Leitfaden zur Durchführung von fachspezifischen Projekten in den Studiengängen Software Engineering

Stand: 03.07.2017

H. Partsch, M. Dausend, A. Nassal, A. Raschke, M. Tichy

1. Einleitung

Das Anwendungsprojekt (AP) Software Engineering (SE) soll den Studierenden des Studiengangs Software Engineering (Bachelor) die Gelegenheit bieten, neben dem Softwaregrundprojekt (SoPra) ein weiteres umfangreicheres Individualprojekt durchzuführen. Analoges – allerdings mit modifizierter Schwerpunktsetzung gilt für die Projekte Software Engineering 1 (PSE1) und Software Engineering 2 (PSE2) im Master-Studiengang Software Engineering. Alle diese Projekte können von allen Instituten der Fakultät für Ingenieurwissenschaften und Informatik angeboten werden. PSE1 und PSE2 können darüber hinaus auch in Kooperation mit anderen Fakultäten oder industriellen Partnern durchgeführt werden. Dieses Dokument dient dabei als Leitfaden für die Vorgehensweise bei der Planung, Durchführung und Bewertung.

Dieser Leitfaden dient dem Zweck, die Lernziele der Projekte (siehe Abschnitt 3) zu erreichen und die Projekte nicht durch unpräzise und wechselnde Anforderungen, schlecht dokumentierte Schnittstellen oder zu großen Abhängigkeiten von Vorarbeiten in Form von Hardware oder Software von vornherein zum Scheitern zu verurteilen.

2. Organisatorischer Rahmen

Das Anwendungsprojekt Software Engineering (AP SE) ist ein Modul im Studiengang Software Engineering (Bachelor) mit einem Umfang von 12 Leistungspunkten (LP) bzw. 6 Semesterwochenstunden (SWS). Die Projekte PSE1 Und PSE2 sind Module im Masterstudiengang Software Engineering mit einem Umfang von 12 LP / 6 SWS bzw. 16 LP / 8 SWS. Nach Studienplan ist es vorgesehen, AP SE über 2 Semester à jeweils 6 LP laufen zu lassen, es ist aber auch denkbar (und insbesondere für Studierende, welche ein Auslandssemester einschieben möchten, vorteilhaft), das Projekt AP SE in einem Semester mit dann 12 LP Umfang durchzuführen. Allerdings erfordert dies erfahrungsgemäß ein höheres Engagement seitens der Studierenden.

3. Lernziele

AP SE, PSE1 und PSE2 haben *nicht* als primäres Ziel, dass die Anbieter eines Projekts ein Stück lauffähige Software erhalten. Dies ist sicherlich wünschenswert, die für den Studiengang vorgesehenen Lernziele sind aber wie im Folgenden erläutert:

Die Studierenden setzen das in vorherigen Veranstaltungen (Softwaretechnik) erlangte Wissen und die im SoPra gemachten Erfahrungen ein, um ein möglichst realistisches Projekt von Projektstart bis -

abschluss erfolgreich umzusetzen. Nach der Durchführung des AP SE besitzen die Studierenden erste Erfahrungen für ein erfolgreiches Projektmanagement. Sie können Anforderungen strukturiert dokumentieren sowie eine für das Problem adäquate Architektur entwerfen und auf verschiedenen Plattformen realisieren. Die Projektteilnehmer können die Bedienbarkeit einer Software bewerten und eine gut benutzbare Oberfläche realisieren. Sie sind fähig, in den verschiedenen Projektphasen jeweils sinnvolle Qualitätssicherungsmaßnahmen einzuplanen und durchzuführen. Die Studierenden dokumentieren und reflektieren ihre Vorgehensweise sowie sich ergebende Probleme und deren Lösung.

Die für das AP SE genannten Lernziele gelten im Prinzip auch für die Projekte PSE1 und PSE2, wobei für das Erreichen dieser Ziele eine deutlich höhere Selbständigkeit der Studierenden erwartet wird, so dass der Betreuer (s.u.) vor allem eine Beraterfunktion hat. Zudem werden die Studierenden stärker gefordert, Erfahrungen aus AP SE oder auch PSE1 positiv einzubringen. Außerdem kommen weitere Lernziele hinzu: Nach der Durchführung von PSE1 und PSE2 haben die Studierenden fundierte Fähigkeiten für erfolgreiches Projektmanagement, insbesondere in den Bereichen Aufwandsschätzung, Zeit-, Kosten-, und Ressourcenplanung. Sie können Anforderungen systematisch erheben und geeignet dokumentieren. Nicht-funktionale Anforderungen wie z.B. rechtliche Aspekte, Datenschutz und Datensicherheit werden von den Studierenden ebenfalls erkannt und berücksichtigt.

