

# Leitfaden zum Softwarepraktikum im Informatik- und IMT-Studium, BTU Cottbus: Systematische und geplante Durchführung eines kleinen Software-Entwicklungsprojektes

Dirk Beyer, Gerd Köhler, Heinrich Rust  
2003-09-18

## Inhalt

1 Vorbemerkung.....	1
2 Lernziele.....	2
3 Äußerer Rahmen.....	3
4 Hinweise zu Projektdurchführung.....	3
5 Wahrnehmung wichtiger Termine.....	4
6 Notwendige Bedingungen für den Scheinerwerb.....	5
7 Rollenzuteilungen – Koordination der Gruppenarbeit.....	6
8 Zum Text.....	7
Anhang: Zur Gestaltung der anzufertigenden Dokumente.....	8
1 Allgemeine Hinweise.....	8
2 Sitzungsprotokolle.....	8
3 Zeitprotokolle.....	8
4 Pflichtenheft.....	9
1.Zielbestimmung.....	9
2.Produkt-Einsatz.....	9
3.Produkt-Umgebung.....	9
4.Produkt-Funktionen.....	10
5.Produkt-Daten.....	10
6.Produkt-Leistungen.....	10
7.Benutzungsoberfläche.....	10
8.Qualitäts-Zielbestimmung.....	10
9.Globale Testszenarien/Testfälle.....	10
10.Entwicklungs-Umgebung.....	11
11.Ergänzungen.....	11
5 Arbeitspläne.....	11
6 Entwurfsdokumente.....	12
7 Stilfibel.....	13
8 Testpläne und Testprotokolle.....	13
9 Reviews und Reviewprotokolle.....	13
10 Checklisten.....	14
11 Gruppen- und Einzelreflexionspapiere.....	15
12 Gesamtprojektbericht.....	15
13 Literatur.....	16

## 1 Vorbemerkung

Dieses Dokument beschreibt den arbeitsorganisatorischen Rahmen für die Durchführung des Softwarepraktikums. Bitte lesen Sie es sorgsam und vollständig durch und besprechen Sie alle unklaren Punkte mit Ihrem Berater.

Technische Details können hier nicht vertieft werden. Hierfür sind Sie auf Ihre in vorhergehenden Semestern gesammelten Kenntnisse verwiesen. Wir setzen also voraus, dass Sie die Veranstaltungen Informatik I und Informatik II erfolgreich absolviert haben. Dies betrifft insbesondere Fragen des Designs von Softwaresystemen

(Programmierung im Großen) und des Designs und der Analyse von Algorithmen und Datenstrukturen (Programmierung im Kleinen). Auch bezüglich der Durchführung von systematischen Tests und von Reviews wird davon ausgegangen, dass Sie über die nötigen theoretischen Kenntnisse verfügen. Die in diesem Leitfaden angegebenen Hinweise sind nur als Erinnerungshilfen gedacht. Wenn Sie hier Lücken haben, dann finden Sie eine sehr schöne und pragmatische Einführung in das entsprechende Gebiet im Buch von Frühauf et al. [FLS91]. Als Nachschlagewerk in verschiedene Gebiete der Softwaretechnik geeignet sind die Bücher von Balzert [Bal96,Bal98], Pressman [Pre97] und Sommerville [Som92]. Sie finden aber zu allen technischen Fragen auch andere Literatur in der Bibliothek. Es ist **Ihre** Aufgabe, sich die nötigen technischen Kenntnisse und Fertigkeiten anzueignen, soweit Sie sie noch nicht erworben haben. Die Berater werden Sie aber dabei unterstützen, wenn Sie entsprechende Hilfe erbitten.

Dieser Leitfaden ist aus der Auffassung erwachsen, dass es Ihre Sache ist, zu lernen. Die Berater haben nur die Aufgabe, Ihnen diese Tätigkeit zu erleichtern. Diese abstrakte Auffassung wurde konkretisiert,

- indem Aufgabenstellungen vorgegeben werden, die hinreichend umfangreich und mehrdeutig sind, so dass einige praxisrelevante Probleme bei mangelnder Klärung von Missverständnissen mit dem Auftraggeber, den Folgen mangelnder Qualitätssicherung und generell knapper Projektlaufzeit erfahrbar werden;
- indem Sie in Projektteams zusammenarbeiten, so dass Sie voneinander lernen können, aber auch lernen müssen, ihre Arbeit miteinander zu koordinieren;
- indem Ihnen eine zyklische Organisation des Gesamtprojektes vorgeschrieben wird, so dass Erfahrungen aus früheren Teilprojekten in späteren berücksichtigt werden können;
- indem während der Arbeit an einem Teilprojekt die detaillierte Erhebung von Daten über den Aufwand, der für verschiedene Teilaufgaben anfällt, von Ihnen verlangt wird; und
- indem jedes Teilprojekt und das Gesamtprojekt von einer expliziten Reflexionsphase abgeschlossen wird, in der Pläne und wirklicher Projektablauf verglichen und aus dem Ergebnis Lehren gezogen werden, die in Folgeprojekten umgesetzt werden können.

Jedes Softwareprojekt unterliegt anderen Bedingungen. Daher ist es auch weniger wichtig, dass Sie versuchen, einen Standardprozess zu erlernen, der immer funktioniert - denn den gibt es nicht -, sondern dass Sie Reflexionsfähigkeiten erwerben, die Sie in die Lage versetzen, kurzfristig Fehlentwicklungen im Prozessablauf zu erkennen und entsprechende Konsequenzen zu ziehen.

Die Erfahrung zeigt, dass gewisse Vorgehensweisen eine solche Reflexion stärker unterstützen als andere. Diese Vorgehensweisen sind als Handwerkszeug der Softwareentwicklung zu verstehen, das bewährt ist und daher gewöhnlich funktioniert (wenn auch nicht immer). Zu diesen Vorgehensweisen gehören unter anderem die Aufteilung des Gesamtprojektes in weitgehend abgeschlossene Teilprojekte, die explizite Planung von Aufwänden und Ressourcen, die Führung von Zeitprotokollen, die Führung von Gesprächsprotokollen und die Anfertigung von Projektberichten im Anschluss an die Teilprojekte und das Gesamtprojekt.

