

Software-Projekte im Studiengang Angewandte Informatik an der OTH Amberg-Weiden

Allgemeiner Projektleitfaden / Prof. Dr. K. Hoffmann

Revisionen

Name	Datum	Bemerkung
K. Hoffmann (kh)	24.09.2002	Erstellt
kh	15.10.2002	„Lizenzrechtlicher Hinweis“ unter „Äußere Rahmenbedingungen“ ergänzt.
kh	24.10.2002	Punkt 4 „Benotung ...“ präzisiert.
kh	24.09.2003	Aktualisierung für das WS 2003/2004
kh	06.09.2004	Aktualisierung für das WS 2004/2005
kh	13.10.2005	Hinweis zur Datensicherung ergänzt
kh	29.09.2006	Näheres zum erwarteten Inhalt der Reflexionsberichte und Änderung der Notengewichtung in 60/40.
kh	23.09.2007	Erläuterung zu Zeitprotokoll und R.-bericht präzisiert; Angaben zur Labornutzungsordnung entfernt (jetzt separates Dokument)
kh	04.10.2008	Erläuterung zu Zeitprotokoll an verbesserte Vorlage angepasst.
kh	18.3.2009	Aktualisierung für das SS09
kh	22.2.2010	Reflexionsberichte nicht mehr an das Ende von Iterationen gekoppelt.
kh	10.6.2010	Bemerkung zu den zulässigen Tätigkeitskategorien im Zeitprotokoll ergänzt
kh	29.7.2010	Gewünschten Inhalt der Tätigkeitsbeschreibung im R.-bericht präzisiert, Präzisierung unter „regelmäßige Besprechungen“ und Hinweis zur Prüfungsanmeldung hinzugefügt.
kh	13.3.2014	Kleinere Überarbeitungen

Zusammenfassung

Dieses Dokument beschreibt allgemeine Rahmenbedingungen für die Veranstaltung „Software-Projekte“; es enthält keine Einzelheiten über konkrete Projekt-Aufgabenstellungen, Vorgehensmodelle oder technologische Details.

Der Inhalt dieses Dokuments stützt sich auf Erfahrungen, die an anderen Hochschulen mit vergleichbaren Veranstaltungen gemacht wurden – die Quellen hierzu sind:

Lichter Horst, Glinz Martin (Hrsg.), *SEUH 7 Software Engineering im Unterricht der Hochschulen* Zürich 2001 dpunkt.verlag

Die Idee der regelmäßigen „Reflexion“ ist übernommen von:

Rust Heinrich, *Softwarepraktikum im Informatikstudium, BTU Cottbus, Systematische und geplante Durchführung eines kleinen Software-Entwicklungsprojektes* (16.2.2000)

Inhalt

Software-Projekte im Studiengang Angewandte Informatik an der OTH Amberg-Weiden	1
Revisionen	1
Zusammenfassung	1
Inhalt	2
1 Zielsetzung und Abstimmung der Erwartungen	2
Allgemeine Zielsetzung	2
Konkrete Lernziele	3
2 Äußere Rahmenbedingungen	3
3 Abzuliefernde Arbeitsergebnisse	5
Übersicht	5
Näheres zu den erwarteten Inhalten der Arbeitsergebnisse	6
Wie sind die Arbeitsergebnisse abzuliefern?	11
4 Bewertung der Arbeitsergebnisse	12

1 Zielsetzung und Abstimmung der Erwartungen

Allgemeine Zielsetzung

Wir (d.h. alle studentischen Teilnehmer und Betreuer der Veranstaltung „Software-Projekte“) wollen während des gesamten Semesters im Software-Projektlabor ein „Experiment“ machen – dieses Experiment besteht in der weit gehend eigenständigen Durchführung eines Software-Entwicklungsprojekts durch die studentischen Teilnehmer.

Experimente werden gemacht, um etwas zu lernen: Wir wollen etwas über arbeitsteilige Software-Entwicklung im Team lernen; den Rahmen dazu gibt uns das zu entwickelnde Stück Software – das „Produkt“. Damit wir etwas lernen, muss ausreichend Zeit vorhanden sein, um Fehler machen und eigene Erfahrungen sammeln zu können. Um sich Erfahrungen bewusst zu machen, sie zu bewerten und Konsequenzen daraus zu ziehen, bedarf es ausreichender Reflexion, deswegen werden von jedem Teilnehmer persönliche Reflexionsberichte verlangt.

Neben gewissen technischen Herausforderungen müssen die Teilnehmer vor allem die Schwierigkeit meistern, die richtige Balance zwischen dem „Produkt“ und dem „Prozess“ zu finden:

- Eine Konzentration allein auf das „Produkt“ führt leicht in eine Situation, in der nur das „Fertigwerden um jeden Preis“ selbst unter unrealistischen Terminvorgaben und Anforderungen zählt: Es wird nur codiert (da man ja „fertig werden“ muss) - systematisches Vorgehen bei Systemanalyse, Entwurf, Test und auch die Reflexion über die gemachten Fehler und Erfahrungen kommen zu kurz. Unter solch einem Druck gerät man oft ins Chaos: es wird zwar jede Menge Code fabriziert, aber spätestens mit dem Näherücken des unverrückbaren Endtermins (= Semesterende) stellt sich heraus, dass man das System entweder überhaupt nicht zum Laufen bringt oder nur unter großer Mühe ein halb funktionierendes Produkt hinkommt (von dem man insgeheim schon weiß, dass seine Wartung und Weiterentwicklung ein Albtraum wäre).

