Richtlinie zur Gestaltung des Portfolios INF-B-320 (Softwaretechnologie-Projekt)

Version 0.14, 01.09.2020

Prof. Uwe Aßmann, Dr. Birgit Demuth, Softwaretechnologie, Fakultät Informatik

Vorbemerkung

Dieser Leitfaden nimmt einige Begriffe der Rahmenprüfungsordnung der TU Dresden auf, die im Mai 2020 vom Senat verabschiedet wurde. Da die zukünftigen Ordnungen der Studiengänge der Informatik diese Rahmenprüfungsordnung verwenden werden, werden auch in dieser Richtlinie die Begriffe der Rahmenprüfungsordnung bereits verwendet.

1. Merkmale

1.1. Qualifikationsziele

Die Studierenden besitzen praktische ingenieursmäßige Kenntnisse in der Durchführung von teamorientierten, arbeitsteiligen Softwareprojekten. Die Studierenden sind in der Lage, in Zusammenarbeit mit einem Kunden dessen Anforderungen zu analysieren sowie arbeitsteilig ein Softwaresystem zu entwerfen, zu implementieren, zu testen und vom Kunden abnehmen zu lassen.

1.2. Inhalte

Es wird ein teamorientierter Softwareentwicklungsprozess durchgeführt, der Kundenanforderungen aufnimmt und abarbeitet. Dazu gehören für die für den Kunden zu realisierende Anwendung die Erstellung einer Anforderungsspezifikation, eines Softwareentwurfs, kleiner Prototypen zur Einarbeitung in die zu verwendenden Frameworks bzw. Technologien, einer Implementierung und einer Dokumentation. Weitere Aufgaben stellen die Qualitätssicherung wie die Erstellung einer Testsuite und die Auswertung von Softwareanalysen dar. Daneben werden Tätigkeiten des Projektmanagements geübt, wie Teamtreffen und deren Protokollierung, Kundengespräche, Arbeitsstundenerfassung, Reflektion und Controlling des Projektstandes an wohldefinierten Meilensteinen, sowie eine Abschlusspräsentation vor dem Kunden.

1.3. Lehr- und Lernformen

Das Modul besteht aus einem Portfolio als Resultat einer teamorientierten Projektarbeit und einem Referat, unterstützt durch Tutorien und Selbststudium.

1.4. Voraussetzungen zur Teilnahme

Es werden die Kenntnisse vorausgesetzt, die im Modul “Softwaretechnologie“ (INF-B-310 in den Bachelor-Studiengängen Informatik und Medieninformatik bzw. INF-D-240 im Diplomstudiengang Informatik) erworben werden. Darunter zählen vor allem Methoden zur Entwicklung großer Softwaresysteme, Objektorientierung, die Verwendung einer Modellierungssprache wie die Unified Modeling Language (UML) in Analyse, Entwurf und Implementierung sowie die Programmierung in einer objektorientierten Programmiersprache wie Java.

1.5. Zur parallelen bzw. sequentiellen Studierbarkeit des Modules INF-B-320

Die Definition von Modulabhängigkeiten in einem Studiengang ist dann geboten, wenn zum Studieren des abhängigen Moduls das vorangehende Modul nötig ist. In der Regel weist man Kenntnisse durch eine bestandene Klausur im Modul aus, das die Abhängigkeit erzeugt. Daher sollte eine Modulabhängigkeit dann definiert werden, wenn man ein Modul nur dann studieren kann, wenn man das vorangehende erfolgreich absolviert hat.

In der Modulbeschreibung von INF-B-320 ist das Attribut “Voraussetzungen zur Teilnahme” beschrieben durch diejenigen Kenntnisse, die durch die erfolgreiche Teilnahme Module INF-B-310 oder INF-D-240 erzielt werden können. In der Technischen Universität Dresden legt man das Attribut “Voraussetzungen zur Teilnahme” als “Kann”-Bestimmung aus, d.h. die beschriebenen Kenntnisse sind zum Beginn von INF-B-320 formal nicht notwendig.