## 2 Lernziele

Sie erarbeiten sich praktische Erfahrungen in den folgenden Gebieten:

- Arbeit im Team koordinieren, Probleme erkennen und lösen.
- Arbeitspläne erstellen und (regelmäßig) anhand von Erfahrungsdaten korrigieren.
- Unsicherheiten der Aufgabenstellung frühzeitig erkennen und klären.
- Regelmäßig den Prozessverlauf reflektieren und Ergebnisse für die Planung und Durchführung der Folge teilprojekte nutzen; Reflexionsergebnisse protokollieren. Hierzu führen Sie Zeitprotokolle über die projektbezogenen Tätigkeiten.
- Software entwerfen und Entwürfe dokumentieren.
- Ein größeres Programm implementieren.

- Kodier- und Dokumentierstandards explizit festhalten und einhalten.
- Reviews einplanen und durchführen.
- Systematische Tests einplanen und durchführen.
- Entwurfs-, Codier- und Testerfahrung sammeln.

### 3 Äußerer Rahmen

Das Praktikum wird im folgenden äußeren Rahmen durchgeführt:

- Der planmäßige Bearbeitungszeitraum ist während der Vorlesungszeit des Semesters.
- Die Gruppen haben etwa vier Mitglieder.
- Die gleiche Praktikumsaufgabe kann mehrfach vergeben werden. In diesem Fall dürfen (und sollten) die Gruppen, die dieselbe Aufgabe bearbeiten, Erfahrungen austauschen, aber keine Ergebnisdokumente.
- viermal während der Praktikumslaufzeit werden Teilprojekte abgeschlossen.
- Jede Gruppe wird während des Praktikums von einem Auftraggeber und einem Berater unterstützt. Der **Auftraggeber** ist in der Regel Mitarbeiter eines Lehrstuhls. Er stellt die Aufgabe, achtet auf die Erfüllung der Produktanforderungen und ist letztlich für die Vergabe des Leistungsscheins zuständig. Der **Berater** ist in der Regel ein Student oder eine Studentin im höheren Semester. Er oder sie berät die Gruppe bei der Projektplanung und Durchführung sowie bei der Erarbeitung der Projektdokumente.
- Auf wöchentlichen Treffen innerhalb der Projektgruppe werden Erfahrungen ausgetauscht, Fragen geklärt und der Arbeitsstand festgehalten. Die Ergebnisse jedes Gruppentreffens werden in einem Ergebnisprotokoll festgehalten. Zur leichteren Koordination legt die Gruppe hierfür gemeinsam mit Auftraggeber und Berater einen festen regelmäßigen Termin fest.
- Jeder einzelne Teilnehmer führt ein Zeitprotokoll, in dem er erfasst, mit welcher projektbezogenen Beschäftigung er wieviel Zeit zugebracht hat.

### 4 Hinweise zu Projektdurchführung

Die Probleme bei der Durchführung eines umfangreichen Softwareentwicklungsprojektes können in einer Vielzahl unterschiedlicher Unwägbarkeiten liegen. Jede Gruppe hat eigene Probleme. Die Projektdurchführung muss so geplant werden, dass möglichst viele Unwägbarkeiten möglichst früh erkannt und behandelt werden können. Typische Unwägbarkeiten sind:

- Probleme bei der Koordination der Teilnehmer (Teamarbeit bzw. Aufgabenverteilung). *Lösungsmöglichkeit: langfristige Urlaubs- und sonstige Terminplanung, insbesondere kurz vor Ende der Projektlaufzeit. Eventuellen Verlängerungsbedarf einplanen. Festlegung eines wöchentlichen Besprechungstermins.*
- Missverständnisse der Problemstellung. *Lösungsmöglichkeit: Anfertigung eines detaillierten Pflichtenheftes und Diskussion darüber mit dem Auftraggeber. Systematische Suche nach Möglichkeiten für Missverständnisse. Frühe und häufige Rücksprachen über die Aufgabenstellung mit dem Auftraggeber. Frühe Anfertigung und Diskussion von Prototypen.*
- Nicht ausreichende Kenntnis der Programmierumgebung. *Lösungsmöglichkeit: Einplanung einer hinreichenden Einarbeitungszeit. Wahl einer vertrauten Entwicklungsumgebung und frühzeitiges Festlegen einer für die Gruppe einheitlichen Programmierumgebung. Hinzuziehung von Experten.*

- Bei Maschinenansteuerung (Sensoren/Aktuatoren): Ungenauigkeiten, Zeitverzug. *Lösungsmöglichkeit: Frühzeitige Anfertigung von Prototypen, um schnell ein Gefühl für Problempunkte zu erarbeiten. Systematische Untersuchung der Genauigkeiten der Sensoren und Aktuatoren.*
- Effizienzprobleme bei der Arbeit mit großen Datenmengen. *Lösungsmöglichkeit: Frühzeitige Arbeit mit realitätsnahen, d. h. in der Regel umfangreichen Datenmengen.*
- Gestaltung der Benutzerschnittstelle. *Lösungsmöglichkeit: Frühzeitige Rückmeldung vom Auftraggeber anhand von Prototypen der Benutzerschnittstelle.*
- Aufwände für Tests, Fehlersuche, Dokumentation. *Lösungsmöglichkeit: Mehrfacher Durchlauf des Gesamtentwicklungsablaufs in Teilprojekten, und deren sorgfältige anschließende Auswertung, um in früheren Phasen Erfahrungen für die späteren zu sammeln.*
- Koordination bei knappen Ressourcen (Rechenzeit, Testumgebungen), dies insbesondere zu Projektende, wenn die Arbeitsintensität bei der eigenen und anderen Gruppen zunimmt. *Lösungsmöglichkeit: Gute Planung, um zum Ende der Projektlaufzeit nicht überproportional viel Ressourcen zu benötigen. Womöglich vorzeitige Fertigstellung des Projektes.*
- Konflikte bei gemeinsamer Arbeit mehrerer Personen auf derselben Codebasis. *Lösungsmöglichkeit: Nutzung eines Konfigurations- und Versionsmanagementsystems, beispielsweise CVS (Concurrent Versions System; frei benutzbar, im Internet erhältlich, Einsatz auf verschiedenen Plattformen möglich). Codierstandard frühzeitig festlegen.*

Solche Unwägbarkeiten können zu erheblicher Mehrarbeit führen, wenn sie nicht frühzeitig erkannt und berücksichtigt werden.