Neben diesen fachlichen Zielen sollen die Soft Skills gestärkt werden. Die Studierenden steigern ihre Kommunikationskompetenz, die Fähigkeit mit Fachleuten anderer Disziplinen zusammenzuarbeiten, insbesondere, diese zu verstehen und sie über den Projektfortschritt zu unterrichten. Da alle Projekte in der Regel durch mindestens 3 Personen durchgeführt werden sollen, spielt die Entwicklung der Teamfähigkeit eine große Rolle. Das Benutzen von verschiedenen für Software Engineering typischen Werkzeugen steigert die Kompetenz, mit diesen Werkzeugen umzugehen und das jeweils passende Werkzeug auszuwählen.

4. Rollen

In AP SE, PSE1 und PSE2 gibt es die folgenden Rollen:

Kunde (Anbieter/Auftraggeber/Betreuer)
Kunde ist der Anbieter eines Projekts. Er dient als Ansprechpartner für inhaltliche Fragen des Projekts. Typischerweise sind dies in AP SE Professoren oder Mitarbeiter eines Instituts der Fakultät für Ingenieurwissenschaften und Informatik. In PSE1 und PSE2 können die Kunden auch aus anderen Fakultäten oder der Industrie kommen.

Studierende (Projektteam/Auftragnehmer)
Studierende führen ein Projekt durch. Die Gruppe der Studierenden definiert evtl. intern weitere Verantwortlichkeiten wie Projektleiter, Dokumentation, Qualitätssicherungsingenieur, Architekt, etc.

Koordinator
Der Koordinator ist der für die spezifischen Projekte in den Studiengängen Software Engineering verantwortliche Mitarbeiter des Instituts für Programmiermethodik und Compilerbau (derzeit Dr. Alexander Raschke). Er organisiert die Bekanntgabe und Verteilung der Projekte, dient als Ansprechpartner hinsichtlich software-technischer oder organisatorischer Fragen für das Projekt und aktualisiert diesen Leitfaden auf Basis der gesammelten Erfahrungen.

5. Ablauf

Inhalt der Projekte ist in der Regel die Entwicklung eines (neuen) Softwaresystems.

Dieser Abschnitt gibt einen Überblick über den geplanten Projektverlauf von Angebot über Annahme bis zur Abnahme. Die einzelnen Phasen werden in den nachfolgenden Abschnitten detaillierter mit Begründungen erläutert. Mit * gekennzeichnete Informationen betreffen nur die Projekte PSE1 und PSE2.

Phase	Tätigkeiten	Beteiligte	Ergebnisse
Projektvorbereitung	Vorschlag erarbeiten	Kunde	Initialer Projektantrag mit u.a. (s.u.) informeller Anforderungsbeschreibung, Zieldefinition (Lastenheft)
	Vorschlag abstimmen	Koordinator, Kunde	endgültiger Projektantrag mit detaillierter Projektbeschreibung, (grober) Aufwandsschätzung, Rahmenbedingungen, Betreuungskonzept, Abnahmekriterien
Projektvorstellung	Themenvorstellung	Koordinator, Kunde, Studierende	Themenüberblick
	Themenvergabe	Koordinator, Kunde, Studierende	Teambildung, Themenzuteilung
Projekteinführung	Projektplanung*	Kunde, **Projektteam**	Vorgehensmodell, Projektplan, Rollen mit Verantwortlichkeiten, Aufwands-/Kostenschätzung*, Risikoanalyse/Machbarkeits-studie*
	Präsentation: Ideen & Projektplanung	**Projektteam,** Kunde, Koordinator	Grobplanung, Kritik
Projektdurchführung (klassisch)	Analyse (inkl. Anforderungs-erhebung*	Projektteam	Pflichtenheft (auch mit nicht-funktionalen Anforderungen*)
	Design/Entwurf		Architekturentwurf
	Implementierung		Code
	Integration		lauffähige Software
	Test		Testplan & Testkriterien
	Qualitätssicherung		QS-Planung & QS-