Fachlich ist diese Abhängigkeit dennoch geboten; INF-B-310/INF-D-240 und INF-B-320 können nicht parallel studiert werden, sondern stehen in Abhängigkeit und müssen sequentiell studiert werden, wobei der erfolgreiche Erwerb der Kenntnisse von INF-B-310/INF-D-240 zur Teilnahme an INF-B-320 nötig ist. Praktisch ist allerdings die Organisation von INF-B-320 nur einmal im Kalenderjahr möglich, sodass sich bei einem verzögerten Start eine unverhältnismäßig lange Wartezeit auf seine Teilnahme ergibt. Daher wird, obwohl sie aus fachlicher Sicht notwendig ist, auf die formale Definition einer Modulabhängigkeit verzichtet. Es wird allerdings *dringend* empfohlen, vor Beginn von INF-B-320 die Kenntnisse von INF-B-310/INF-D-240 durch eine bestandene Klausur nachzuweisen.

2. Aufgabenstellungen und Projektpraktikumsbetreuung

Das Softwaretechnologie-Projekt INF-B-320 ist ein Projektpraktikum, in dem eine Projekt-Team (Praktikumsgruppe) auf der Basis mehrerer vorgegebener Frameworks und einer objektorientierten Programmiersprache, derzeit Java, eine Anwendung erstellt. Das Projektpraktikum vermittelt die Fähigkeit zur Softwareentwicklung im Team, eine Kompetenz, die zur Berufsqualifikation gehört. Auch wird geübt, wie man vom Kunden Anforderungen aufnimmt und ihm gegenüber die Projektergebnisse validiert. Daher sind für den Fortschritt im Lernerfolg Präsenzzeiten nötig. Da der zu lernende Softwareentwicklungsprozess phasengegliedert ist, ist für das Projektpraktikum ebenfalls eine Phasengliederung mit Meilensteinen nötig.

2.1. Tutoren, Kunden und Pflichtkonsultationen

Jedem Projektteam werden eine konkrete Aufgabenstellung und ein Tutor zugeordnet. Der Tutor kann in zwei Entwicklungsrollen auftreten, sowohl als softwaretechnischer Betreuer als auch als Kunde. Der Tutor führt wöchentliche Pflichtkonsultationen durch, um mit jedem Mitglied des Projektteams seinen Projektfortschritt zu besprechen, sowie, im Sinne des agilen Prozessmodells, in der Rolle des Kunden beratend dem Projektteam zur Seite zu stehen. Der Besuch dieser Konsultationen ist zwar verpflichtend, die Termine dafür werden aber in Absprache zwischen Projektteam und Tutor individuell festgelegt. Darüber hinaus ist jeder Studierende verpflichtet, sich regelmäßig auf den Webseiten und im Internetforum des Projektpraktikums über aktuelle Themen zu informieren. Es können Projektteams auch extern bei Firmen, die Kunden spielen, eingesetzt sein; für sie gelten die gleichen Randbedingungen.

2.2. Phasengliederung und Meilensteine

INF-B-320 ist in mehrere Projektphasen eingeteilt, die hintereinander abgeleistet, nachgewiesen und bestanden werden müssen. Die Meilensteine sind aus fachlicher Sicht sequentiell voneinander abhängig, weil die weiter hinten liegenden Phasen die Studienleistungen der weiter vorne liegenden Phasen benötigen.

Die Projektlaufzeit beträgt in der Regel insgesamt 12 Wochen. Als Softwareprozess wird ein agiler Prozess verwendet, der in wöchentliche Sprints eingeteilt ist. Deren Aufgaben werden in den Pflichtkonsultationen an einzelne Projektmitglieder, an Teilteams, oder an das Team insgesamt verteilt. Bei einer Verteilung an Teilteams oder das Team insgesamt wird darauf geachtet, dass die individuellen Arbeitsergebnisse unterscheidbar sind. Die Ergebnisse der verteilten Aufgaben werden in den Pflichtkonsultationen präsentiert, was zu einem Sprint eines agilen Softwareprozesses gehört. Im Semester werden 9 Sprints durchgeführt.

Die Ergebnisse der Sprints, sowohl für einzelne Mitglieder, Teilteams als auch für das gesamte Projektteam, werden dann in der Regel von den Tutoren im Sinne einer Vorkorrektur zur Bewertung dem Prüfer vorgeschlagen. Der Prüfer prüft die Projektergebnisse und die Vorkorrektur der Tutoren abschließend als bestanden bzw. nicht bestanden. Bei einer Bewertung mit „nicht-bestanden“ kann das Projektmitglied, das Teilteam oder das gesamte Projektteam Einsicht in die Korrektur verlangen.