Daher wird ein zyklisches Vorgehen vorgegeben. Die Aufgabenbearbeitung ist in vier Teilprojekten (TP1-TP4) durchzuführen, so dass etwa alle drei bis vier Wochen ein Teilprojekt abgeschlossen wird. Hierzu wird die Gesamtlösung in mehrere Teilschritte so aufgeteilt, dass nicht erst am Ende der Projektlaufzeit in sich abgeschlossene Arbeitsergebnisse präsentiert werden, sondern bereits relativ früh. Wir empfehlen, die als schwierig erachteten Teilprobleme zunächst möglichst isoliert, in einem Klein-Prototyp oder in einer Anzahl von frühen Experimenten anzugehen. Diese Teilschritte können aufeinander aufbauen oder voneinander unabhängig sein.

Eine dreiwöchige Anfangsphase (TP1) wird genutzt, (1) um eine Anforderungsanalyse durchzuführen, das Ergebnis in einem Pflichtenheft festzuhalten und es mit dem Auftraggeber abzusprechen, (2) um einen Entwurf der zu erstellenden Software anzufertigen und (3) die in diesem Entwurf enthaltene Funktionalität in einem initialen Arbeitsplan auf die Teilprojekte aufzuteilen.

Für die anderen drei Teilprojekte (TP2,TP3,TP4) durchläuft das Projektteam alle Phasen des Softwareentwicklungsprozesses, von der Spezifikation bis zu Dokumentationsentwicklung, Abnahmetest und Reflexion. So lassen sich Erfahrungen und insbesondere erfasste Aufwandsdaten für die Planung späterer Phasen des Projektes nutzen.

Es wird nicht erwartet, dass dieses initiale Design und der initiale Plan in den folgenden Teilprojekten unverändert umgesetzt werden. Die Erfahrungen, die eine Gruppe bei der Durchführung eines Teilprojektes sammelt, werden gewöhnlich Design- und Planänderungen für die Folgeprojekte nach sich ziehen. Diese sind jeweils zu dokumentieren, so dass am Ende jedes Teilprojektes ein aktuelles Design der Gesamtsoftware mit einem angemessenen Detaillierungsgrad vorliegt.

## 5 Wahrnehmung wichtiger Termine

### • Plenumstermine

Nach einem festen Zeitplan treffen sich alle Praktikumsteilnehmer zu gemeinsamen Veranstaltungen, die die Form von Präsentationen oder Vorlesungen haben:

- Präsentation eines Statusberichtes: einige Gruppen stellen ihren Arbeitsstand anhand von Dokumenten und Produkten vor und berichten über ihre Erfahrungen und Probleme im Softwareentwicklungs-Prozess. Da an Präsentationsterminen Teilprojekte abgeschlossen sein sollten, kann man normalerweise zur Präsentation Material der Teilprojektberichte nutzen. Wenn zu einem Präsentationstermin das

zugehörige Teilprojekt noch nicht abgeschlossen wurde, dann ist nur der Arbeitsstand präsentierbar.

- Vorlesung: Zu relevanten Aufgaben der Projektbearbeitung werden Informationen gegeben.

In den Abschlussplena präsentiert jede Gruppe ihr Produkt sowie einen Überblick über den bisherigen Verlauf des Praktikums, mit Bezug auf die Reflexionspapiere. Zu den zu behandelnden Punkten gehören:

- bisheriger Gesamtaufwand und durchschnittlicher Aufwand pro Person
- aktueller Stand im aktuellen Plan
- besondere in der Vergangenheit aufgetretene Schwierigkeiten und Gründe für Verzögerungen gegenüber dem Plan sowie notwendige Planänderungen

- **Auftraggebertermine**

Die Gruppen vereinbaren mit dem Auftraggeber einen festen wöchentlichen Termin, der in der Anfangsphase auch wöchentlich, später nach Vereinbarung genutzt werden sollte.

Bei diesen Treffen werden die fachlichen Probleme besprochen, die das Produkt und seine Funktion betreffen. Dazu gehören u.a.:

- Pflichtenheft
- Gestaltung der Programmoberfläche
- Entwurfsskizzen, die die Funktionalität beschreiben

In den Wochen, in denen im Plenum Präsentationen stattfinden, führen die Gruppen ihre Teilprojekte beim Auftraggeber vor.

- **Beratertermine**

Die Gruppen vereinbaren mit ihrem Berater einen festen wöchentlichen Termin. Aus praktischen Gründen sollte überlegt werden, ob dieser Termin mit den Auftraggebertermin zusammen gelegt werden kann.

Bei diesen Treffen werden Aspekte und Dokumente des Softwareentwicklungsprozesses besprochen. Dazu gehören u.a.:

- Projektplanung, Entwurfsskizzen, Zeitprotokolle, Stilfibel, Reviews, Testen.

In den Wochen, in denen im Plenum Präsentationen stattfinden, sind die entsprechenden Dokumente beim Berater abzugeben.

## 6 Notwendige Bedingungen für den Scheinerwerb

Für den Scheinerwerb sind die in der folgenden Liste angegebenen Dokumente entsprechend der beendeten Phase vorzulegen, die am Ende in einem Gesamtprojektbericht gesammelt werden. Der Anhang beschreibt für jeden Dokumenttyp, was die verschiedenen Dokumente enthalten sollen.

Nach dem initialen Entwurf (am Ende von TP1) sind **sowohl dem Berater als auch dem Auftraggeber die folgenden Dokumente** vorzulegen, entweder in einem Ordner als Papierdokumente, oder als eine einzelne PDF- oder Postscript-Datei, die die Dokumente zusammenfasst:

- Aussagekräftige Zeitprotokolle jedes einzelnen Gruppenmitglieds für die erste Arbeitsphase.
- Die Ergebnisprotokolle der gemeinsamen Sitzungen.
- Ein Pflichtenheft, in dem die Interpretation der Aufgabenstellung durch die Gruppe festgehalten ist, und das mit dem Auftraggeber abgesprochen wurde.
- Eine Stilfibel für das Kodieren.
- Der Grobentwurf des Gesamtprojekts, d.h. die Architektur des Systems. Dieser Entwurf darf ziemlich abstrakt sein, er muss nur die Aufteilung der Arbeit auf TP2/TP3/TP4 ermöglichen.
- Die Grobplanung für das Gesamtprojekt, also die Aufteilung der Arbeit auf TP2, TP3, TP4 und Festlegung von Start- und Enddaten für die Teilprojekte. Die Teilprojekte sollten so geplant sein, dass sie zu den Präsentationsterminen weitgehend abgeschlossen sind.
- Ein Reflexionspapier des Gesamtteams und einzelne Reflexionspapiere jedes einzelnen Team-Mitglieds.
- Ein Feinentwurf für das nächste Teilprojekt.
- Eine Feinplanung für das nächste Teilprojekt, mit inhaltlicher Beschreibung, Ressourcenabschätzungen, Rollenzuteilungen, Angabe des geplanten Abschlussdatums sowie Datum der Planung.