- Eine Konzentration allein auf den „Prozess“ führt dagegen leicht zu einer Situation der Lähmung durch Prozessbürokratie: es wird nur „Papier produziert“ und es kommt kein ausführbares Programm heraus.

Der Eindruck, dass „Produkt“ und „Prozess“ gegensätzliche Extreme sind, zwischen denen man nur schwer die Balance halten kann, ist für ein Team, das einen geeigneten Prozess ausreichend gut trainiert hat, wahrscheinlich weniger stark; ein Team mit wenig Training empfindet dies aber sicher nicht so. Die Veranstaltung „Software-Projekte“ geht davon aus, dass das Team über wenig Übung mit dem „Ernstfall“ der Software-Entwicklung im Team verfügt; deswegen wird hier versucht, einerseits ausreichend Freiraum zu lassen, damit man etwas über die Vorgehensweise lernen kann¹, andererseits aber den Druck, „ein Produkt“ liefern zu müssen, in angemessener Stärke aufrecht zu erhalten.

Konkrete Lernziele

- Arbeit im Team, u.a.
 - Sich koordinieren und miteinander kommunizieren.
 - Sich gemeinsame Regeln geben (z.B. Codierrichtlinien, Standards für Dokumentation, Regeln für die effiziente Abhaltung von Team-Meetings, aber auch für die Art des Umgangs miteinander).
 - Probleme (Risiken) rechtzeitig erkennen und damit umgehen.
 - Unklarheiten (z.B. in der Problemstellung) frühzeitig erkennen und klären.
 - Arbeit planen und verteilen, Zeitschätzungen abgeben.
 - Zeitaufwände diszipliniert protokollieren.
 - Arbeitspläne und Zeitschätzungen durch Erfahrungen verbessern (Reflexion).
 - Sich in eine kleine Software-Entwicklungsumgebung eingewöhnen.
- Vorgehen nach einem geeignet zurecht geschneiderten „Prozess“, u.a.
 - Erstellen bestimmter „Arbeitsprodukte“.
 - Planen und Durchführen bestimmter „Aktivitäten“.
- Möglichkeiten zur Prozessverbesserung erkennen (Reflexion).
- Software-Entwicklungserfahrung („Methodenwissen“) festigen, u.a.
 - Systemanalyse – Erhebung der Anforderungen, Modellierung des Problembereichs (→ „konzeptuelles Datenmodell“)
 - Entwurf eines Systems – Zerlegung in Teilsysteme, Abstimmung der Schnittstellen.
 - Implementierung unter Einhaltung gegebener Codierrichtlinien.
 - Systematische Tests einplanen, entwerfen und durchführen.
 - Reviews einplanen und durchführen.

2 Äußere Rahmenbedingungen

- **Die Benutzung des Software-Projektlabors** unterliegt der separaten „**Nutzungsordnung für das Software-Projektlabor der OTH Amberg-Weiden**“; mit Ihrer Unterschrift auf der Teilnehmerliste erkennen Sie die darin enthaltenen Bestimmungen an.

¹ Aus diesem Grund dürften z.B. Projekte mit Industrieunternehmen, die ein kommerzielles Interesse verfolgen müssen und von daher eine Konzentration allein auf das „Produkt“ fast immer herausfordern, für die Veranstaltung „Software-Projekte“ eher schlecht geeignet sein.

