



Vorlesung Software Engineering, WS 2013

Übungsaufgabe Projektverwaltung

Sie betreiben mit einigen Kommilitonen eine kleine, schon recht erfolgreiche Softwarefirma. Vor kurzem ist eine Ihnen gelungen, einen lukrativen Auftrag der Firma Continental AG zu gewinnen. Die Continental AG möchte eine neue Software zur Verwaltung und Dokumentation von Projekten entwickeln und hat Sie mit der Entwicklung eines ersten Prototyps betraut. Die Software soll mit einem kompletten Engineering Zyklus entwickelt werden.

Im Folgenden wird die Aufgabenstellung näher beschrieben:

Allgemeines:

Die Projektverwaltung soll es dem Projektleiter ermöglichen, Planungs- und Abrechnungsdaten für (Software-) Projekte zu verwalten. D.h es können mehrere gleichzeitig Projekte verwaltet werden, Work Packages angelegt werden, Projektmitarbeiter zu Work Packages zugeordnet werden.

In den Work Packages soll ein Titel mit Aufgabenbeschreibung, Aufwandsabschätzung sowie Ist-Aufwand eingetragen werden können. Im Weiteren sollen SW-Releases verwaltet werden, sowie die Zuordnung der Work Packages zu den Releases.

Anforderungen im Detail:

1. Es soll möglich sein, Projekte neu anzulegen.
2. Jedes Projekt muss durch einen eindeutigen Namen gekennzeichnet sein.
3. Es soll möglich sein, Projektinformationen zuzuweisen und zu ändern.
4. Es soll möglich sein, Projektmitarbeiter neu anzulegen und Projekten zuzuordnen. Dabei kann ein Mitarbeiter in mehreren Projekten arbeiten
5. Es soll möglich sein, Arbeitspakete zu definieren und diese den Projekten zuzuordnen
6. Es soll möglich sein, (Software-) Freigaben zu definieren und diese den Projekten zuzuordnen
7. Es soll möglich sein, Arbeitspaketes zu Freigaben zuzuordnen
8. Folgende Daten eines Projektes sollen gespeichert werden:
 - Projektnummer
 - Titel
 - Kurzbeschreibung
 - Budget
9. Folgende Daten eines Mitarbeiters sollen gespeichert werden:
 - Mitarbeiterpersonalnummer
 - Vor- und Nachname
 - Abteilung
 - Stundensatz
10. Folgende Daten eines Arbeitspaketes sollen gespeichert werden:
 - Arbeitspaket-Nummer



- Titel
 - Kurzbeschreibung
 - Änderung oder Fehler
 - Plan-Aufwand
 - Ist-Aufwand
 - Zuordnung zum Projekt
 - Zuordnung zum Release
11. Folgende Daten einer (Software-) Freigabe sollen gespeichert werden:
- Freigabe-Nummer
 - Projektbezeichnung
 - Kurzbeschreibung (Prototyp, Alpha-, Beta-, Produktversion)
 - Versions-(bzw. Build-) Nummer
 - Enthaltene Arbeitspakete (unterteilt nach Änderungen und Fehlern)
 - Gesamt-Planaufwand
 - Gesamt-Ist-Aufwand
 - Zuordnung zum Projekt
12. Der Zugriff auf Projektdaten soll komfortabel möglich sein
13. Es sollen ausreichend viele Projekte, Arbeitspakete, etc. verwaltet werden können
14. Alle Daten sollten komfortabel eingebbar, editierbar, und ausdrückbar sein



Aufgabenschritte:

1. Organisieren Sie Ihr Team, verteilen Sie Aufgaben, achten Sie darauf, dass alle Teammitglieder bei allen Aufgabenschritten beteiligt sind. Wechseln Sie die Aufgaben an verschiedenen Übungsterminen.
2. Erstellen Sie einen Projektantrag unter Verwendung des vorgegebenen Formulars. Lassen Sie den Projektantrag beim Aufgabensteller genehmigen.
3. Planen Sie Ihr Projekt ausgehend von den Aufgabenstellung
 - a. zeitliche Planung von Arbeitspaketen und deren Abhängigkeit
 - b. Ressourcenplanung (wer macht was)
 - c. Konfigurationsmanagement (wo wird was abgelegt, wie ist der Zugriff geregelt)

Verwenden Sie für die Planung das Tool Open Project (oder ggf. auch Excel)

4. Requirements
 - a. Analysieren Sie die Requirements auf
 - Verständlichkeit
 - Vollständigkeit
 - Umsetzbarkeit
 - b. Klären und vervollständigen Sie die Requirements mit dem Aufgabensteller (wo notwendig)

Dokumentieren Sie die Requirements in einem geeigneten Word Dokument (ggf. unter Verwendung des vorhandenen Templates). Verwenden Sie die Requirements aus der Aufgabenstellung als Ausgangspunkt. Achten Sie darauf, dass die Requirements eindeutig identifizierbar und auch testbar sind.

5. Führen Sie unter Verwendung der UML eine objektorientierte Analyse der Aufgabenstellung durch
 - a. Stellen Sie die wichtigsten Use-Cases auf.
 - b. Erstellen Sie die Programmstruktur aus Analysesicht.

Dokumentieren Sie Ihre Ergebnisse mittels eines geeigneten Tools (z.B. Together oder Visio)

6. Erstellen Sie unter Verwendung der UML das Design (SW-Architektur und Feindesign) des Programms (Feindesign ggf. nur von Teilen des Programms). Berücksichtigen Sie dabei statische und dynamische Elemente.
 - a. Identifizieren und beschreiben Sie die Komponenten des Programms inkl. ihrer Schnittstellen und Beziehungen zueinander
 - b. Identifizieren und beschreiben Sie die Klassen des Programms inkl. ihrer Schnittstellen und Beziehungen zueinander
 - c. Identifizieren Sie Programmzustände und beschreiben Sie diese (wo notwendig) mittels Zustandsautomaten
 - d. Identifizieren Sie wichtige Abläufe und beschreiben Sie diese mittels Sequenzdiagrammen

7. Implementieren Sie die Klassen in einer objektorientierten Programmiersprache
8. Führen Sie ein Codereview einiger Ihrer Klassen in Ihrem Team durch. Dokumentieren Sie das Ergebnis.
9. Führen Sie einen Modultest ihrer Klassen durch. Erstellen Sie dazu zunächst eine Testspezifikation. Implementieren Sie die Testfälle und führen Sie die Tests aus.
10. Führen Sie eine Validierung Ihres Programms durch



- a. Erstellen Sie eine Testspezifikation (Word Dokument) für die Validierung Ihres Programms.
 - unter Verwendung geeigneter Teststrategien
 - zum Test aller Requirements
- b. Stellen Sie sicher, dass sich die Testfälle den zu testenden Requirements zuordnen lassen.
- c. Implementieren Sie die Testfälle und führen Sie den Test durch. Dokumentieren Sie die Testergebnisse.