Zu festgelegten Stichtagen sind sowohl dem Berater als auch dem Auftraggeber folgende Dokumente als Statusbericht vorzulegen, entweder als Papierdokument, oder als eine einzelne PDF- oder Postscript-Datei:

- Aussagekräftige Zeitprotokolle jedes einzelnen Gruppenmitglieds für das letzte Berichtsintervall.
- Die Ergebnisprotokolle der gemeinsamen Sitzungen des letzten Berichtsintervalls.
- Ein Reflexionspapier des Gesamtteams und Reflexionspapiere der einzelnen Gruppenmitglieder.. Insbesondere sind Konsequenzen für die künftige Arbeitsweise explizit zu machen.
- Ein Testplan und ein Testprotokoll über den Testverlauf.
- Getesteter, dokumentierter und der Stilfibel entsprechender Programmcode (elektronisch).

Zum Abschluss der Teilprojekte TP2 und TP3 sind vorzulegen:

- Ein Feinentwurf für das nächste Teilprojekt.
- Eine Feinplanung für den Ablauf des nächsten Teilprojektes, mit Ressourcenabschätzungen, und Rollenzuteilungen.

Nach Abschluss des Gesamtprojekts sind sowohl dem Auftraggeber als Papierendokument vorzulegen:

- Aussagekräftige Zeitprotokolle jedes einzelnen Gruppenmitglieds über die gesamte Projektlaufzeit.
- Die Ergebnisprotokolle der gemeinsamen Sitzungen, über die gesamte Projektlaufzeit.
- Wenigstens ein Protokoll über die Durchführung eines Reviews.
- Benutzerdokumentation mit Erläuterung des Programmaufrufs und der Benutzung des Programms.
- Einen Gesamtprojektbericht zum Projektschluss, in dem alle Dokumente (auch die der Anfangsphase und der Teilprojekte) übersichtlich gesammelt sind.

Weitere notwendige Bedingungen für den Scheinerwerb sind:

- Kontinuierliche Mitarbeit an den Teilprojekten über die gesamte Projektlaufzeit
- Regelmäßige Teilnahme an den Plenums- und den Beraterterminen; wir werden Anwesenheitslisten führen, um Ihre Teilnahme zu prüfen
- Zwei Präsentationen von Arbeitsergebnissen im Plenum: (1) Ein Zwischenstand der Arbeit wird nach TP1, TP2 oder TP3 präsentiert werden. (2) Am Ende der Vorlesungszeit wird der erreichte Endstand präsentiert.
- Ein wesentlicher Beitrag zum Quellcodeergebnis muss geleistet werden.
- Eingehende Kenntnis nicht nur der selbst angefertigten Teile des Programms und anderer Arbeitsergebnisse, sondern auch der Teile, die von anderen Gruppenmitgliedern angefertigt wurden, und der Gesamtarchitektur.

Wir werden die von Ihnen erstellten Unterlagen in Forschung und Lehre einsetzen, um sie etwa als Beispiele für Softwarepraktikumsberichte an nachfolgende Generationen von Softwarepraktikumsteilnehmern auszuleihen, oder um sie in Studien- oder Diplomarbeiten auswerten zu lassen; berücksichtigen Sie das bei der Anfertigung.

## **7 Rollenzuteilungen – Koordination der Gruppenarbeit**

In jedem Softwareentwicklungsprojekt sind einige übergreifende Aufgaben zu erfüllen, die während der gesamten Projektlaufzeit bedeutsam sein können. Ein typisches Beispiel ist die Projektverfolgung. Aufgrund des übergreifenden Charakters können diese Aufgaben nicht in einzelnen Aufgabenpaketen gesammelt werden, und können daher im Rahmen der Aufgabenpaketplanung nicht einzelnen Verantwortlichen zugewiesen werden.

In manchen Projekten werden diese Aufgaben von einem oder mehreren Gruppenmitgliedern erledigt, ohne dass diesen diese Aufgaben explizit zugewiesen worden wären. In anderen fallen diese Aufgaben einfach unter den Tisch, weil sich niemand dafür verantwortlich fühlt, was zu Problemen führt.

In unserem Softwarepraktikum wollen wir dies Problem lösen, indem (1) eine Anzahl von Rollen explizit definiert wird, denen inhaltlich zusammenhängende übergreifende Aufgaben zugeordnet sind, und indem (2) in jedem Teilprojekt jede dieser Rollen einem Gruppenmitglied zugewiesen wird. Da die mit den Rollen verbundenen Aufgaben sowieso erledigt werden müssen, liegt der einzige Mehraufwand unseres Verfahrens darin, eine Verantwortungszuweisung explizit festzulegen und diese schriftlich im Teilprojektplan festzuhalten. Es dürfen mehrere Rollen demselben Gruppenmitglied zugewiesen werden, solange dieses dadurch nicht überfordert wird. Die Rollenzuweisungen können über die Projektlaufzeit hinweg wechseln oder gleich bleiben. Es sind die in der folgenden Liste angegebenen Rollen explizit zu vergeben:

- Der **Besprechungskordinator** achtet darauf, dass Besprechungstermine festgelegt und eingehalten werden, dass die Termine effizient genutzt werden, dass die Besprechungsergebnisse schriftlich protokolliert, nicht zu spät nach der Besprechung an alle Teilnehmer verteilt, von diesen geprüft und gegebenenfalls korrigiert werden, und dass auch der Berater ein Besprechungsprotokoll erhält.
- Der **Planungsmanager und Projektverfolger** achtet darauf, dass Pläne angefertigt werden (der Planungsmanager des ersten Teilprojekts muss auch dafür sorgen, dass ein grober Plan für das Gesamtprojekt angefertigt wird), dass die Pläne angemessen dokumentiert werden, dass hinreichend kleine Arbeitspakete mit klaren Abschlusskriterien und Verantwortungszuordnungen definiert werden, und dass der Abschluss solcher Arbeitspakete für die Verfolgung des Projektfortschritts genutzt wird. Auf die wöchentlichen Sitzungen bereitet sich der Planungsmanager und Projektverfolger vor, indem er zuvor von den anderen Gruppenmitgliedern den aktuellen Projektstand erfragt, um diesen dann in der Sitzung zu präsentieren.  
Die Pläne sollten von der gesamten Gruppe aufgestellt und bis zum Konsens diskutiert werden. Die Rolle des Planungsmanager besteht nur darin, zu sichern, dass dies auch wirklich stattfindet und dass die Pläne angemessen dokumentiert werden.
- Der **Anforderungsmanager** sorgt dafür, dass Missverständnisse, Mehrdeutigkeiten, Widersprüche und andere Unklarheiten hinsichtlich der Aufgabenstellung nicht implizit bleiben. Er sorgt dafür, dass ein Pflichtenheft angefertigt und gepflegt wird, dass es also bei zusätzlichen Absprachen und Klärungen mit dem Auftraggeber auf den neuesten Stand gebracht wird.
- Der **Entwurfsmanager** achtet darauf, dass die Architektur der entstehenden Software und die verwendeten Algorithmen in angemessener Form dokumentiert und in der gesamten Gruppe bis zum Konsens diskutiert werden, und dass spätere Entscheidungen über Entwurfsänderungen in der Entwurfsdokumentation nachgezogen werden.
- Der **Codemanager** achtet darauf, dass der Quellcode immer in einem geordneten Zustand ist. Dies bedeutet erstens, dass die Stilfibel eingehalten wird, und zweitens, dass die verschiedenen Versionen, die von verschiedenen Dateien existieren, systematisch verwaltet werden. Wir empfehlen den Einsatz von CVS (Concurrent Versions System).
- Der **Qualitätsmanager** achtet darauf, dass Qualitätssicherungen verschiedener Art systematisch durchgeführt werden, also etwa:
  - dass Design-Checklisten vorbereitet und Design-Reviews durchgeführt und dokumentiert werden,
  - dass Code-Checklisten vorbereitet und Code-Reviews durchgeführt und dokumentiert werden,
  - dass Testpläne aufgestellt und die Tests durchgeführt und dokumentiert werden,
  - dass anzufertigende Prosa-Dokumente auf geforderte Struktur, Verständlichkeit, Orthographie etc. geprüft werden.
- Der **Reflexionsmanager** sorgt dafür, dass für jedes Teilprojekt, und beim letzten Teilprojekt auch für das Gesamtprojekt, ein aussagefähiger Reflexionsbericht entsteht. Er achtet darauf, dass während des Teilprojektablaufs von allen Gruppenmitgliedern die Daten, die für den Reflexionsbericht nötig sind, gesammelt werden -- insbesondere sind dies Daten über die Aufwände der einzelnen Gruppenmitglieder für verschiedene Tätigkeiten in verschiedenen Teilaufgaben, die Abschlussdaten für die Aufgabenpakete des Teilprojektes, und über Aufgabenpakete, die in das Teilprojekt gehören, aber die bei der Planung vergessen worden waren. Er achtet darauf, dass bei der Erstellung der Teilprojektberichte und des Gesamtprojektberichtes die Ergebnisprotokolle der Besprechungen ausgewertet werden. Insbesondere achtet der Reflexionsmanager darauf, dass aus den dokumentierten und interpretierten Erfahrungen auch Konsequenzen gezogen und schriftlich festgehalten werden.  
Der Reflexionsmanager des letzten Teilprojektes ist dafür zuständig, dass pünktlich zum Projektschluss der vollständige Gesamtprojektbericht mit der vorgegebenen Struktur vorliegt.

Der jeweilige Rollenträger muss die zugehörigen Tätigkeiten nicht selbst ausführen; die erfolgreiche Bearbeitung des Softwarepraktikums ist schließlich eine Aufgabe der gesamten Gruppe. Er muss nur darauf achten, dass die Tätigkeiten im Eifer des Projektgeschäftes nicht zu kurz kommen. Er könnte etwa darauf achten,

- dass das Thema in Plänen angemessen berücksichtigt wird;
- dass das Thema zum gegebenen Zeitpunkt in Gruppensitzungen hinreichend detailliert besprochen wird;
- dass in den Sitzungen die notwendigen Entscheidungen getroffen werden;
- dass notwendige Aufgaben verschiedenen Gruppenmitgliedern zur Erledigung zugewiesen werden;
- und dass verfolgt wird, ob die zugewiesenen Aufgaben von den verschiedenen Gruppenmitgliedern auch erfüllt werden.

## **8 Aufgaben der Berater**

Die Berater beraten die Arbeitsgruppen bezüglich ihrer Arbeitsorganisation und bezüglich der Interpretation der Kriterien für den Scheinerwerb. Ihnen gegenüber haben die Gruppen regelmäßig Rechenschaft über ihren Arbeitsstand abzulegen; normalerweise beurteilen sie die abgelieferten Unterlagen dahingehend, ob sie für einen Scheinerwerb hinreichend sind; in Streitfällen wird die Praktikumsleitung zuständig. Die Berater helfen den beratenen Gruppen insbesondere dabei, aus ihren Erfahrungen zu lernen. Dafür müssen die beratenen Gruppen diese Erfahrungen nachvollziehbar dokumentieren und diese Dokumente den Beratern zugänglich machen; hierzu dienen insbesondere die verschiedenen Pläne, die Zeitprotokolle und die Reflexionspapiere. Zu den Stichtagen erhalten die Berater daher die Statusberichte und geben den Gruppen darüber ein Feedback, welche Elemente von hinreichender Qualität waren und welche nicht.

## **9 Zum Text**

In diesen Vorschlag zur Gestaltung des Softwarepraktikums gingen Erfahrungen mit der Organisation und der Durchführung mehreren Jahre ein. Wir danken studentischen Teilnehmern, Mit-Beratern und Mitarbeitern am Lehrstuhl für Software-Systemtechnik, die uns mit Verbesserungshinweisen weitergeholfen haben. Besonders danken wir Claus Lewerentz und Thomas Mohaupt für Diskussionen und konstruktive Vorschläge.



# Anhang: Zur Gestaltung der anzufertigenden Dokumente

## 1 Allgemeine Hinweise

- Es ist mit dem Auftraggeber abzusprechen, ob die verlangten Dokumente in elektronischer Form oder ausgedruckt abzugeben sind. Bei der elektronischen Form ist zweckmäßigerweise ein allgemein anerkanntes, portables Dokumentenformat zu wählen (PDF oder PS).
- Alle erstellten Dokumente sollten von der Gruppe in einem Ordner o. ä. gesammelt werden.
- Jedes Dokument ist mit einer Kopfzeile zu versehen, die über Gruppe, Aufgabe und Datum der Erstellung Auskunft gibt.