- **Bearbeitungszeit:** Mitte März bis Mitte Juli – Projektabschluss rechtzeitig vor dem Beginn der Prüfungszeit (Zeitpuffer einplanen!).
- **Empfehlung zur Teamgröße:** untere Grenze (5-6 Teilnehmer) obere Grenze (10-12 Teilnehmer). Evtl. muss ein Studiengang auf mehrere Teams aufgeteilt werden – falls dies nötig sein sollte, kann die gleiche Aufgabe an mehrere Teams vergeben werden. In diesem Fall dürfen die Teams zwar Erfahrungen austauschen, jedoch keine Arbeitsergebnisse (kein „copy and paste“!).
- **Empfehlung zur Vorgehensweise:** Um das Risiko zu senken, dass nur Prozessbürokratie betrieben wird, ohne am Ende ein ausführbares Programm zu haben, sollten mehrere „Iterationen“ stattfinden. Am Ende jeder Iteration² soll das System auch in Form eines ausführbaren und getesteten Programms vorliegen, in dem die zu Beginn der Iteration geplanten Anforderungen umgesetzt sind. Das Vorgehen innerhalb einer Iteration kann nach Art eines Wasserfallmodells erfolgen. Spätestens am Ende einer Iteration ist eine Integration des Gesamtsystems nötig; durch regelmäßige Builds innerhalb einer Iteration kann diese Integration erheblich erleichtert werden. Das Ergebnis jeder Iteration wird vollständig in das Versionsverwaltungssystem eingecheckt und es wird eine eigene Konfiguration definiert, damit man in „schlimmen Fällen“ leicht zum (stabilen) Stand am Ende der vorigen Iteration zurückkehren kann. Auch für stabile Zwischenstände innerhalb einer Iteration können separate Konfigurationen sinnvoll sein. Die Detailplanung der nächsten Iteration erfolgt am Ende einer Iteration (oder möglichst schon etwas vorher). Drei oder vier Iterationen für das gesamte Projekt scheinen ein vernünftiger Kompromiss zwischen zu langen Iterationen einerseits und zu viel Planungsaufwand andererseits zu sein.
- **Persönliche Zeitprotokolle** Jeder Teilnehmer erfasst in einem Zeitprotokoll für jede seiner Aufgaben detailliert die tatsächlichen und die geplanten Zeitaufwände. Es ist wichtig, dass diese Zeitprotokolle exakt geführt und nicht nach einigen Wochen aus dem Gedächtnis mit geschätzten Zahlen ausgefüllt werden. In den zu liefernden Reflexionsberichten ist u.a. auf Abweichungen zwischen geschätzten und tatsächlichen Zeitaufwänden einzugehen. Es sind die Ursachen aus der Sicht des Teilnehmers und daraus gezogene Konsequenzen darzustellen.
- **Regelmäßige Besprechungen** Jedes Team vereinbart einen festen Termin für ein wöchentliches Team-Meeting, bei dem für alle Teammitglieder Anwesenheitspflicht besteht. Folgende Punkte sollten regelmäßig auf der Agenda stehen: der aktuelle Projektstand, Kurzbericht jedes Teilnehmers, woran er/sie gerade arbeitet, ggf. ob und welche Probleme dabei aktuell vorliegen³. Über jede Besprechung ist ein Protokoll zu verfassen. Das Team sollte darauf achten, dass Team-Meetings effizient abgehalten werden⁴. Wie sich aus den vorigen Vorschlägen zur Agenda entnehmen lässt, ist hier mit „Besprechung“ oder „Team-Meeting“ ein Treffen gemeint, das in erster Linie der besseren Organisation des Projektablaufs dient und daher i. A. das gesamte Projektteam betrifft; informelle Besprechungen im kleinen Kreis, in denen z.B. Mitglieder einer Arbeitsgruppe Detailfragen erörtern, sind hiermit nicht gemeint.

² Evtl. mit Ausnahme der ersten Iteration

³ Das gibt anderen Mitgliedern die Möglichkeit, Tipps zu geben oder Hilfe anzubieten

⁴ So sollte es z.B. zu Beginn eines Meetings eine Agenda geben, damit Besprechungen nicht „ausufern“. Es sollte auch darauf geachtet werden, ob bestimmte Punkte wirklich im gesamten Team besprochen werden müssen oder ob nicht etwa ein kleinerer Kreis genügt. Evtl. wählt sich das Team einen Moderator.

- **Empfehlungen für Rollen** Es scheint sinnvoll, zumindest folgende Rollen fest an ein bestimmtes Team-Mitglied zu vergeben:

Dem **Projektleiter** obliegen Planung und Verfolgung des Projektverlaufs; er hält Ausschau, wo Planung und Realität auseinander laufen und führt gemeinsam mit dem Team entsprechende Beschlüsse zur Kurskorrektur herbei. Es sollte zu den Regeln des guten Umgangs im Team gehören, dass das Team und der Projektleiter wirklich ernsthaft versuchen, sich gegenseitig zu unterstützen.

Ein **Build-Administrator** sollte dafür sorgen, dass möglichst immer ein funktionsfähiger regelmäßiger Build verfügbar ist. Der Build-Administrator stößt dazu regelmäßig einen Gesamtbuid auf dem aktuellen Stand aller Quellen an. Er kontrolliert, ob der Build vollständig und fehlerfrei durchlaufen wird und kontrolliert evtl. in einem Schnelltest („Smoke-Test“), ob bestimmte zentrale Funktionen (z.B. Programm starten, Datei mit Testdaten laden, ...) im neu erstellten Programm noch funktionieren. Schlägt dieser Schnelltest fehl, so gilt der Build als gescheitert. Im Falle eines gescheiterten Builds informiert der Build-Administrator das Team, und das Team betreibt die Reparatur des Builds mit höchster Priorität. Der Build-Administrator definiert außerdem am Ende jeder Iteration (evtl. auch öfter) mit Hilfe der Versionsverwaltung Konfigurationen.

- **Hinweis zur Datensicherung:** Im Software-Projektlabor wird der Inhalt der Versionsverwaltung während der Projektlaufzeit i. A. wie folgt gesichert:
 - An den Tagen Montag bis Samstag findet jeweils eine inkrementelle Sicherung statt.
 - An jedem Sonntag findet eine Komplettsicherung statt.