	Dokumentation		Dokumentation Sitzungsprotokolle, Handbuch, …
	Organisation/ Administration	**Projektteam,** Koordinator, Kunde	Projekttagebuch, Zeitaufwand, Protokolle
	Präsentationen	**Projektteam**, Koordinator, Kunde	Präsentationsunterlagen
Projektdurchführung (agil)	Sprints (inkl. Anforderungs-erhebung, Implementierung, Integration, Test)	Projektteam	Backlog, Sprintlog, Code, Testfälle
	Dokumentation		Sitzungsprotokolle, Handbuch, …
	Organisation/ Administration	**Projektteam,** Koordinator, Kunde	Projekttagebuch, Zeitaufwand, Protokolle
	Präsentationen	**Projektteam**, Koordinator, Kunde	Präsentationsunterlagen
Projektabschluss	Projektabnahme	Koordinator, Kunde, Projektteam	Kundenabnahme, alle Unterlagen vollständig
	Präsentation der Ergebnisse	Studierende, Interessierte	Präsentation der Ergebnisse für Interessierte (Werbung für neue Projekte)

5.1 Projektvorbereitung

Folgenden Informationen müssen von dem Kunden im Rahmen der Projektvorbereitung zusammengetragen und definiert werden:

- **Verantwortliche Person**
  Die festgelegte Person muss den Studierenden für Fragen zur Verfügung stehen. Sie muss inhaltliche Fragen zeitnah beantworten können. Sie trifft – wo nötig – Entscheidungen.

- **Ziel und (informelle) Anforderungsbeschreibung**
  Die informelle Anforderungsbeschreibung dient dem Studierenden dazu, sich über den Inhalt zu informieren und das Thema zumindest grob abschätzen zu können. Auch die Projektart wird damit festgelegt (Webanwendung, Desktopapplikation, eingebettetes System, Verwaltungssoftware, komplexe Berechnungen, umfangreiche Datenmanipulation,

Reimplementierung, Migration…).
Außerdem soll damit sichergestellt werden, dass der Kunde sich genaue Gedanken über den Inhalt und das (fachliche) Ziel des Projekts Gedanken macht und einen unter den Randbedingungen realisierbaren Rahmen absteckt.

- **Grobplanung mit grob geschätztem Aufwand und Anzahl der gewünschten/benötigten Teilnehmer**
  Dies dient dazu, den Aufwand für die Studierenden transparenter zu machen und den Kunden zu zwingen, den Umfang realistisch einzuschränken. Da die Studierenden einen nicht zu vernachlässigenden Aufwand in typische softwaretechnische Aufgaben wie Projektplanung und -organisation, (Projekt-)Dokumentation, Analyse, Entwurf und Qualitätssicherung stecken müssen, ist (nach gängigen Erfahrungswerten) bei AP SE und PSE1 mit nicht mehr als **120-150h reiner Implementierungsaufwand pro Teilnehmer** zu schätzen (bei 12 LP à 30h = 360h Gesamtaufwand). Entsprechend ergibt sich bei PSE2 ein reiner Implementierungsaufwand von 180-210h pro Teilnehmer (bei 16 LP = 480h Gesamtaufwand).

- **Abnahmekriterien**
  Die für die erfolgreiche Abnahme am Ende notwendigen Kriterien werden kurz zusammengefasst beschrieben. Dabei sollte der Schwerpunkt auf den projektspezifischen Anforderungen liegen.

- **Einschränkungen**
  Die zur Durchführung des Projekts festgelegten Einschränkungen, wie z.B. zu verwendende Programmiersprachen, Programmierrichtlinien, Frameworks, Software und Hardware müssen festgehalten werden. Wenn ein bestimmtes Vorgehensmodell gewünscht wird, muss auch dieses definiert werden.

- **Voraussetzungen**
  Für die erfolgreiche Projektdurchführung notwendiges Wissen aus Büchern, Vorlesungen, Literatur muss dokumentiert und ggf. der Umfang des Projekts so eingeschränkt werden, dass den Studierenden Zeit bleibt, sich dieses Wissen anzueignen.