## 2 Sitzungsprotokolle

Die Ergebnisse aller Gruppenbesprechungen, ob sie mit oder ohne Berater durchgeführt werden, werden in Form von Ergebnisprotokollen festgehalten. Ein solches Protokoll enthält wenigstens die folgenden Punkte:

- Datum, Anfangszeitpunkt, Endzeitpunkt, Anwesende, Name des Protokollführers
- Auflistung der erörterten Themen. Zu jedem Thema: Entscheidungen über zu ergreifende Maßnahmen, und Zuordnung von Verantwortlichkeiten für die Durchführung von Maßnahmen an einzelne Personen.

Die Ergebnisprotokolle halten also Besprechungsergebnisse fest. Sie sind baldmöglichst nach der jeweiligen Besprechung anzufertigen und, beispielsweise per eMail, an die Besprechungsteilnehmer und den Berater zu verteilen, damit diese darauf reagieren können, etwa, wenn sie Missverständnisse oder Lücken feststellen.

An jedem Teilprojektende sind die Ergebnisprotokolle ein wichtiges Hilfsmittel bei der Erstellung des Teilprojektberichtes, und am Ende des Gesamtprojektes helfen sie noch einmal bei der Erstellung des Gesamtprojektberichtes.

## 3 Zeitprotokolle

Jeder einzelne Projektteilnehmer führt ab dem ersten Tag des Praktikums ein persönliches Zeitprotokoll über die für das Softwarepraktikum aufgewendeten Zeit. Am günstigsten fertigen Sie sich für solche Protokolle ein Formular an, das die folgenden Einträge vorsieht:

- Name der Person, Anfangsdatum und Name des Teilprojektes.
- Eine Tabelle, in der jede Tätigkeit in einer Zeile dokumentiert wird. Die Tabelle enthält folgende Spalten:
  - Datum der Tätigkeit
  - Minutengenau: Zeitpunkte von Beginn und Ende der Tätigkeit
  - Unterbrechungsdauer in Minuten
  - Dauer der Tätigkeit, abzüglich der Unterbrechungen
  - Kurzbeschreibung des Arbeitspaketes (und evtl der Teilaufgabe darin); hieraus muss auch der spezifische Tätigkeitstyp hervorgehen, mit der die Zeit zugebracht wurde (wenigstens sind abzudecken: Besprechung, Entwurf, Implementierung, Test, Review, Debugging, Dokumentation, Reflexion). Bei der Auswertung werden Sie entsprechende Statistiken aufstellen, um sich Ihre eigene Arbeitsweise klarer zu machen. Mit zu erfassen sind ebenfalls Zeiten für Literaturstudium und Internetrecherchen.
- Ein einheitliches Format für alle Gruppenmitglieder erleichtert die Auswertung.

Aufgrund der so erhobenen Zeiten können folgende Informationen abgeleitet werden:

- Die für das Softwareprojekt aufgewendete Zeit, etwa pro Woche. Dies ist die Grundlage für genauere Abschätzungen der für das Softwareprojekt wöchentlich aufwendbaren Zeiten in späteren Teilprojekten.
- Die für eine Teilaufgabe in einem Arbeitspaket insgesamt aufgewendete Zeit, und der für die einzelnen Arbeitspakete prozentual aufgewendete Anteil der Arbeitszeit am Teilprojekt. Dies ist die Grundlage eines Vergleichs der für die Arbeitspakete des Teilprojektes geplanten Aufwände mit den tatsächlichen Aufwänden, und die Grundlage für bessere Planungen für die folgenden Teilprojekte.

- Die für verschiedene Tätigkeitstypen (etwa wie oben: Besprechung, Entwurf, Implementierung, Test, Review, Debugging, Dokumentation, Reflexion etc.) absolut und prozentual aufgewendete Zeit in verschiedenen Teilprojekten und im Gesamtprojekt.

Diese Auswertungen sind für die Teil-Projektberichte anzuferigen.

## 4 Pflichtenheft

Das Pflichtenheft enthält eine Zusammenfassung aller fachlichen Anforderungen, die das zu entwickelnde Softwareprodukt aus der Sicht des Auftraggebers erfüllen muss. Dabei geht es um den Funktions-, Leistungs- und Qualitätsumfang des Produktes, konzentriert auf das WAS, nicht das WIE. Das Pflichtenheft muss so abgefasst sein, dass es als Basis eines juristischen Vertrages dienen kann. Das Pflichtenheft stellt also die vertragliche Beschreibung des Lieferumfanges dar. Anhand des Pflichtenheftes soll das fertige Produkt abgenommen werden können. Die beschriebenen Anforderungen sollen realisierbar sein. Entwurfs- und Implementierungsentscheidungen sollen nicht vorweggenommen oder unnötig eingeschränkt werden. Adressaten des Pflichtenheftes sind der Auftraggeber und der Auftragnehmer, repräsentiert durch den Projektleiter und die Systemanalytiker, Entwerfer, Qualitätskontrolle, Benutzerrepräsentant oder ausgewählte potentielle Benutzer. Der Umfang des Pflichtenheftes kann je nach Projektart durchaus mehrere 100 Seiten haben. Das Pflichtenheft kann als Grundlage des Vertrags zwischen Auftraggeber und Auftragnehmer verstanden werden. Bestandteile des Pflichtenhefts sind gewöhnlich:

### 1. Zielbestimmung

- 1.1 Musskriterien
- 1.2 Wunschkriterien
- 1.3 Abgrenzungskriterien

In diesem Kapitel wird beschrieben, welche Ziele durch den Einsatz des Produktes erreicht werden sollen. Um den Entscheidungsraum für die Realisierung abzustecken und um die Wahl von Realisierungsalternativen zu erleichtern, erfolgt die Zielbestimmung durch die Festlegung von Muss-, Wunsch- und Abgrenzungskriterien. Unter Musskriterien wird aufgeführt, welche Leistungen für das Produkt unabdingbar sind, damit es für den vorgesehenen Einsatzzweck verwendet werden kann. Sie müssen auf jeden Fall erfüllt werden. Wunschkriterien beschreiben Wünsche an das zu entwickelnde Produkt, die nicht unabdingbar sind, deren Erfüllung aber so gut wie möglich angestrebt werden sollte. Abgrenzungskriterien sollen deutlich machen, welche Ziele mit dem Produkt bewusst nicht erreicht werden sollen. Da die Wünsche an ein Produkt im allgemeinen sehr umfangreich und oft leicht formulierbar sind, soll dieser Abschnitt dazu dienen, Abgrenzungen des Produktes zu definieren.

