Informationen zum Softwaretechnik-Projekt WS 2012/13

Ziel:

Das Software-Projekt soll Ihnen erste Erfahrungen hinsichtlich des Entwurfs, der Implementierung und der Qualitätssicherung von Software im Großen geben. Dies umfasst die Phasen Analyse bis Test im unten abgebildeten Top-Down-Entwicklungsprozess nach einem Phasenmodell. Es sollen die in den Vorlesungen besprochenen Techniken zur Anwendung kommen.

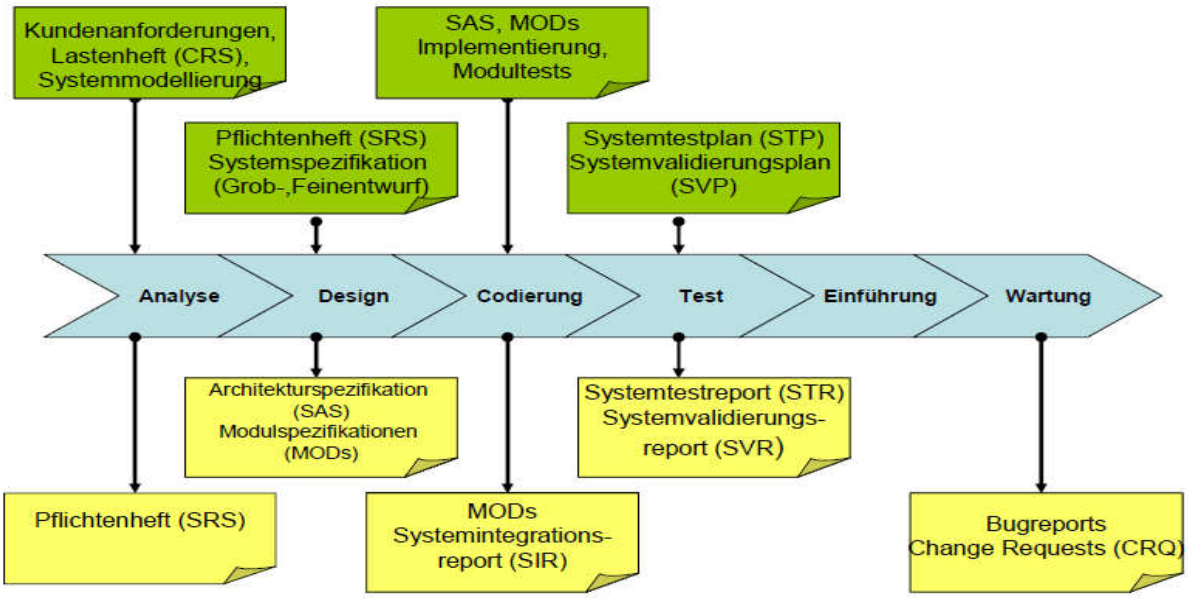
Rahmenbedingungen:

- Gruppen zu 5-6 Studenten.
- Jeder im Team hat eine bestimmte Rolle und die zugehörigen Aufgaben.
- Jeder im Team muss zusätzlich für mindestens ein Programmmodul die Verantwortung inkl. Implementierung übernehmen.
- Die Teamsitzungen müssen durch Protokolle dokumentiert werden, die Beschlüsse, Status und Ergebnisse darstellen.

Bewertung:

Die Note ergibt sich aus folgender prozentualer Verteilung der Bewertung der Gruppenarbeit:

- 30% kollektive Leistung für Entwicklungsdokumentation inklusive Präsentationen
- 40% individuelle Leistung des Einzelnen in seiner Rolle und für sein verantwortliches Modul
- 30% kollektive Leistung der Implementierung (inkl. Präsentation, Software und Dokumentation)
- Das jeweils beste Teamergebnis soll als nichtkommerzielles Open-Source-Softwareprojekt veröffentlicht werden. Da Sie das Urheberrecht an dem von Ihnen erstellten Code haben, müssen Sie sich hierzu vorab einverstanden erklären!