- **Glossar**
  Ein anfängliches projektspezifisches Glossar kann Sinn machen, damit evtl. auftretende Fachbegriffe im entsprechenden Kontext eindeutig definiert sind.

- **Dokumentation**
  Vorhandene Schnittstellen (HW/SW) und zu benutzende Software müssen ausführlich und präzise dokumentiert sein, damit die Studierenden eine vernünftige Planung machen können, ohne sich langwierig durch undokumentierten Code quälen zu müssen.

- **Infrastruktur**
  Eine gewisse softwaretechnische Infrastruktur (Konfigurations-/Versionierungswerkzeuge, Webserver, CASE-Tools, etc.) wird zentral zur Verfügung gestellt. Der Kunde hat sich darüber hinaus um die Organisation der folgenden Infrastruktur zu kümmern:

- Räumlichkeiten, in denen das Projektteam zusammen arbeiten kann (möglichst mit Tafel, Möglichkeit Informationen länger zu lagern); ggf. sind auch die vorhandenen PC-Pools dafür geeignet (sofern die notwendige Software installiert ist und die Studierenden die benötigten Rechte haben)
- Benötigte Hard- und Software (PC, Laptop, spezielle HW)
- Zugänge zu vorhandener Software/Daten und zu Räumlichkeiten (Schlüssel)

Der Koordinator ist für folgende Aspekte verantwortlich:

- **Infrastruktur**
  - Konfigurationsmanagement (SVN, Git, TeamFoundationServer, …)
  - Projektmanagementwerkzeuge (webbasiert)
  - Projektverzeichnisse (Samba, NFS, …)
  - Wiki (Mediawiki, TikiWiki, …)
  - Tracking-Werkzeuge (Trac, Bugzilla, Gitlab, …)
  - Continous Integration Server (Jenkins)
  - Ggf. virtuelle Server
  - Webspace (Apache, nginx)
  - Diverse CASE-Tools (z.B. Enterprise Architect, Rational DOORS, Rational Rhapsody, Rational Statemate, …)

- **Verwaltung der Projektangebote**
  Projektvorschläge werden beim Koordinator eingereicht, der diese gesammelt den Studierenden zur Verfügung gestellt.

5.2 Projektvorstellung

Am Ende eines jeden Semesters wird im Rahmen einer Projektvorstellung die möglichen Projekte für das nächste Semester von den Kunden vorgestellt.

5.3 Projekteinführung

Im Rahmen der Projekteinführung findet sich das dem Projekt zugeordnete Team der Studierenden zusammen und baut eine Projektumgebung auf (Zugänge, Softwarewerkzeuge, Hardware, Kommunikationsplattform, etc.).

Anschließend werden in einem Zeitraum von ca. 2-4 Wochen folgende Details in Absprache mit dem Team, dem Koordinator und dem Kunden festgelegt und dokumentiert (Informationen, die die Projekte PSE1 und PSE2 betreffen, sind auch hier wieder mit * markiert):

- Das zu verwendende Vorgehensmodell (z.B. klassisches Phasenmodell, V-Modell, Agiles Vorgehen, RUP, entsprechend angepasste Versionen davon)
- Rollen/Verantwortlichkeiten
  Die Studierenden definieren unter sich Verantwortlichkeiten für Teilaspekte des Softwareentwicklungsprozesses und dokumentieren diese.
- Verfeinerter Projektplan
  Basierend auf der Vorgabe vom Kunden, werden die inhaltlichen Aspekte des Projektplans mit eigenen Vorstellungen abgestimmt. Nach einer umfangreichen Analysephase können diese aber auch noch einmal angepasst werden.

- Aufwands-, Kostenschätzung auf Basis der vorhandenen Informationen*
  Wie beim Projektplan, kann auch dies nach der Analysephase angepasst werden müssen.
- Machbarkeitsstudie
  Bei technisch schwierigen Details kann zunächst eine Machbarkeitsstudie durchgeführt werden, um das Risiko des Projekts zu verringern.
- Plattform, Sprache, Programmierrichtlinien
  Die Studierenden einigen sich (in Absprache mit dem Kunden) auf die Plattform, die benutzte Programmiersprache und dafür geeignete Programmierrichtlinien.

Zum Abschluss der Projekteinführungsphase werden die gewonnen Erkenntnisse und mögliche Lösungsideen in einer Präsentation dem Kunden und dem Koordinator vorgestellt und diskutiert.