Es wird jedoch keine Gewähr für die Integrität der gesicherten Daten übernommen! Sie sollten sich daher nicht blind darauf verlassen, dass Daten automatisch gesichert werden! **Vielmehr bleibt jedes Team und jeder Teilnehmer für die Sicherung seiner Daten selbst verantwortlich!**

3 Abzuliefernde Arbeitsergebnisse

Übersicht

Was	Von wem	Wann	Bemerkungen
Individuelle Projektverfolgung durch jeden Teilnehmer			
Zeitprotokoll	Jeder Teilnehmer	Am Ende jeder Iteration. Endtermine sind im „Projektkalender“ vermerkt.	Die Protokolle sind <u>täglich</u> zu pflegen und aktuell zu halten (Stichproben während der Iteration!). Sie sind wichtiger Input für die Reflexionsberichte. Zeitprotokolle sind <u>sowohl</u> ausgedruckt (und bitte mit einer Klammer

			<u>zusammengeheftet!</u>) <u>als auch</u> in Dateiform (PDF-Format) zu liefern.
Reflexionsbericht	Jeder Teilnehmer	Berichtstermine sind im „Projektkalender“ vermerkt.	Damit nicht bei jedem Iterationsende ein Bericht fällig ist, können Berichtstermine unabhängig von den Endterminen für Iterationen angesetzt werden: ein Zwischenbericht in der Mitte des Semesters und ein Schlussbericht am Semesterende erscheinen ausreichend. Berichte sind <u>sowohl</u> ausgedruckt (und bitte <u>mit einer Klammer zusammengeheftet!</u>) <u>als auch</u> in Dateiform (PDF-Format) zu liefern.
Gesamtbilanz über alle Zeitprotokolle (als Teil des letzten Zeitprotokolls)	Jeder Teilnehmer	Nur am Ende des Projekts	Fasst die tatsächlichen Arbeitszeiten über alle Iterationen zusammen.
Fazit zum gesamten Projekt (als Teil des letzten Reflexionsberichts)	Jeder Teilnehmer	Nur am Ende des Projekts	Zieht ein Resümee aus allen Reflexionsberichten.
Projektverfolgung durch das Team			
Arbeitsplan	Team (bzw. Projektleiter)	Bei Projektbeginn und vor jeder Iteration	Liefert i. A. den Stoff für Team-Meetings: Feststellung des Ist-Zustands, Besprechung über Änderungen, etc.
Protokolle von Besprechungen	Team	Nach jeder Besprechung	Protokolle können wichtigen Input für Reflexionsberichte liefern.
Vorgabe und Kontrolle selbst aufgestellter Regeln			
Codierrichtlinien	Team	Ende der ersten Iteration	Sind Grundlage für die Durchführung eines Code-Reviews.
Mindestens ein Protokoll über ein Review (Code-Review oder Review des Lastenhefts etc.)	Team	Ende des Projekts	
Bestandteile des Produkts			
Detaillierte Festlegung im Projektauftrag.			Hierunter fallen neben Quellcode, z.B. Testfälle, Spezifikationen, Entwurfsdokumente, ...

Näheres zu den erwarteten Inhalten der Arbeitsergebnisse

Zeitprotokoll

Im Zeitprotokoll erfassen Sie Schätzungen für die von Ihnen übernommenen Arbeiten und die tatsächlich aufgewendete Zeit. Bilanz gezogen wird jeweils am Ende jeder Iteration. Am Ende des Projekts wird zusätzlich die Gesamtbilanz gezogen. Die Zeitschätzungen sollen Ihnen die Gelegenheit bieten, im Laufe der Zeit ein „besseres Gefühl“ für Zeitschätzungen zu entwickeln: Am Anfang des Projekts werden Sie sich aufgrund mangelnder Erfahrung möglicherweise grob verschätzen; vielleicht werden Sie aber im Laufe des Projekts lernen, Ihre Schätzungen zu verbessern⁵.

Verwenden Sie hierzu bitte die Excel-Datei „ZeitprotokollVorlage“; benennen Sie diese Datei bei Projektbeginn bitte in „Zeitprotokoll_<IhrVor-und Nachname>“ um (beim Ausdrucken wird in der Fußzeile automatisch der Name des Dokuments und des Tabellenblatts eingefügt, und man kann damit einzelne Seiten im Notfall besser zuordnen).

In der Vorlagendatei finden Sie Tabellenblätter „Zeitprotokoll für I1“ bis „Zeitprotokoll für I3“; jedes einzelne Blatt enthält das Protokoll für die entsprechende Iteration⁶. Ferner enthält die Vorlagendatei ein Tabellenblatt „Gesamtbilanz (alle Iterationen)“, das am Ende des Projekts zusätzlich benötigt wird.