Aufgabe im 3. Semester:

1. Analysieren Sie das (vom Kunden) bereitgestellte Lastenheft (CRS). Ergänzen Sie die zusätzlichen Kundenanforderungen im CRS, die sich in der Anforderungserhebungsphase ergeben.
2. Erstellen Sie einen einfachen (!) Business Case (BC), indem Sie von einer fiktiven kommerziellen Auftragsentwicklung ausgehen.
3. Modellieren und definieren sie das System aus Black-Box-Sicht (idealerweise mithilfe einer prototypischen Implementierung) und erstellen Sie das Pflichtenheft (SRS).
4. Jedes Team präsentiert die Ergebnisse in einer 20minütigen Präsentation mit anschließender 10minütiger Verteidigung.

Achten Sie auf die in der Vorlesung behandelten Punkte zum Thema Requirements Engineering und Systemanalyse. Versuchen Sie in Ihrer Gruppe durch Reviews zu gut spezifizierten Anforderungen und den geeigneten Modellierungstechniken zu kommen.

5. Modellieren Sie das System aus White-Box-Sicht und erstellen Sie die Architekturspezifikation (SAS). Leiten Sie die zu implementierenden Module daraus ab und weisen Sie diese als Arbeitspakete den Teammitgliedern zu.
6. Erstellen Sie den detaillierten Projektplan. Versuchen Sie dabei die Aufgaben zu parallelisieren und beachten Sie Abhängigkeiten zwischen Arbeitspaketen.

Abzugeben sind zwei Tage vor dem Präsentationstermin folgende Dokumente in elektronischer Form (PDF und Basisformat):

- CRS, BC, SRS, SAS, Projektplan

Aufgabe bis Ende des 4. Semesters:

1. Implementieren und Testen Sie das System.
2. Erstellen bzw. finalisieren Sie die zugehörige Dokumentation wie SRS, MODs, STP und STR.
3. Jedes Team präsentiert seine Ergebnisse in einer 20minütigen Präsentation mit anschließender 10minütiger Verteidigung.

Abzugeben sind zwei Tage vor dem Präsentationstermin sämtliche Projektdokumente in elektronischer Form (PDF und Basisformat):
CRS, BC, SRS, SAS, Projektplan, MODs, STP,STR, Manual, Projektpräsentationen.

Projektthemen

1. **Projekt: „Vorlesungsunterlagen SWE I“** 4
2. **Projekt: „Salome-TMF 3.x: Usability-Verbesserung“** 5
3. **Projekt: „Salome 4: Neuentwicklung SDLC-Tool“** 6
4. **Projekt: „MultiCastor 3.0“** 7
5. **Projekt: „Generic Assessment Tool 2.0“** 8

Projekt 1 - "Vorlesungsunterlagen SWE I"

Benjamin Salchow (Projektleiter)
Rene Kienzle
Marius Erlen

~~Projekt 2 - "Salome-TMF 3.x: Usability-Verbesserung"~~ **(aufgelöst)**

~~Sebastian Simon (Projektleiter)~~
~~Petra Wolf,~~
~~René Kienzle,~~
~~Benjamin Salchow~~

Projekt 3 - "Salome 4: Neuentwicklung SDLC-Tool"

Giorgio Pirri (Projektleiter)
Ephraim Petry
Christian Hilb (Qualitätssicherung)
Christian Kahlau
Stefan Blum
Bereket Bahta

Projekt 4 – „MultiCastor 3.0“

Nick Herrmannsdörfer (Projektleiter)
Stefan Heßler (Lead Engineer)
Patrick Robinson
Erwin Stamm
Kai Brennenstuhl

Projekt 5 – „Generic Assessment Tool 2.0“

Florian Harr (Projektleiter)
Jan Forster (Lead Engineer)
Lucas Bader
Matthias Harrer
David Randler

1. Projekt: „Vorlesungsunterlagen SWE I“

Die Vorlesung Software Engineering I der Dozenten Rentschler & Stuckert wurde im Punkt Präsentations- und Vorlesungsunterlagen in der Vergangenheit leider nicht immer mit „gut“ oder „sehr gut“ bewertet. Regelmäßige Hauptkritikpunkte sind:

1. Präsentation/Skript wird nicht in elektronische Form den Studenten zur Verfügung gestellt (Anmerkung: Das hat seinen Hauptgrund darin, dass zum Teil Bilder ohne Quellenangabe verwendet werden).
2. Präsentation/Skript ist inhaltlich nicht optimal gestaltet, z.B. zu wenig / zu viel oder redundante Information.
3. Es werden keine oder zuwenig Übungsaufgaben bereitgestellt.

