

Software-Engineering-Übung 2

Aufgabe 1

Das hier vorliegende Essay befasst sich mit der Thematik der häufig auftretenden Probleme beim Management von Software-Projekten und basiert auf den von uns gesammelten Projekt-Erfahrungen des gemeinsam absolvierten Praktikums der Veranstaltung „Grundlagen der Informatik I“. Zudem bilden die Texte „2010 IT Project Success Rates“ von Scott W. Ambler sowie John S. Reel's „Critical Success Factors in Software Projects“ eine Grundlage für die Abwägung von den wichtigsten Faktoren, die zu einem erfolgreichen Verlauf von Software-Projekten führen können.

Das benannte Praktikum, eine Implementierung des Spieles Sokoban, wurde in einer Gruppe von vier Studierenden bearbeitet und würde somit nach der Definition von Ambler als kleines Team beschrieben. Dass eine kleinere Teamgröße bei Projekten jeglicher Art meistens mit einer höheren Wahrscheinlichkeit erfolgreich abgeschlossen werden, kann von uns bejaht werden, da es sich schon bei uns vieren des Öfftern als schwierig herausstellte einen gemeinsamen Termin zu finden und die Arbeitsaufteilung günstig zu verteilen. Ebenfalls aus dem Text hervorzuheben ist, dass die disziplinierten Formen der Projektausführung, so wie die agile oder iterative Vorgehensweise sich als effektiver für alle Teamgrößen herausgestellt haben. Dennoch hat sich bei unserem Projekt die ad-hoc-Software-Entwicklung durchgesetzt, also wurde von uns keinem vorher genau festgelegtem Prozess gefolgt, sondern die in der Aufgabenstellung aufgelisteten Anforderungen wurden nacheinander je nach Fortschritt des Projektes, des Schwierigkeitsgrades der Teilaufgabe und der verbleibenden Zeit bearbeitet. Laut der erstellten Statistik funktioniert bei kleinen Teams diese Arbeitsweise auch besser als die alternative traditionelle Vorgehensweise. Gemäß der Umfrage werden hierbei rund 49% der Projekte erfolgreich abgeschlossen und bei weiteren 37% wird eine laufende Softwareversion erstellt, die jedoch nicht alle Erfolgskriterien erfüllt. Da ein „erfolgreicher Abschluss“ auch immer davon abhängt, wie die Kriterien definiert sind, würden wir unser Projekt als erfolgreich beschreiben, da unsere Erwartungen an einen positiven Verlauf erfüllt wurden, obgleich wir nicht die maximale Punktzahl erreichten.

Aufgabe 2

Bewertung der nachfolgenden nicht-funktionalen Anforderungen danach, ob diese erkennen lassen, wann die Anforderung erfüllt ist:

- 1.) „Die Flashcard-Anwendung muss als Inhalt der Karteikarten alle gängigen Bildformate unterstützen.“

➔ Fraglich ist was unter „alle gängigen Bildformate“ verstanden wird. Manche Nutzer verstehen darunter mehr als andere User, da es immer auf den Kenntnisstand der Personen ankommt, die mit der Anwendung arbeiten. Mit einer entsprechenden Norm, nach welcher festgelegt wird, welche Formate unterstützt werden müssen, kann diese Anforderung jedoch klar erkennen lassen, ob sie erfüllt ist oder nicht.

- 2.) „Die Karteikarten können im Rahmen einer Webanwendung im Internet mit Freunden ausgetauscht werden. Die online gestellten Karten dürfen keine Schimpfwörter und Kraftausdrücke enthalten.“

➔ Auch hier muss genau definiert werden, was unter „Kraftausdrücken“ und „Schimpfwörtern“ verstanden wird. In diesem Fall würde es sich anbieten, nach den Definitionen im Duden zu gehen und alle fraglichen Ausdrücke zu verbieten. Auch bei der Verwendung einer Blacklist kann hier nicht komplett ausgeschlossen werden, dass keine umgangssprachlichen Ausdrücke mehr auftreten.

- 3.) „Die Flashcards können im Rahmen einer Webanwendung online gelernt werden. Das Laden der nächsten zu lernenden Flashcard aus dem Internet und die Anzeige im Browser dürfen höchstens 50 Millisekunden dauern.“

➔ Anforderungen sind umzusetzende Bestandteile einer Realisierung. Da mit dem Laden aus dem Internet und der Wahl des Browsers Unsicherheiten in der Anforderungen enthalten sind, die gar nicht beeinflusst werden können. Man kann diese Anforderung so nicht mit aufnehmen, da lediglich Bedingungen aufgenommen werden sollten, die beeinflusst werden können.

- 4.) „ Die Registrierung als Benutzer für die Flashcards-Webanwendung soll einfach sein.“

➔ Dieses Fallbeispiel ist ein typischer Fall dafür, dass die Anforderung nicht ohne weiteres als erfüllt gelten kann. Die Definition von „einfach“ ist sehr stark abhängig von dem Kenntnisstand der Zielgruppe von den Personen, die später einmal mit dieser Applikation arbeiten werden. Außerdem sollte, falls eine einfache Gestaltung die Funktionalität einschränkt, auf einen Kompromiss beiderseits geachtet werden.

Aufgabe 3a)

- 1.) „Die Flashcard-Anwendung muss als Inhalt der Karteikarten alle gängigen Bildformate unterstützen.“

➔ Product requirements, Portability requirements

- 2.) „Die Karteikarten können im Rahmen einer Webanwendung im Internet mit Freunden ausgetauscht werden. Die online gestellten Karten dürfen keine Schimpfwörter und Kraftausdrücke enthalten.“

➔ External requirements, Ethical requirements

- 3.) „Die Flashcards können im Rahmen einer Webanwendung online gelernt werden. Das Laden der nächsten zu lernenden Flashcard aus dem Internet und die Anzeige im Browser dürfen höchstens 50 Millisekunden dauern.“

➔ Product requirements, Efficiency requirements, Performance requirements

- 4.) „ Die Registrierung als Benutzer für die Flashcards-Webanwendung soll einfach sein.“

➔ Product requirements, Usability requirements

Aufgabe 3b)

Product requirements:

Reliability requirements: Bsp: Der Server der Webanwendung soll 99.9% im Jahr verfügbar sein.

Organizational requirements:

Delivery requirements: Die Flashcard-Anwendung soll bis zum Datum x fertiggestellt sein.

Implementation requirements: Die zu verwendende Programmiersprache soll Java sein.

Standards requirements: Die Software soll ein HTML5 nach W3C-Standard erfüllen.

External requirements:

Interoperability requirements: Die Einbindung von Excel-Dateien soll gewährleistet sein.

Legislative requirements: Datenschutzlinien müssen eingehalten werden, z.B. im Zusammenhang mit Speicherung von Kenn- und Passwörtern.