

Softwaretechnik

Aufgabenblatt I

Daoud Ali 376352, Cihan Orhan 377061

Luca Gaudino 379780, Paul Sonnleitner

Aufgabe 1

Gegeben sei eine Software mit n Modulen. Wir modellieren die Anzahl der ausfallenden Softwaremodule mit einer binomialverteilten Zufallsvariablen X . Dabei fällt eine der Komponenten mit einer Wahrscheinlichkeit von $p = 0,01$ aus. Die Wahrscheinlichkeit, dass keine der Komponenten ausfällt ist gegeben durch:

$$P(X = 0) = \binom{n}{0} \cdot p^0 \cdot (1 - p)^n = 0,99^n$$

Damit die Software eher Einen, als keinen Fehler enthält muss die Software so viele Komponenten n enthalten, dass die Wahrscheinlichkeit $P(X = 0)$ unter $0,5$ sinkt.

$$P(X = 0) < 0,5 \Leftrightarrow 0,99^n < 0,5 \Leftrightarrow n > \log_{0,99}(0,5) \Leftrightarrow n > 68,96...$$

Somit muss das System mindestens 69 Komponenten enthalten, damit Wahrscheinlichkeit eines oder mehrere Fehler größer als 50% ist.

Aufgabe 2

- Benutzer der Software schulen: Test-Integration
Nach der Fertigstellung der Software.
- Qualitätssicherung des Pflichtenheftes durchführen: Analyse
Nach der Erstellung wird das Pflichtenheft nochmal überprüft/verifiziert.
- Gesetzliche Rahmenbedingungen prüfen: Analyse
Am Anfang zu überprüfen um das Projekt danach auszurichten.

- Konzept und Prototyp einer Benutzeroberfläche erstellen: Analyse, Entwurf
Um die Anforderungen des Kunden besser zu verstehen und als Entwurf für das zu bauende Produkt.
- Entwicklerteam zusammenstellen: Analyse, Entwurf, Implementierung
Das Team wird von Anfang an zusammengestellt. Dabei können aber Programmierer auch erst in der Implementierungsphase hinzukommen.
- Code eines Programmmoduls debuggen: Implementierung, Test
Um sicherzugehen, dass das Modul korrekt funktioniert. Falls Fehler beim testen gefunden werden.
- Zwei Subsysteme verbinden und testen: Test-Integration
Nach dem die Subsysteme implementiert und einzeln getestet sind, sich werden sie verbunden.
- Termine und Kosten des Projektes planen: Entwurf
Nach der Analyse wird das Projekt durchgeplant.
- Datenstrukturen festlegen: Entwurf
Die Datenstrukturen werden vor der Implementierung bereits festgelegt.
- Vorhandene Altlasten des Kunden analysieren: Analyse
Vorherige Software analysieren und in das neue Projekt mit einbeziehen.
- Schnittstellen von Programmmodulen definieren: Entwurf
Um Interaktion und aufbauende Funktionalitäten zu implementieren.
- Leistung der Entwickler bewerten und belohnen: Implementierung, Test-Integration
Bei Implementierung und Test zeigt sich die Effizienz der einzelnen Entwickler.
- Software an neue Umgebung anpassen: Wartung
Falls sich nach der Integration etwas an der Umgebung ändert muss die Software angepasst werden. Mögliche Fehler oder Performanceverluste.
- Kunden eine Rechnung stellen: keine Zuordnung
Nach dem Entwurf erste Prognose. Nach der Integration, Rechnung für das Produkt. Dann Wartungsvertrag.
- Test-Eingabedaten für ein Programmmodul ermitteln: Implementierung
Vor der Implementierung z.B. XP oder vor dem Testen.
- Strukturmodell des gesamten Softwaresystems entwerfen: Entwurf
Struktur muss für eine geordnete Implementierung feststehen.

- Dokumentation des Projektablaufes bewerten und archivieren: keine Zuordnung
Immer sinnvoll, um aus Fehlern zu lernen. Für das aktuelle oder folgende Projekte.
- Nach bereits vorhandenen, wiederverwendbaren Software-Bibliotheken suchen: Entwurf
Um Kosten und Zeit durch Wiederverwendung einzusparen.
- Performance-Prognose des Softwaresystems erstellen: Entwurf
Um den Entwurf performance-technisch anzupassen.
- Programmcode kommentieren: Entwurf, Implementierung
Sinnvoll vor und während der Implementierung, um TODO's festzuhalten und den Programmcode nachzuvollziehen.

Aufgabe 3

SCRUM

- Transparent → Fortschritt und Probleme werden täglich festgehalten
- feste Zeitintervalle

XP

- Sehr effizient
- Nicht für Großprojekte geeignet

Aufgabe 4

Funktionale Anforderungen

- Mechanismus zur Synchronisation von Veranstaltungen, Lernräumen etc. mit dem vorhandenen Management System
- Erstellung und Verwaltung eines Benutzerprofils
- Funktion zur Ermittlung potentieller Übungspartner mit Zugriff auf den Standort der Geräte
- Funktion zum Versenden von Nachrichten und Anfragen

Nicht-funktionale Anforderungen

- Automatische Synchronisation täglich, manuelle Synchronisation möglich
- Verwendung der Daten gemäß Datenschutzrichtlinien
- Auf den Standort darf nur unter Einverständnis des Benutzers zugegriffen werden.
- Senden und Empfangen einer Nachricht darf nicht länger als 10s dauern.
- Die App muss mit Android und iOS kompatibel sein.