Da die Studenten als Zielgruppe selbst am Besten die Qualität der Skripte und der Vorlesungsinhalte bewerten kann, sollte hier effektiv eine signifikante Verbesserung erarbeitet werden können.

Deshalb soll das Präsentationsskript der Vorlesung durch ein studentisches Projektteam nach zu definierenden didaktischen Konzepten überarbeitet und verbessert werden. Dies soll vorlesungsbegleitend erfolgen, d.h. während der Vorlesung wird beobachtet und simultan notiert, was verbessert werden kann und dies dann zeitnah umgesetzt. Diese Optimierungsmaßnahmen sollen in enger Zusammenarbeit mit den Dozenten erfolgen.

Im Rahmen des Projekts sollen folgende Artefakte erstellt werden:

1) Eine überarbeitete PPT-Präsentation und Prozess-Templates, die veröffentlichungssicher sind.
2) Ein veröffentlichungssicheres Manuskript/Handout (in MS-Word und ODT Format) als Fachliteraturergänzung. Hier kann auf einem bestehenden Dokument aufgesetzt werden.
3) Ein überarbeitetes EXCEL-Sheet zur Notenberechnung mit statistischen Auswertungen (Zahlen + Grafiken) für die Berechnung der ECTS-Klassifikation.
4) Eine webbasierte eLearning-Software mit Multiple-Choice-Lernerfolgskontrolle. Hierfür sollte als Basis das Generic Assessment Tool 2.0 (Projekt 5) verwendet werden.

Die erzielten Resultate sollen anschließend an der DHBW Stuttgart in die Open-Source gestellt werden, d.h. auch andere DHBW-Dozenten sollen es benützen können.

Bei dieser Aufgabe stehen Grundlagen des Software-Engineering, didaktische Konzepte, Präsentationstechnik, Texterstellung und -redaktion im Vordergrund. Das Projekt besitzt Programmierungsanteile im Bereich der eLearning-Software und wird wie jedes andere SWE-Projekt nach Vorgehensmodell abgearbeitet mit Erstellung der projektbegleitenden Dokumentation inkl. UML-Modellierung.

ECTS-Klassifikation: (*http://www.klaus-gach.de/noten.htm*)

- A für die besten 10 Prozent,
- B für die nächsten 25 Prozent,
- C für die nächsten 30 Prozent,
- D für die nächsten 25 Prozent,
- E für die nächsten 10 Prozent.

2. Projekt: „Salome-TMF 3.x: Usability-Verbesserung“

Die Testmanagement-Lösung „***Salome-TMF***“ (https://wiki.ow2.org/salome-tmf/) bietet umfassende Werkzeugunterstützung für den Software-Qualitätssicherungsprozess, vom Requirements-Management über Testfallmanagement, Testprojektmanagement und Bugtracking bis zum Reporting und hat eine breite Benutzerbasis.

http://www.java2s.com/Open-Source/Java-Document/Test-Coverage/salome-tmf/Catalogsalome-tmf.htm

Salome-TMF ist in Java programmiert und wurde mit einer Architektur entworfen, die Schnittstellen für sog. „Plugins“ bereitstellt. Dadurch ist das Tool flexibel erweiter- und konfigurierbar.
Die Software soll überarbeitet werden. Hierzu sollte der französische Maintainer des Open-Source-Projekts kontaktiert werden, der den allerneuesten Bugfix-Code zur Verfügung stellen kann, auf dessen Basis dann gearbeitet werden soll.