Zu den Tabellenblättern für die einzelnen Iterationen:

Hier finden Sie Mustereinträge für Arbeitspakete und einzelne projektbezogene Tätigkeiten; jedes Arbeitspaket/jede Tätigkeit hat eine möglichst aussagekräftige Bezeichnung. Außerdem sind alle Arbeitspakete und Tätigkeiten wie folgt gruppiert:

Arbeitspakete und Tätigkeiten zur Koordination, Unterstützung, Dokumentation des Projektverlaufs - z.B.:

- Tätigkeiten zur Erstellung/Pflege der Zeitprotokolle und Reflexionsberichte
- Tätigkeiten im Zusammenhang mit Teammeetings
- Tätigkeiten des Projektleiters bzw. Build-Administrators

Für die Tätigkeiten in dieser Gruppe brauchen keine Zeitschätzungen abgegeben werden, es sind nur die tatsächlich verbrauchten Zeiten zu erfassen.

Arbeitspakete und Tätigkeiten, die zu Kerndisziplinen (Analyse, Entwurf, Implementierung, Test, etc.) gehören:

Für die Tätigkeiten in dieser Gruppe müssen zu Beginn jeder Iteration auch Zeitschätzungen abgegeben werden. In den Reflexionsberichten soll für jede Iteration, die im zugehörigen Berichtszeitraum beendet wurde, zwischen geschätzten und tatsächlichen Aufwänden verglichen werden.

Die Datei „ZeitprotokollVorlage“ enthält bereits Mustereinträge für Arbeitspakete und Tätigkeiten; Sie sollten daher eigentlich für jede Ihrer Tätigkeiten einen passenden Mustereintrag vorfinden, den Sie bei Bedarf kopieren können. Im Einzelfall können Sie

⁵ Schätzungen können auch dadurch besser werden, indem man sich angewöhnt, Arbeitspakete genauer zu definieren: Je genauer man weiß, was in einem Arbeitspaket drinsteckt, um so besser kann man den Aufwand dafür abschätzen.

⁶ Sollten mehr als drei Iterationen vorgesehen sein, kopieren Sie bitte eines dieser Tabellenblätter und passen Sie es entsprechend an.

jedoch auch noch nicht genannte Tätigkeiten hinzufügen – achten Sie in diesem Fall aber auf eine aussagekräftige Bezeichnung. Tätigkeiten, die Sie neu hinzufügen, sollen unter den vorhandenen Tätigkeitskategorien⁷ eingruppiert werden – bitte keine neuen Tätigkeitskategorien hinzufügen!

Sauber geführte Zeitprotokolle und deren termingerechte und vollständige Abgabe in ausgedruckter Form und als Datei (s.o.) sind notwendige Voraussetzung für das Erreichen der Note 4 (oder besser)!

Reflexionsbericht

Verwenden sie hierzu bitte die Word-Datei „ReflexionsberichtVorlage“; benennen Sie diese Datei wie folgt um: „Reflexionsbericht_<Ihr Vor-und Nachname>“ (beim Ausdrucken wird in der Fußzeile automatisch der Name des Dokuments eingefügt, und man kann damit einzelne Seiten im Notfall besser zuordnen).

Der Reflexionsbericht soll aus der persönlichen Sicht des Teilnehmers zu folgenden Punkten Stellung nehmen:

- Darstellung der persönlichen Tätigkeiten im Berichtszeitraum.
- Reflexion über den Berichtszeitraum:
 - Beschreibung der Ereignisse im Berichtszeitraum mit persönlicher Bewertung: was lief z.B. gut/schlecht und warum?
 - Auswertung der Zeitprotokolle: Wo gibt es gravierende Abweichungen zwischen Schätzung und Realität und wo sehen Sie die Ursachen für die Abweichungen?
 - Welche Lernergebnisse/Erfahrungen nehmen Sie für die Zukunft mit? Z.B.: Welche Konsequenzen ziehen Sie aus den Ereignissen und der Auswertung Ihrer Zeitprotokolle? Haben Sie Möglichkeiten erkannt, bestimmte Probleme künftig zu vermeiden bzw. abzumildern? Haben sich bestimmte Praktiken bewährt?
- Persönliches Fazit (nur im letzten Reflexionsbericht): Im letzten Reflexionsbericht soll jeder Teilnehmer in einem separaten Abschnitt aus all seinen Reflexionsberichten ein kurzes Resümee ziehen: Was sind z.B. Ihre wichtigsten Erfahrungen, die Sie aus dem Projekt mitnehmen?

Sie müssen im Reflexionsbericht keinen „langen Roman“ verfassen – der Inhalt sollte aber doch ausreichend detailliert und konkret genug sein, so dass man wirklich verstehen kann, was Sie im Berichtszeitraum getan bzw. beobachtet haben.