Die 9 Sprints sind darüber hinaus in 6 Phasen mit abschließendem Meilenstein eingeteilt (Objektorientierte Anforderungsanalyse, Objektorientierter Entwurf, Objektorientiertes Programmieren Runde I, Objektorientiertes Programmieren Runde II, Objektorientiertes Programmieren Runde III, Objektorientiertes Programmieren Runde IV, siehe Abschnitt 3). In Abhängigkeit vom Meilenstein sind Phasenergebnisse dem Tutor und dem Prüfer vorzulegen, wozu Modelle, Dokumente, lauffähige Prototypen, und am Ende der Programmierphasen, eine getestete und lauffähige Anwendung gehören. Die Phasenergebnisse bilden die Teile des Portfolios. Die Ergebnisse eines Meilensteins werden dann in der Regel von den Tutoren im Sinne einer Vorkorrektur zur Bewertung dem Prüfer vorgeschlagen. Der Prüfer prüft die Projektergebnisse unter dem Einbezug der Vorkorrektur der Tutoren abschließend als bestanden bzw. nicht bestanden. Der Natur des Softwareprozesses entsprechend muss jeder Meilenstein bestanden werden. Zusätzlich ist am Ende der Phase „Objektorientierter Entwurf“ der entstandene Prototyp

in einer Zwischenverteidigung zu präsentieren. In externen Projekten nimmt daran der reale Kunde teil.

Als Grundlage zur Bewertung der Ergebnisse eines Sprints oder Meilensteins dienen die Bewertungskategorien in Abschnitt 4. Die Bewertung ist Gegenstand der Pflichtkonsultationen, über die ein Protokoll angefertigt wird. Wird eine Studienleistung (Ergebnis eines Meilensteins) im Erstversuch mit „Nicht bestanden“ bewertet, kann diese maximal zweimal wiederholt werden. Die Wiederholungsversuche zur Erbringung der Studienleistung dürfen insgesamt eine Zeitdauer von 2 Wochen nicht überschreiten.
Nach der insgesamt zweimaligen nicht-bestandenen Wiederholung kann der Prüfer ein Nicht-Bestehen des Projektpraktikums feststellen.

Die Fertigstellung der zu erstellenden Anwendung muss drei Wochen vor dem Ende der Vorlesungszeit erfolgen, um die Anwendung zu stabilisieren und das Referat, d. h. die Abschlusspräsentation, vorbereiten zu können.

2.3. Referat (Abschlusspräsentation)

Das Portfolio enthält als letzte Studienleistung ein Referat, eine Präsentation des Projektes. Dazu gehört sowohl die Vorstellung des Entwicklungsprozesses als auch die Vorführung des entwickelten Softwareprodukts. Zum Projektende sind diese Folien sowie alle Ergebnisse des Projektpraktikums, die im nächsten Abschnitt geschildert werden, in einer vorgegebenen strukturierten Form in einem Portfolio bereit zu stellen. Diese Präsenzpflicht ist durch die Natur der Lehrform angezeigt.

2.4. Qualitätskontrolle des Softwareentwicklungsprozesses

Die Anwendungen werden in den Programmierphasen durch Werkzeuge des Testens (Continuous Integration) und Programmanalyse einer automatischen Qualitätskontrolle unterzogen. Der Quellcode und die Projektergebnisse müssen in einem für den Kurs bereitgestellten Repository eines Versionsverwaltungssystems abgelegt werden.

Um eine Qualitätskontrolle des Softwareentwicklungsprozesses verfolgen zu können, muss der Studierende, wie in einem richtigen Software-Projekt, seine Aufwände dokumentieren. Diese Daten werden zur Qualitätsanalyse der Lehre und ihrer Optimierung eingesetzt.

2.5. Ausscheiden von Gruppenmitgliedern

Im Falle, dass einzelne Mitglieder aus der Projektteams ausscheiden, wird die Aufgabenstellung im Umfang an die verbliebenen Mitglieder neu verteilt und angepasst. Dazu werden die bereits ausgeführten Aufwände aller verbleibenden Projektmitglieder analysiert und die noch zu leistende Arbeit neu auf die Gruppe verteilt. Es wird darauf geachtet, dass die Arbeitslast jedes verbliebenen Gruppenmitglieds im Rahmen der Workload des Moduls bleibt und dass es eine gute Chance hat, das so in seinem Umfang reduzierte Projektergebnis zu erbringen.