Es müssen folgende Punkte bearbeitet werden:

1) Lauffähige Installation von Salome-TMF 3.x auf einem DHBW-Server. Entwicklung eines Installers.
2) Usability-Analyse und ggf. Verbesserung der Screens, Dialoge und Icons, insbesondere der Installationsroutine, des Login-Screens, des Projektverwaltungs-Moduls und der Artefakte-Bäume.
3) Sprachunterstützung des Tools auf Deutsch erweitern und korrekte Sprachunterstützung (Deutsch, Englisch, Französisch) prüfen und ggf. verbessern. Rudimentäre Manualdateien in den verschiedenen Sprachen erstellen.
4) Weiterentwicklung des generischen Hilfe-Konzepts, so dass über den jeweiligen Modul/Plugin-Screen die zugehörige Manual-Datei aufgerufen werden kann.
5) Externe oder angehängte Objekt- und Grafik-Dateien sollen im Text eingebettet bzw. verlinkt werden können und bei Doppelklick bzw. Rechtsklick und Auswahlmenü auf das Objekt bzw. den Link die zugehörige Anzeige- oder Bearbeitungsapplikation gestartet werden können.
6) Im Beschreibungstext integrierte Verlinkung zwischen Artefakten (Requirements bzw. Testfälle) sollte möglich sein und durch eine Art Browser-Funktion mit Vor- und Zurück-Navigation unterstützt werden.
7) Die integrierte Verlinkung sollte auf Konsistenz geprüft werden können, wenn ein neuer Link eingefügt oder ein bestehendes -evtl. verlinktes- Element umbenannt oder gelöscht wird.
8) Die geänderte Software muss am Ende sorgfältig getestet werden.

Das Projekt ist entwicklungsbegleitend weitgehend in Salome-TMF selbst zu dokumentieren.
Die erzielten Ergebnisse sollen anschliessend in das zugehörige Open-Source-Projekt einfliessen.

Bei dieser Aufgabe steht der Entwurf und Umsetzung von Methoden zum Requirements Engineering im Vordergrund. Ein geeigneter Workflow mit benutzerfreundlicher GUI, Datenmodellierung und integrierte Dokumentengenerierung sind die Hauptaspekte.

3. Projekt: „Salome 4: Neuentwicklung SDLC-Tool“

Die Testmanagement-Lösung „***Salome-TMF***“ (https://wiki.ow2.org/salome-tmf/) bietet umfassende Werkzeugunterstützung für den Software-Qualitätssicherungsprozess, vom Requirements-Management über Testfallmanagement, Testprojektmanagement und Bugtracking bis zum Reporting.

Salome-TMF 3.x ist in Java programmiert und technologisch nicht mehr ganz zeitgemäß. Ausserdem ist es vom Benutzerkonzept mehr am Testplan als am Software-Lebenszyklus orientiert. Es soll daher in einer Version 4 in einer Web 2.0 Architektur neuentwickelt werden. Dabei ist es nicht nötig, das Rad von Grund auf neu zu erfinden sondern es soll auf dem offenen Dokumentenmanagementsystem OpenKM aufgesetzt werden, dessen Framework bereits viele wichtige Grundfunktionalitäten zur Verfügung stellt. Es müssen mit Salome 3.x erstellte Datenbanken weiterverwendet werden können. Folgende Teilaufgaben sind zu bearbeiten

1) Benutzbarkeitsanalyse Salome-TMF 3.x durchführen und auswerten.
2) Definition eines modularen, SDLC-orientierten Bedienungskonzepts.
3) Workflow- und Screen-Design für Salome 4, betreffend Projektmanagement, Requirements Management, Testfall-Management. Es muss ein Traceability-Konzept zwischen Requirements – Sourcecode – Testfall pro Projekt und zwischen Projekten möglich sein.
4) Analyse des OpenKM-Frameworks, entfernen nicht benötigter Teile, Weiterverwendung insbesondere der LDAP-fähigen Benutzerverwaltung.
5) Definition einer zukunftsfähigen Software-Architektur auf Basis des OpenKM-Frameworks.
6) Implementierung des SDLC-Managementkonzepts auf der Basis des Open-KM-Frameworks, insbesondere Fertigimplementierung des Requirements-Management-Moduls.
7) Erstellung einer Hilfefunktion mit Sprachunterstützung (Deutsch, Englisch, Französisch) und vom Benutzer erweiterbaren Hilfetextdateien (XML).

Das Projekt ist entwicklungsbegleitend weitgehend in Salome-TMF 3.x selbst zu dokumentieren. Aufgrund der Verwendung des bestehenden OpenKM-Frameworks ist eine agile Vorgehensweise mit am Anfang durchgeführten Prototyping sinnvoll.
Die erzielten Ergebnisse sollen anschliessend in ein Open-Source-Projekt einfliessen, dass in den nächsten Jahren weiterentwickelt werden soll.