Bei der **Beschreibung Ihrer Tätigkeiten** genügt es nicht (!), im Telegrammstil nur relativ allgemeine und unspezifische Angaben wie etwa

- „Use-Cases beschreiben“
- „Implementierung“
- „Testen“
- „Recherche zu Thema XY“
- „Analyse des XY-Algorithmus“

⁷ „Reflexionsberichte und Zeitprotokoll“, „Team-Meetings“, „Projektplanung und -koordination“, „Regelmäßiger Build/“, „Analyse“, „Entwurf“, „Implementierung“, „Test“

zu machen; schreiben Sie stattdessen genauer und konkreter:

- Welche Use-Cases haben Sie beschrieben? Haben Sie die Beschreibung komplett neu erstellt oder überarbeitet? Worin bestand die Überarbeitung bzw. was war der Anlass dafür?
- Welche Klassen haben Sie implementiert? Wofür sind diese Klassen zuständig? Haben Sie den Code neu erstellt oder überarbeitet? Worin bestand die Überarbeitung?
- Haben Sie Testfälle erstellt oder Tests durchgeführt? Welche Testfälle bzw. Tests haben Sie erstellt bzw. durchgeführt?
- Achten Sie generell darauf, dass die von Ihnen erzielten Arbeitsergebnisse einer bestimmten Tätigkeit klar erkennbar sind: Welches Artefakt (Spezifikation, Diagramm, Quellcode(-teil)) ist als Ergebnis entstanden? Bzw. im Falle von „Recherchen“, „Untersuchungen“, „Analysen allgemeiner Art“: Was war die Fragestellung, die Sie klären wollten, und was waren die gewonnenen Erkenntnisse?
- Erklären Sie (zumindest in komplexeren Fällen) auch ein wenig zu den Hintergründen Ihrer Tätigkeit, z.B.: warum haben Sie bestimmte (etwa Design-) Entscheidungen gerade so getroffen und nicht anders? Was war die Ursache für ein bestimmtes Problem, das Sie durch Ihre Tätigkeit behoben haben?
- Nennen Sie generell die Namen aller an einer Tätigkeit beteiligten Kollegen, wenn Sie eine Aufgabenstellung in einer Gruppe mit mehreren Leuten bearbeitet haben. Worin bestand in diesem Fall Ihr Beitrag?
- Vermeiden Sie möglichst Formulierungen wie *„Um die Qualität der erstellten Dokumente zu sichern wurde eine interne Überprüfung des Pflichtenheftes durchgeführt ... Dabei musste man die Anforderungen und ... vergleichen“*. Formulierungen wie *„wurde durchgeführt“* und *„musste man“* klingen nach wenig eigener Beteiligung und Ihr Betreuer muss daraus nicht zwingend schließen, dass wirklich Sie etwas getan haben.

Sollte solch eine Beschreibung nicht für jede einzelne Tätigkeit möglich oder sinnvoll sein, weil Sie im Berichtszeitraum vielleicht besonders viele Tätigkeiten übernommen hatten, dann beschreiben Sie wenigstens die „wichtigsten“ Tätigkeiten (die Sie z.B. am meisten Zeit gekostet haben oder die Ihnen aus anderen Gründen besonders wichtig sind, weil Sie dabei vielleicht am meisten gelernt haben).

Ebenso sollten Sie bei der **Reflexion über den Berichtszeitraum** darauf achten, dass das Ganze konkret genug ist, um nachvollziehbar zu sein(!). Statt z.B. nur Binsenweisheiten im Telegrammstil aufzuzählen, sollten Sie genauer etwas über die konkrete Situation sagen, in der Ihnen etwas bewusst geworden ist: Schreiben Sie als Lernergebnis bitte nicht nur „Gute Kommunikation ist wichtig“ oder „Testen ist wichtig“ ohne zusätzliche Erläuterung, sondern achten Sie darauf, dass man erkennen kann, durch welches konkrete Ereignis Ihnen das bewusst wurde. Sollte dies nicht für alle Ereignisse und Lernergebnisse möglich sein, dann beschränken Sie sich auf das, was Ihnen am wertvollsten erscheint.

Ordentlich verfasste Reflexionsberichte und deren termingerechte und vollständige Abgabe in ausgedruckter Form und als Datei (s.o.) sind notwendige Voraussetzung für das Erreichen der Note 4 (oder besser)!

Arbeitsplan (dieser Punkt ist nur für den Projektleiter wichtig)

Der Arbeitsplan soll für alle Beteiligten den aktuellen Projektstand möglichst gut erkennen lassen (Wo stehen wir im Moment? Was fehlt uns noch? Wo gibt es Probleme/Risiken?). Er

wird vom Projektleiter erstellt und während des Projekts von diesem fortgeschrieben und an Erfahrungen angepasst. Änderungen am Arbeitsplan sollen nachvollziehbar sein.

Zu Beginn des Projekts ist der Arbeitsplan nur sehr grob – es wird grob angegeben, was der Inhalt der Iterationen sein soll und nur die erste Iteration wird detailliert geplant. Basierend auf den Erfahrungen einer Iteration soll die nächste Iteration geplant werden. Als grobe Strategie für die Verteilung der Arbeit auf Iterationen empfiehlt es sich, die am wichtigsten bzw. am riskantesten erscheinenden Dinge vorrangig in Angriff zu nehmen, um möglichst früh Risiken zu erkennen und gleichzeitig in jeder Iteration die jeweils „wertvollste“ Funktionalität zu realisieren.

