: 1574373, rbg: hu56nifa)
3. Sebastian Weicker (matr.: 1625099, rbg: we87obyq)

## Aufgabe 1: Erfahrungen mit der Planung von Softwareprojekten sammeln:

Ein Story-Point (sp) ist 30min wert.

### 1) *User Story 1 – Stoppuhr:*

Geschätze Story-Points : 2 Story Points

#### **Begründung:**

Da wir im GDI-I-Praktikum schon mit dem swing-Timer gearbeitet haben, schätzen wir den Aufwand den Timer zu implementieren gering ein (0,5sp).

Für die Anzeige in der GUI ist mehr Aufwand nötig (1sp), da wir uns mit der Karten-GUI noch auseinandersetzen müssen. Außerdem wissen wir noch nicht wie wir die Zeit formatieren müssen, damit wir sie anzeigen können.

Tests (0,5sp), da es sich um eher rudimentäre Funktionalität handelt, die sich leicht testen lässt.

### 2) *User Story 2 - Statistik & Farbcodierung:*

Geschätze Story-Points : 4 Story Points

#### **Begründung:**

Erheblicher Programmieraufwand, zur Verwaltung der Kartenhistorie und der Kartenstatistik (2,5sp). Tests werden aufwendig, da noch nicht überschaubar und komplexe Funktionalität (1,5sp).

Hinzuzufügen ist, dass nicht klar spezifiziert wird, was als häufig, selten usw. Angesehen wird.

### 3) *User Story 3 – Lernen mit unterschiedlichen Systemen:*

Geschätze Story-Points : 5 Story Points

#### **Begründung:**

Die Userstory wurde von uns mit 5 sp bewertet, da die beschriebenen Funktionalitäten erhebliche Eingriffe in das System erfordern. Die Implementierung

wird sich als aufwendig erweisen (3sp) und es werden einige Tests zu schreiben sein (2sp).

#### 4) User Story 4 „Suchen und Sortieren“:

Geschätze Story-Points : 4 Story Points

##### **Begründung:**

Sortierfunktionen gibt es in Java bereits für Datum und Integerzahlen und wir können darauf zurückgreifen.(1.5sp). Die Anzeige in der GUI zu implementieren ist überschaubar (1sp)

Tests sind aufwändig, da die Suche eine sehr große Eingabemenge hat und viele Sonderfälle (z.B. Leere suche, sonderzeichen usw.)(1.5sp)

#### 5) User Story 5 – Undo & Redo:

Geschätze Story-Points : 3.5 Story Points

##### **Begründung:**

Undo & Redo ist relativ schwer zu implementieren, nach unserer Einschätzung, da auch bereits rückgängig gemachte Änderungen wieder hergestellt werden müssen (2sp). Die Testfälle werden entsprechend komplex ausfallen (1.5 sp).

### Aufgabe 2: Implementierung einer User-Story zur Bestimmung der velocity:

a)  siehe Code

#### Testplan:

Test	Durchführung	Erwartetes Ergebnis
<b>Timer startet immer bei 0 beim starten des Lernprozesses</b>	<ul style="list-style-type: none"><li>• Der Lernprozesses wird mit einige Karten gestartet</li><li>• Überprüfen, ob Timer bei 0 startet (Timer-Anzeige)</li></ul>	Timer steht zu Beginn des Lernprozesses (Anzeige der ersten Frage) auf 0
<b>Timer startet bei 0 bei neuer Fragekarte</b>	<ul style="list-style-type: none"><li>• Der Lernprozesses wird mit einige Karten gestartet</li><li>• Einige Zeit (&gt;15sec) warten</li></ul>	Timer fängt bei jeder neuen Fragekarte vom Wert 0 an zu zählen.

	<ul style="list-style-type: none"> <li>• Zur nächsten Fragekarte schalten</li> <li>• Überprüfen ob Timer bei 0 startet (Timer-Anzeige)</li> </ul>	
<b>Timer zählt richtig(jede Sekunde)</b>	<ul style="list-style-type: none"> <li>• Der Lernprozesses wird mit einige Karten gestartet, gleichzeitig mit einem Zeitmessinstrument (z.B. Stopuhr)</li> <li>• Vergleich der Messung mit Anzeige nach einigen Sekunden.</li> </ul>	Der Timer zählt sekundlich
<b>Auch bei mehren Startvorgängen des Lernprozesses startet der Timer bei 0</b>	<ul style="list-style-type: none"> <li>• Der Lernprozesses wird mit einige Karten gestartet</li> <li>• Lernprozess beenden</li> <li>• Erneut Karten auswählen und Lernprozess starten</li> <li>• Überprüfen ob Timer bei 0 startet (Timer-Anzeige)</li> </ul>	Timer steht auch beim erneuten Starten des Lernprozesses zu begin auf 0

**b) Velocity-Berechnung:**

Benötigte Zeit (User-Story 1):

- Timer Code: 10 Min
- GUI Code: 10 Min
- Ausgabe in GUI: 5 Min
- Testen: 5 min

**Verbesserte Schätzung:**

Die Implementierung der User-Story 1 hat uns 30 Minuten gekostet, statt, wie anfangs geschätzt, eine Stunde. Dies ist auf unsere Erfahrung mit dem `java.swing.Timer` aus dem GDI I-Projekt zurückzuführen, weshalb wir unsere Velocity für die übrigen User-Stories

nicht ändern würden. Würden wir davon ausgehen, dass wir alle übrigen User-Stories auch schneller als geschätzt lösen könnten, so würde ein Story-Point nur noch 15min wert sein. Dies scheint uns allerdings eher unrealistisch, weshalb wir lieber die Story-Points von User-Story 1 (Timer) von zwei sp auf einen sp herabsetzen.

### **Aufgabe 3: Erarbeiten einer User-Story und Verbesserung der Schätzung der velocity:**

#### **a) User-Story:**

Es soll möglich seine Kartensets mit Freunden auszutauschen. Diese Kartensets sollen beim importieren in ein vorhandenes Kartenset integriert werden.

Sollten Karten doppelt vorhanden sein (im eigenem Kartenset & im importierten Kartenset), dürfen diese nicht importiert werden. Die eigene Karte des Benutzers wird behalten, wenn die Karten doppelt auftaucht, die Karten aus dem zu importierenden Sets wird verworfen.

Karten gelten dann als doppelt, wenn die Frageseite identisch ist. Die Antwortseite wird dabei nicht überprüft.

Die Meta-Daten der Karteikarte (Lernerfolg usw.) wird nicht übernommen.

#### **b) siehe Code**