### 2. Produkt-Einsatz

- 2.1 Anwendungsbereiche
- 2.2 Zielgruppen
- 2.3 Betriebsbedingungen

Da der geplante Produkteinsatz wesentliche Auswirkungen auf die funktionale Mächtigkeit und auf die Qualitätsmerkmale hat, werden in diesem Abschnitt die Anwendungsbereiche, z.B. Textverarbeitung im Büro, und die Zielgruppen, z.B. Sekretärinnen, Schreibkräfte, definiert. Unter Umständen sollte auch festgelegt werden, von welchen Voraussetzungen, z.B. bezüglich des Qualifikationsniveaus des Benutzers, ausgegangen wird. Ebenfalls kann es sinnvoll sein, explizit anzugeben, für welche Anwendungsbereiche und Zielgruppen das Produkt nicht vorgesehen ist, z.B. für den DV-unkundigen Benutzer. Deckt das Produkt verschiedene Anwendungsbereiche und Zielgruppen ab, dann ist eine Aufzählung der unterschiedlichen Bedürfnisse und Anforderungen nötig. Unter Betriebsbedingungen werden folgende Punkte beschrieben:

- physikalische Umgebung des Systems,
- tägliche Betriebszeit,
- ständige Beobachtung des Systems durch Bediener oder unbeaufsichtigter Betrieb.

### 3. Produkt-Umgebung

- 3.1 Software
- 3.2 Hardware
- 3.3 Orgware
- 3.4 Produkt-Schnittstellen

In diesem Kapitel wird die Umgebung des Produktes beschrieben. Unter Software wird angegeben, welche Software-Systeme (Betriebssystem, Laufzeitsystem, Datenbank, Fenstersystem usw.) auf der Zielmaschine (Maschine, auf der das fertiggestellte Produkt eingesetzt werden soll) zur Verfügung stehen. Unter Hardware wird aufgeführt, welche Hardware-Komponenten (CPU, Peripherie, z.B. Grafikkarte, Drucker) in minimaler und maximaler Konfiguration für den Produkteinsatz vorgesehen sind. Unter Orgware wird aufgeführt, unter welchen organisatorischen Randbedingungen bzw. Voraussetzungen das Produkt eingesetzt

werden soll (z.B. »Elektronische Post ist nur dann sinnvoll einsetzbar, wenn die wichtigsten Empfänger organisatorisch und technisch in das elektronische Postsystem eingegliedert sind, d.h. ein LAN-Anschluss ist erforderlich«). Unter Produkt-Schnittstellen wird das Produkt in eine bestehende oder geplante Produkt-Familie eingeordnet oder die geforderten bzw. genutzten Schnittstellen zu anderen Produkten werden definiert bzw. vereinbart (z.B. Schnittstelle zum Ferndiagnosesystem). Außerdem kann auf andere Produkte verwiesen werden, die denselben Anwendungsbereich abdecken oder dieselbe Zielgruppe ansprechen.

#### **4. Produkt-Funktionen**

4.1 Funktion1

4.2 Funktion2 usw.

Unter Produkt-Funktionen erfolgt die funktionale Beschreibung des Produktes aus Benutzersicht. Dieses Kapitel sollte in so viele Abschnitte gegliedert werden, wie das Produkt Funktionen oder Funktionsbereiche aufweist. Eine Funktion kann auch durch Aufgliederung in Unterabschnitte weiter verfeinert werden. Es sollte hier besonders darauf geachtet werden, dass nicht das WIE, sondern ausschließlich das WAS definiert wird. Innerhalb jeder Funktion sollen Einzelanforderungen in verbaler Form beschrieben werden. Jede Einzelanforderung ist durch eine vorangesetzte Zahl mit vorangeseztem F, eingeschlossen in Schrägstriche, zu markieren (z.B. /F10/), um eindeutig referenzieren zu können. Die Funktionen sollen unabhängig von einem bestimmten Bildschirm-Layout oder einer bestimmten Tastenbelegung beschrieben werden. Diese Festlegungen erfolgen erst im "Benutzungsschnittstelle." Bei Produkten, die keine Benutzungsoberfläche besitzen, werden hier analog die Funktionen beschrieben, die das anwendende System benötigt.

#### **5. Produkt-Daten**

5.1 Daten1

5.2 Daten2

Beschreibung der langfristig zu speichernden Daten aus Benutzersicht.

Referenzierung: /D10/ usw.

#### **6. Produkt-Leistungen**

Unter Produkt-Leistungen werden alle Anforderungen aufgeführt, die zeitbezogen oder umfangsbezogen sind, z.B. maximale Dialogantwortzeiten bei speziellen Funktionen, maximaler Datenumfang bzw. Datendurchsatz (Durchschnittswerte und Spitzenbelastung), Genauigkeit bei numerischen Daten usw. Die einzelnen Leistungsanforderungen werden analog wie die Funktionsanforderungen nummeriert, allerdings mit dem vorangestellten Buchstaben L (z.B./L30/).

#### **7. Benutzungsoberfläche**

In diesem Kapitel werden grundlegende Anforderungen an die Benutzungsoberfläche festgelegt. In Abhängigkeit vom Produkt sollten folgende Gesichtspunkte berücksichtigt bzw. festgelegt werden:

- Bildschirmlayout,
- Drucklayout,
- Tastaturbelegung,
- Dialogstruktur usw.

Die Festlegungen sollten sich auf die produktspezifischen Ausprägungen beschränken, die nicht durch das Qualitätsmerkmal Benutzungsfreundlichkeit im Kapitel 8 Qualitäts-Zielbestimmung abgedeckt wird. Die einzelnen Anforderungen werden analog wie die Funktionsanforderungen nummeriert, allerdings mit dem vorangesezten Buchstaben B. Bei Produkten, die keine Benutzungsoberfläche besitzen, werden hier analog die Schnittstellenkonventionen beschrieben, die für das anwendende System wichtig sind.

#### **8. Qualitäts-Zielbestimmung**

In diesem Kapitel wird festgelegt, welche Qualitäts-Merkmale das zu entwickelnde Produkt in welcher Qualitätsstufe besitzen soll. Voraussetzung für die Qualitäts-Zielbestimmung ist, dass die Qualitäts-Merkmale in operationalisierter Form vorliegen. Die operationalisierten Qualitäts-Merkmale sind als Anhang dem Pflichtenheft beizufügen, wenn sie nicht als allgemeine Richtlinie (Standard, Werknorm) zur Verfügung stehen.