Zeitaufwandsschätzungen werden wegen mangelnder Erfahrung wahrscheinlich zumindest am Anfang sehr grob sein. Die Führung und Auswertung der Zeitprotokolle soll aber dazu verhelfen, die Schätzungen zu verbessern. Die Zeitschätzung für bestimmte Arbeitspakete soll vom dafür verantwortlichen Teammitglied kommen – der Projektleiter arbeitet diese Schätzungen in seinen Arbeitsplan ein und greift korrigierend ein, wenn Probleme erkennbar werden.

Aus dem Arbeitsplan sollte insbesondere Folgendes erkennbar sein:

- Revisionen: Wer hat wann was geändert?
- Datum des aktuellen Planungsstandes.
- Aktuelle Iteration und eine detaillierte Planung für die aktuelle Iteration:
 - Ziel der aktuellen Iteration (beschrieben durch überprüfbare Meilensteine), z.B.:
 - Welche Anforderungen sind in welchem Umfang bis zum Ende der Iteration umgesetzt?
 - Welche Artefakte sind bis zum Ende der Iteration in welchem Ausbaurzustand?
 - etc.

Achten Sie bei der Festlegung der Meilensteine darauf, dass möglichst gut festgestellt werden kann, ob ein Meilenstein erreicht wurde oder nicht.

- Definition in sich geschlossener Arbeitspakete. Ein Arbeitspaket soll ein möglichst klar definiertes greifbares Ergebnis liefern. Es soll möglichst gut feststellbar sein, ob ein Arbeitspaket abgeschlossen ist oder nicht. Ein Arbeitspaket kann z.B. die Realisierung eines bestimmten Features oder die Erstellung eines bestimmten Artefakts zum Gegenstand haben. Zur Abarbeitung eines Arbeitspakets können mehrere verschiedene Tätigkeiten erforderlich sein, der Umfang sollte nicht zu groß aber auch nicht zu geringfügig sein, da die Arbeitspaketplanung sonst zu grob bzw. zu unübersichtlich wird. Zu jedem Arbeitspaket sollte folgende Information gehören:
 - Aussagekräftige Bezeichnung/Kennung des Arbeitspakets
 - Ggf. Kurzbeschreibung
 - Aufwandsabschätzung in Stunden (soll von den Bearbeitern stammen!)
 - Zuständige(r) Bearbeiter
 - Geplantes Abschlussdatum
- Ggf. Abhängigkeiten zwischen Arbeitspaketen
- Für jede Person eine Abschätzung der wöchentlich für das Projekt verfügbaren Arbeitszeit in Stunden
- Liste der „Top Ten“ Risiken für das Projekt möglichst mit Nennung von
 - Wie soll mit dem Risiko umgegangen werden? (Maßnahmen)
 - Wer ist für die Durchführung der Maßnahmen zuständig?

-
- Vorausschau auf künftige Iterationen in Form einer Grobplanung.

Besprechungsprotokolle

Verwenden Sie hierzu bitte die Vorlage `Besprechungsprotokoll_jj_mm_tt_Vorlage.doc` und benennen Sie die Datei mit Ihrem fertigen Protokoll jeweils um in `Besprechungsprotokoll_2014_05_02`, `Besprechungsprotokoll_2014_05_09`, etc. Durch die Angabe des Datums im Protokollnamen kann dann auch der Windows-Explorer die Protokolle in der richtigen zeitlichen Reihenfolge auflisten.

Der aktuelle Stand des Projekts, aktuell erkennbare Risiken für das Projekt, künftige Planungen sollten i.A. Themen in jeder Besprechung sein.

Codierrichtlinien

regeln die Gestaltung des Quellcodes. Hier muss kein umfangreiches Gesetzbuch verfasst werden, einige Anhaltspunkte für evtl. zu treffende Regelungen könnten aber sein:

- Kriterien zur Aufteilung des Codes auf Quelldateien, Namenskonventionen für Quellcodedateinamen
- Konventionen für Bezeichner (z.B. Klassennamen, Methodennamen, Variablennamen)
- Umgang mit gewissen Konstrukten der Programmiersprache - welche Konstrukte sollen z.B. vermieden oder nur eingeschränkt genutzt werden?
- Regelungen zur Dokumentation durch Kommentare (z.B. Beschreibung der Zuständigkeit jeder Klasse in einem Kommentar, ...)
- Regelungen zur Formatierung – Einrückungen, ...

Sie können natürlich auf allgemein übliche Regelkataloge (die sie z.B. im WWW gefunden haben) zurückgreifen und diese ggf. ergänzen.

Reviewprotokoll

Enthält folgende Information über ein Review:

Datum, Anfangszeitpunkt, Endpunktzeitpunkt, Anwesende und ihre Rollen (z.B. Reviewer), Name des Protokollführers, Kennzeichnung des Prüfobjekts (inkl. Versionsangabe), Einschätzung des Prüfobjekts durch die Reviewer, Liste der beschlossenen Nachbearbeitungen, Checklisten der einzelnen Reviewer.

