



Vorlesung Software Engineering, WS 2013

Übungsaufgabe Patientenverwaltung

Sie betreiben mit einigen Kommilitonen eine kleine, schon recht erfolgreiche Softwarefirma. Vor kurzem ist eine Ihnen gelungen, einen lukrativen Auftrag der Kassenärztlichen Vereinigung zu gewinnen. Die Kassenärztliche Vereinigung möchte eine neue Software zur Verwaltung von Patientendaten und Rechnungen 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:

Lastenheft

Allgemeines:

Die Patientenverwaltung soll es einer Arztpraxis ermöglichen, Daten von Patienten zu speichern, zu ändern und zu löschen. Zusätzlich soll es möglich sein, für alle Patienten die erbrachten Leistungen zu erfassen und entsprechende Rechnungen oder ggf. Mahnungen zu erstellen, auszudrucken und abzuspeichern. Dabei ist zwischen Privatpatienten und Kassenpatienten zu unterscheiden. Leistungen werden je nach Kassenzugehörigkeit unterschiedlich abgerechnet.

Anforderungen im Detail:

1. Es soll möglich sein, Patientendaten neu anzulegen.
2. Es soll möglich sein, Patientendaten zu ändern.
3. Es soll möglich sein, Patientendaten zu löschen.
4. Folgende Daten eines Patienten sollen gespeichert werden:
 - Vor- und Nachname
 - Geschlecht
 - Geburtsdatum
 - Adresse
 - Art der Krankenversicherung (privat / pflichtversichert)
5. Der Zugriff auf Patientendaten soll komfortabel möglich sein
6. Es sollen ausreichend viele Patienten verwaltet werden können
7. Es soll möglich sein, Leistungen neu anzulegen.
8. Es soll möglich sein, Leistungen zu ändern.
9. Es soll möglich sein, Leistungen zu löschen.
10. Folgende Daten einer Leistung sollen gespeichert werden:
 - Nummer der Leistung
 - Name der Leistung (Kurztext)
 - Wert der Leistung für Kassenpatienten in €
 - Wert der Leistung für Privatpatienten in €
11. Der Zugriff auf Leistungsdaten soll komfortabel möglich sein
12. Es sollen ausreichend viele Leistungen verwaltet werden können



13. Es soll möglich sein, Rechnungen für einen Patienten zu erstellen mit folgenden Informationen
 - Anschrift
 - Absender
 - erbrachte Leistungen mit dem jeweiligen Datum
 - Gesamtbetrag
 - Bankverbindung der Arztpraxis
14. Es soll möglich sein, erstellten Rechnungen auszudrucken
15. Es soll möglich sein, erstellten Rechnungen abzuspeichern
16. Es soll möglich sein, gespeicherte Rechnungen zu ändern
17. Es soll möglich sein, gespeicherte Rechnungen zu löschen
18. Bezahlte Rechnungen sollen als solche gekennzeichnet werden
19. Es soll möglich sein, zu einer bereits erstellten Rechnung eine Mahnung zu erstellen
20. Die Erstellung und Verwaltung von Rechnungen und Mahnungen soll komfortabel möglich sein.
21. Es sollen ausreichend viele Rechnungen verwaltet werden können



Aufgabenschritte:

1. Erstellen Sie einen Projektantrag unter Verwendung des vorgegebenen Formulars. Lassen Sie den Projektantrag beim Aufgabensteller genehmigen.
2. Planen Sie Ihr Projekt ausgehend von den Aufgabenstellung
 - a. zeitliche Planung
 - b. Ressourcenplanung (wer macht was)
 - c. Konfigurationsmanagement (wo wird was abgelegt, wie ist der Zugriff geregelt)
3. Requirements
 - a. Analysieren Sie die Requirements auf
 - Verständlichkeit
 - Vollständigkeit
 - Umsetzbarkeit
 - b. Klären und vervollständigen Sie die Requirements mit dem Aufgabensteller (wo notwendig)
4. 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
5. Erstellen Sie unter Verwendung der UML das Design (SW-Architektur und Feindesign) des Programms
 - 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 Programzzustände und beschreiben Sie diese (wo notwendig)
 - d. Identifizieren Sie wichtige Abläufe und beschreiben Sie diese
6. Implementieren Sie die Klassen in einer objektorientierten Programmiersprache
7. Führen Sie ein Codereview einiger Ihrer Klassen in Ihrem Team durch. Dokumentieren Sie das Ergebnis.
8. Führen Sie einen Modultest ihrer Klassen durch. Erstellen Sie dazu zunächst eine Testspezifikation. Implementieren Sie die Testfälle und führen Sie die Tests aus.
9. Führen Sie eine Validierung Ihres Programms durch
 - a. Erstellen Sie eine Testspezifikation für die Validierung Ihres Programms.
 - unter Verwendung geeigneter Teststrategien
 - zum Test aller Requirements
 - b. Implementieren Sie die Testfälle und führen Sie den Test durch. Dokumentieren Sie die Testergebnisse.