Bei dieser Aufgabe steht der Entwurf und Umsetzung von Methoden zum Software-Lebenszyklus vom Requirements Managementbis zum Testfall-Management im Vordergrund. Ein geeigneter Workflow mit benutzerfreundlicher GUI, Datenmodellierung und integrierte Dokumentengenerierung sind die Hauptaspekte dieser Aufgabe.

4. Projekt: „MultiCastor 3.0“

*Die bestehende Open-Source-Software „MultiCastor 2.0“ ist ein Tool zum Testen der Multicast-Funktionalitäten von Netzwerkgeräten. Es soll mit neuen Features und Verbesserungen der Usability zu „**MultiCastor 3.0**“ weiterentwickelt werden. Ziel ist es, das beste freie Multicast-Testtool weltweit zu realisieren.*

Insbesondere ist die Erweiterung des Tools um das Protokoll GMRP nach IEEE 802.1D und SRP nach IEEE802.1Qat geplant. Die Multicastregistrierung in den Netzwerk-Switchen erfolgt dann entweder über IGMP(IPv4), MLD(IPv6), MMRP oder GMRP (Layer 2).
Für die Anwendung des Tools zu Testautomatisierungszecken soll ausserdem die Integration in das Automatisierungs-Framework STAF/STAX nachgewiesen werden.
Darüberhinaus soll ein (teil)automatisiertes Regressionstestkonzept auf Black-Box-Ebene für die Software selbst entworfen werden.
Die erzielten Ergebnisse sollen anschliessend in das zugehörige Open-Source-Projekt einfliessen.

Siehe auch: http://studium.dhbw-stuttgart.de/informatik/projekte.html

Diese Aufgabe ist gut geeignet für technisch orientierte Leute mit Interesse an Ethernet- und TCP/IP-Netzwerken sowie Protokoll-Zustandsautomaten. Sehr interessant sind auch die Testautomatisierungsaspekte.
GUI-Entwicklung spielt insoweit eine Rolle, da die bestehende GUI an die neuen Funktionen angepasst werden muss und einige Usability-Verbesserungen vorgenommen werden sollten.

5. Projekt: „Generic Assessment Tool 2.0“

Tests, Bewertungen, Audits, Assessments usw. werden zu vielen unterschiedlichen Themen durchgeführt.
Im Kern werden dabei immer Fragen zu bestimmten Themengebieten gestellt und anhand der Antworten der Reifegrad des Untersuchungsobjekts festgestellt. Der Fragenkatalog modelliert dabei das zu bewertende System bzw. Untersuchungsobjekt. Die Modellkomplexität und Anzahl der Modelldimensionen kann dabei variieren, so sind mehrere Verknüpfungstiefen einzelner Themengebiete möglich.

Die angewendete Bewertungsskala orientiert sich am jeweiligen Themengebiet. Im einfachen eindimensionalen Fall von Schulnoten bspw. von 1 bis 6, bei IQ-Tests von 0-180 Punkten, bei Prozessbewertungen bspw. an Reifegradstufen von 1 bis 5 etc.

Nützlich wäre ein Software-Werkzeug, dass in der Lage ist, derartige Szenarien abzubilden. Hierzu müssen die zugehörigen Fragekataloge strukturiert verwaltet werden können und die erzielten Ergebnisse auch bei größeren Komplexitätsgraden und mehreren Dimensionen anschaulich dargestellt werden. Neben numerischen Metriken ist in solchen Fällen insbesondere eine zwei- oder dreidimensionale grafische Darstellung zielführend.

Deshalb soll ein im letzten Studienjahr begonnene Toolentwicklung (https://sourceforge.net/projects/assessmenttool/files/) fortgeführt und daraus ein verbessertes Software-Werkzeug namens „**Generic Assessment Tool 2.0**“ erstellt und anschliessend zum Wohle der Menscheit als Open-Source-Software freigegeben werden.

Bei dieser Aufgabe steht der Entwurf eines Workflows, einer benutzerfreundlichen GUI und, Datenmodellierung mit XML-Strukturen sowie (dreidimensionale) grafische Ergebnis-Darstellung im Vordergrund.