5.4 Projektdurchführung

Während der Projektdurchführungsphase liegt die Hauptarbeit beim Projektteam. In Abhängigkeit von dem gewählten Vorgehensmodell werden verschiedene Projektphasen durchlaufen. Typischerweise sollten aber die folgenden auf alle Fälle in irgendeiner Form vorkommen:

- **Analyse**
  Aufbauend auf der informellen Anforderungsbeschreibung des Kunden wird in Absprache mit diesem ein strukturiertes Anforderungsdokument (textuelle und/oder formal) erstellt, welches die Grundlage für die weitere Arbeit darstellt. Es sollte auch Modelle enthalten und kann auf einem rudimentären Prototypen aufbauen.
- **Entwurf**
  Das Entwurfsdokument enthält eine ausführliche Beschreibung der Systemarchitektur, der Systemschnittstellen, der verwendeten Bibliotheken und das zu benutzende Datenbankschema.
- **Implementierung**
  Die Implementierung besteht aus gut dokumentiertem, nach den festgelegten Programmierrichtlinien erstellten Code, evtl. Modellen und Generatoren.
- **Integration**
  Die verschiedenen entwickelten und getesteten Module werden in ein Gesamtsystem integriert und die dabei gewonnenen Erfahrungen dokumentiert.
- **Qualitätssicherung/Tests**
  In allen Softwareentwicklungsphasen werden verschiedene Qualitätssicherungsmaßnahmen eingeplant und durchgeführt (Reviews, Unit-Tests, Komponenten-, Integrations-, Systemtests, Usabilitytests, etc.). Verantwortlich für die Einhaltung des Plans und die Dokumentation der durchgeführten Tätigkeiten (Reviewprotokolle, Testprotokolle, etc.) ist der QS-Beauftragte/-Verantwortliche des Teams.
- **Dokumentation**
  Der Verantwortliche für Dokumentation achtet auf eine saubere und aktuelle Dokumentation der erstellten Software (Anforderungen, Entwurf, Code, Benutzerhandbuch, etc.). Darüber hinaus wird der Projektverlauf im Projekttagebuch dokumentiert (Arbeitszeiten, Probleme und Entscheidungen, Aktivitäten, Sitzungsprotokolle, etc.). Auch der Prozess wird mit evtl. Abweichungen (und Begründungen dazu) dokumentiert.
- **Treffen**
  Mit dem Kunden sollten regelmäßige Treffen spätestens alle 2 Wochen stattfinden um den

Projektfortschritt zu demonstrieren und Feedback einzuholen bzw. Fragen zu klären. Die in diesen Treffen besprochenen Inhalte werden in Ergebnisprotokollen festgehalten. Diese dienen insbesondere auch als Absicherung für die Studierenden!

- **Präsentationen**
  Ca. in der Mitte der Projektlaufzeit und am Ende werden Präsentationen vorbereitet, welche auch vor einem größeren Publikum (z.B. allen Institutsmitarbeitern) gehalten werden sollen, um auch von dieser Seite Feedback und Anregungen zu bekommen.
- **Abgabe**
  Am Ende des Projekts werden alle Ergebnisse übersichtlich zusammengestellt und dem Kunden zur Verfügung gestellt.

5.5 Projektabschluss

Im Rahmen des Projektabschlusses wird das Projekt von dem Kunden formal abgenommen. Die Projekte werden im Rahmen eines Projekttags einem breiten Publikum (allen Mitgliedern der Fakultät + Externe) vorgestellt.

6. Projektbewertung

In die Projektbewertung gehen (unter anderem) folgende Aspekte ein:

- Entwickeltes System aus Anwendungs- und software-technischer Sicht
- Erstellte Dokumente
- Arbeitsweise (insgesamt und individuell; Beitrag einzelner Teammitglieder)
- Präsentationen (inkl. Reflexion).

Die Gewichtung dieser Aspekte für die Gesamtbenotung ist jeweils projektspezifisch und wird vom Koordinator in Absprache mit dem Kunden festgelegt. Mögliche detaillierte Kriterien zur Bewertung der einzelnen Aspekte werden in einem separaten Dokument erfasst.