Wie sind die Arbeitsergebnisse abzuliefern?

Mit Ausnahme der persönlich verfassten Dokumente (Reflexionsberichte und Zeitprotokolle), die Sie dem Betreuer jeweils selbst abliefern, sollen alle anderen geforderten Arbeitsergebnisse (Quellcode-Dateien, Diagramme, Spezifikationsdokumente, Protokolle, Richtlinien, Planungsunterlagen, ...) spätestens am vereinbarten Endtermin in ihrem Endzustand in die Versionsverwaltung eingecheckt sein und es muss eine entsprechende Konfiguration definiert worden sein. Dieser Endzustand muss im Arbeitsordner des Build-Admin „brennfertig“ vorliegen, d.h. die zugehörige Konfiguration ist vollständig aus der Versionsverwaltung in das Arbeitsverzeichnis des Build-Administrators abgerufen worden und es wurde ein Gesamtbuild für alle Build-Konfigurationen (Debug und Release) durchgeführt.

4 Bewertung der Arbeitsergebnisse

Die Benotung wird wie folgt vorgenommen:

- Es erfolgt keine („mündliche“) Benotung von Präsentationen, Diskussionsbeiträgen etc., die in Gegenwart des Betreuers z.B. in Teambesprechungen oder während der gemeinsamen Begutachtung der Ergebnisse einer Iteration erfolgen.
- Die Benotung stützt sich auf die explizit geforderten („schriftlichen“) Arbeitsergebnisse.
- Die Gesamtnote jedes Teilnehmers setzt sich wie folgt zusammen:
 - a) Eine „individuelle Note“ für jeden Teilnehmer stützt sich vor allem auf das Bild, das sich aus den persönlichen Zeitprotokollen und Reflexionsberichten ergibt, und stellt den individuell beeinflussbaren Anteil an der Endnote dar. Hier muss mindestens die Note 4 erreicht werden. Unerlässlich hierfür ist die sorgfältige Führung und vollständige Lieferung aller Zeitprotokolle und Reflexionsberichte.
 - b) Eine „Teamnote“ für jedes Team stützt sich auf den Gesamteindruck, den das vom Team gelieferte „Produkt“ und die Arbeitsweise des Teams vermitteln (dies stellt den durch Teamarbeit beeinflussbaren Anteil an der Endnote dar). Was außer Quellcode noch zum Produkt gehört, wird im Projektauftrag festgelegt. Bei der Bewertung des Produkts spielen neben der Frage, in wie weit das Produkt die Erwartungen an die geforderte Funktionalität erfüllt, u. a. auch die Qualität der Erfassung der Anforderungen, sowie die Qualität des Entwurfs, der Implementierung und der Tests eine Rolle. Auch hier muss mindestens die Note 4 erreicht werden.
 - c) Die „individuelle Note“ gemäß a) wird mit **60%**, die Teamnote gemäß b) mit **40%** in der Endnote jedes Teilnehmers gewichtet.

Hinweis zur Prüfungsanmeldung: Einen Teil Ihrer Prüfungsleistung erbringen Sie jeweils, wenn Sie z.B. die geforderten Berichte bzw. Zeitprotokolle abgeben oder Ihr Team bestimmte im Projektauftrag geforderte Ergebnisse vorlegt; dies geschieht i. A. nicht nur am Semesterende, sondern auch zu mehreren genau festgelegten Zeitpunkten während des Semesters. Aus diesem Grund erfolgt die offizielle Anmeldung zur Prüfung zu Beginn des Semesters, indem Sie sich mit Ihrer Unterschrift in eine Anmeldeliste eintragen. Danach können Sie nicht mehr von der Prüfung zurücktreten (abgesehen von Ausnahmefällen wie z.B. einer länger andauernden Erkrankung⁸): ein „Ausstieg“ aus dem Projekt hätte für Sie also die Note 5 zur Folge.

Ebenfalls noch wichtig: Bitte melden Sie sich gegen Ende des Semesters über die elektronische Prüfungsanmeldung der OTH auch zur Prüfung im Fach „Software-Projekte“ an, wie Sie das bei jeder anderen Prüfung tun. Dies hat für das Software-Projekt allerdings nur⁹ die Bedeutung, dass der Betreuer Ihre Note in das EDV-gestützte Notenerfassungssystem eintragen kann. Wenn Sie sich hier nicht anmelden, erfordert das im Prüfungsamt manuelle

⁸ Bitte beachten Sie in solch einem Fall die entsprechenden Regelungen der Prüfungsordnung.

⁹ Offiziell angemeldet haben Sie sich ja schon zu Beginn des Semesters durch Eintrag in die Teilnehmerliste.

Nacharbeit und Ihre Note wird deswegen nicht pünktlich am Tag der Notenbekanntgabe über die OTH-Webseite abrufbar sein.