3. Meilensteine und deren Ergebnisse (Deliverables)

OOA: Objektorientierte Anforderungsanalyse

- Pflichtenheft
- Erweiterung der von der Professur zur Verfügung gestellten Musteranwendungen zur Einarbeitung in die Technologie (pro Teammitglied)

OOD: Objektorientierter Entwurf

- Kleiner Anwendungsprototyp pro Teammitglied
- Testplan
- Entwicklerdokumentation

OOP I: Objektorientiertes Programmieren Runde I

- Basisfunktionalität der Anwendung (einschließlich Tests)

OOP II: Objektorientiertes Programmieren Runde II

- Erfüllung der Muss-Kriterien (einschließlich Tests) als Voraussetzung für das Crostesting

OOP III: Objektorientiertes Programmieren Runde III

- Ergebnisse des Crosstestings
- Implementierung der Kann-Kriterien (einschließlich Tests)

OOP IV: Objektorientiertes Programmieren Runde IV

- Fertige Anwendung
- Fertige Dokumentation einschließlich Javadoc-API-Dokumentation
- Auswertung der erreichten Lernziele durch jedes Teammitglied

4. Bewertungskategorien

Die Bewertungskriterien sind im Softwareentwicklungsprozess von INF-B-320 in Kategorien eingeteilt. Für einzelne Kriterien jeder Kategorie vergibt das Betreuungspersonal an jedem Meilenstein dem Projektteam eine Güteeinschätzung. Diese Einschätzung dient in erster Linie als fachliches Feedback für die Projektgruppe. Bei Weiterentwicklung des Kurses können weitere Kriterien definiert werden, sie sind aber zu Beginn des Projektpraktikums bekanntzugeben.

Die Bewertungskategorien mit den Kriterien sind:

Softwareentwicklungsprozess

1. Pflichtenheft
2. Qualität der UML-Dokumente

2.1. in der Analysephase

2.2. in der Entwurfsphase

3. Anwendung der Analysemethode, z.B. CRC-Karten-Methode
4. Benutzerschnittstellenentwurf
5. Großer Prototyp
6. Wiederverwendung von Frameworks und Klassenbibliotheken
7. Begründung von Entwurfsentscheidungen
8. Qualität des Testens
   8.1. Junit-Tests durch die einzelnen Entwickler
   8.2. Crosstesting
   8.3. Testabdeckung
9. Planmäßigkeit der Entwicklung
   9.1. Forward Engineering
   9.2. Termintreue
10. Versionsmanagement mit Git und Arbeit mit GitHub
11. OOP_III
    11.1. Stabilisierung der Anwendung
    11.2. Erfüllung des Kundenwunsches

Anwendung/Endprodukt

1. Erfüllung des Pflichtenheftes
   1.1. Musskriterien (OOP_II)
   1.2. Kannkriterien (OOP_III)
2. Funktionsumfang
3. Zuverlässigkeit (Robustheit, Fehlertoleranz)
4. Benutzbarkeit/Ästhetik/Verständlichkeit/Navigation (Usability)
5. Wartbarkeit (Qualität der Erfüllung zusätzlicher Kundenwunsch)
6. Codequalität (statische Codeanalyse (Sonarqube))
7. Qualität Javadoc

8. Anwenderdokumentation

Projektmanagement

1. Planmäßigkeit der Entwicklung
2. Protokolle der Treffen und Kontrolle der Einhaltung der Festlegungen
3. Protokollierung der Arbeitsaufwände

Teamarbeit

1. Auftreten als Team nach außen (Tutor/Kunde)
2. Auftreten der Teammitglieder im Team
3. klare/gerechte Aufgabenteilung
4. Kommunikation mit dem Tutor
5. Umgang mit Problemen und Konflikten
6. Selbstkritische Einschätzung durch das Team

Referat/Abschlusspräsentation (AP)

1. Zwischenpräsentation
2. AP Zeiteinhaltung
3. AP Qualität des Vortrages
4. AP Qualität der Vorführung
5. AP Diskussion/Reaktion auf Fragen