Prinzipiell soll der Erkenntnisgewinn im Vordergrund stehen, d.h. auch ein schiefgelaufenes Projekt kann (unter Umständen) zu einer guten Note führen, so lange die Studierenden entsprechend reflektiert die Gründe für das Scheitern ausfindig machen konnten und dies auch entsprechend dokumentieren. Faulheit, Unfähigkeit oder ähnliches kann natürlich kein akzeptabler Grund für Scheitern sein.

Die Bewertung der einzelnen Teammitglieder erfolgt in Abhängigkeit der anfangs abgesprochenen Rolle, je nachdem wie gut die einzelne Rolle ausgeführt wurde.

Die Abwechslung von Präsentation, Dokumentation und Erstellen von Inhalt soll es ermöglichen, ein breiter fundiertes Bild über die (Mit-)Arbeit der einzelnen Teammitglieder zu erlangen und jedem Mitglied die Möglichkeit bieten, sich mit seinen Stärken in das Team einzubringen (und entsprechend bewertet zu werden).

Bewertungsaspekte für SE-Projekte in verschiedenen Bereichen (Stand 5.4.2017)

Prozess(einhaltung)	Requirements Engineering	Architektur
• präzise definiert • befolgt (auch und gerade bei agilem Vorgehen!) und dokumentiert • Versionierungssystem benutzt • Continuous Integration • Protokolle von allen Treffen (inkl. Entscheidungen und Verantwortlichen für ToDos) • Rollendefinition + Zuordnung • Analyse des Projektverlaufs (z.B. Schätzungen, Meilensteintrendanalyse)	• Anforderungsdokument mit funktionalen und nicht-funktionalen Anforderungen • Backlog mit Userstories • Sprintlog mit verfeinerten Userstories • Verwendung von Werkzeugen • Schätzungen • Traceability von Anforderungen	• klares, wiederverwendbares Design • Design (in Modellen) dokumentiert • Verwendung und Benennung von Patterns • komplexere Abläufe beschrieben/dokumentiert/modelliert • Designentscheidungen begründet/dokumentiert • Modulares Design → Arbeitsteilung • Hohe Kohäsion, geringe Kopplung

Usability	Implementierung	Big system/platform, libraries
• Die GUI enthält eine (sinnvolle und hilfreiche) Hilfefunktion • GUI geht über das einfache Maß hinaus: Suchfunktionen für größere Listen, Shortcuts für wichtigste Funktionen, Experten-/Laienmodus, anpassbar an verschiedene Bildschirmgrößen/Plattformen, spezielle Features wie Touchbedienung oder Stifteingabe, Toolbars • Konsistentes Bedienkonzept • Usability-Tests • ISO 9241-110 beachtet	• ausreichend gute Kommentare • Coding Styleguide definiert und eingehalten • sinnvolle Aufteilung des Codes in wiederverwendbare (und testbare) funktionale Einheiten (Klassen, Methoden, Funktionen, Prozeduren, etc.) • Wahl von Plattform und Sprache (sinnvoll und begründet)	• an passenden Stellen verschiedene Libraries verwendet und dokumentiert • auf/für verschiedene Plattformen implementiert • Lizenzen berücksichtigt und eingehalten

Qualitätssicherung	Dokumentation	Ergebnis
• automatisierte Tests mit einer Coverage von > 90% • Issuetracker eingerichtet und verwendet, um Issues zu verfolgen • Reviews (Prozess eingehalten!) • statische Codeanalyse benutzt und dokumentiert • GUI-Tests automatisiert • Abnahmetests definiert und dokumentiert • Konsequent und begleitend (nicht nur am Ende) • Geplant und Durchführung überprüft	• Architektur verständlich und dennoch abstrakt aktuell beschrieben • Alle Informationen (Schritt für Schritt-Anleitung) für Aufsetzen der Entwicklung • Alle Informationen (Schritt für Schritt-Anleitung) für Deployment • Benutzerhandbuch • Daten(bank)modell • individueller Reflexionsbericht	• Anforderungen werden erfüllt • Arbeitsaufwand steht in vernünftigem Verhältnis zum Ergebnis • ansprechende Gestaltung der Oberfläche • gute Präsentation des Produkts • klare und gut dokumentierte Übergabe des Produkts an den Kunden (Installer, Beschreibung der Installation/Konfiguration) • keine nicht mit dem Kunden abgesprochenen Features