#### **9. Globale Testszenarien/Testfälle**

9.1 Testfall 1

9.2 Testfall 2 usw.

In diesem Kapitel werden anwendungsbezogene Testfälle zusammengestellt, die im allgemeinen mehrere Produkt-Funktionen in Anspruch nehmen. Während die Testfälle pro Funktion aus den Funktionsanforderungen abgeleitet werden, sollten in diesem Kapitel globale Testfälle aufgeführt werden. Diese Testfälle sind dann für den Abnahmetest zu verwenden.

## 10. Entwicklungs-Umgebung

10.1 Software

10.2 Hardware

10.2 Orgware

10.3 Entwicklungs-Schnittstellen

In diesem Kapitel wird die Entwicklungsumgebung des Produktes beschrieben. Es wird festgelegt, welche Konfiguration bzgl. Software, Hardware und Orgware für die Entwicklung des Produktes benötigt wird. Diese Festlegungen sind insbesondere dann notwendig, wenn Entwicklungs- und Zielmaschine unterschiedlich sind. Bei Entwicklungs-Schnittstellen ist unter Umständen aufzuführen, über welche einzuhaltenden Hardware- und Software-Schnittstellen Entwicklungs- und Zielrechner gekoppelt sind. Unter Software ist insbesondere aufzuführen, welche Software-Werkzeuge, z.B. Compiler usw., benötigt werden.

## 11. Ergänzungen

In diesem Kapitel werden Ergänzungen oder spezielle Anforderungen beschrieben, die über die aufgeführten Kapitel 1 bis 10 hinausgehen.

Beispielsweise können hier Installationsbedingungen festgelegt werden wie:

- Bauliche und räumliche Voraussetzungen,
- Bereitstellung von Testdaten,
- Bereitstellung von Hilfspersonal.

Außerdem können hier zu berücksichtigende Normen, Vorschriften, Patente und Lizenzen aufgeführt werden.

Oft ist es sinnvoll, alle im Pflichtenheft verwendeten Fachbegriffe zu definieren. Damit soll sichergestellt werden, dass eine einheitliche Terminologie auch in den späteren Phasen verwendet wird. Außerdem sollen dadurch Missverständnisse von vornherein vermieden werden. Ein solches Glossar oder Begriffslexikon kann in diesem Kapitel oder im Anhang angelegt werden.

Das Pflichtenheft dient dazu, Missverständnisse bezüglich der im Produkt umzusetzenden Funktionalität weitmöglichst zu vermeiden. Daher muss die Gruppe ihre Interpretation der Aufgabenstellung so detailliert beschreiben, dass der Auftraggeber Fehlinterpretation auch erkennen kann. Das Pflichtenheft darf Verweise auf die Aufgabenstellung enthalten oder diese zitieren.

## 5 Arbeitspläne

Arbeitspläne (Grobplan für das Gesamtprojekt, Feinpläne für die Teilprojekte TP2 bis TP4) sind so aufzustellen, dass in früheren Arbeitsphasen Erfahrungen gesammelt werden können, die in den späteren nutzbar sind. Dafür wird das Gesamtprojekt in drei Teilprojekte aufgeteilt, die nacheinander bearbeitet werden, und bei denen die für am problematischsten gehaltenen Teilaufgaben möglichst früh bearbeitet werden, damit Risiken früh im Projektverlauf erkannt werden. Daher wird insbesondere verlangt, in jedem Teilprojekt alle typischen Entwicklungstätigkeiten einzuplanen, also alle Schritte von der möglichst exakten Spezifikation der umzusetzenden Funktionalität bis zum abschließenden systematischen Gesamttest für das Teilprojekt und die Dokumentation.

Die Planungen für weit in der Zukunft befindliche Teilprojekte werden nur grob sein. Die Planungen für das nächste durchzuführende Teilprojekt aber sollten so detailliert wie möglich sein, damit das Planungsdokument als Grundlage für eine kritische Diskussion des Plans dienen kann, und damit erkennbar werden kann, ob wichtige Teilaufgaben vergessen wurden. Wir schlagen die Nutzung von drei Granularitätsebenen vor:

- Die oberste Ebene bildet das Gesamtprojekt.
- Das Gesamtprojekt wird in drei Teilprojekte ungefähr gleichen Umfangs aufgeteilt.
- Jedes Teilprojekt besteht aus fünf bis zwanzig Arbeitspaketen.

### Grobplan

Der Grobplan ist ein Ergebnis der Vorbereitungsphase TP1. Er beschreibt, wie die Gesamtarbeit, die während des Projektes zu leisten ist, auf die Teilprojekte TP2 bis TP4 aufgeteilt wird. Für jedes Teilprojekt wird beschrieben, wann es genau beginnt, wann es endet, und es wird detailliert beschrieben, welche Funktionalität des Endproduktes während des Teilprojektes implementiert werden soll. Die Teilprojekte sind so zu planen, dass sie in etwa zu den Präsentationsterminen abgeschlossen sind.

### Feinplan

Die Dokumentation der Planungsergebnisse der Feinplanung für ein Teilprojekt sollte dabei wenigstens folgende Elemente enthalten:

- Name des Teilprojektes, Datum der Planung

- Die Definition einer Anzahl von in sich abgeschlossenen Arbeitspaketen. Zu jedem Arbeitspaket gehören:
  - eine Kenn-Nummer des Arbeitspakets, um in später erstellten Texten Bezüge darauf zu vereinfachen;
  - eine Kurzbeschreibung des Arbeitspakets, mit Überschrift und Erläuterung, in der auch evtl. Teilaufgaben differenziert werden, etwa: Implementierung und zugehörige Qualitätssicherung;
  - eine Aufwandsabschätzung für das Paket (minutengenau, insgesamt, nicht pro Person),
  - Verantwortlichkeitszuordnung(en) für das Paket; und
  - das geplante Anfangs- und das Abschlussdatum des Arbeitspakets.

Arbeitspakete müssen so geplant werden, dass die Tatsache ihrer erfolgten Bearbeitung für den Projektfortschritt aussagekräftig ist; es muss also aussagekräftige Abschlusskriterien für die Arbeitspakete geben. Nur dann kann das Ende eines Arbeitspaketes Ihnen als Meilenstein dienen, der während der Laufzeit des Teilprojektes hilft, festzustellen, ob Sie im Plan liegen.

Ein Beispiel: In einem Arbeitspaket, in dem eine Teilaufgabe „Implementierung“ enthalten ist, sollte auch eine Teilaufgabe „Qualitätssicherung des Implementierungsergebnisses“ vorkommen, etwa als Review oder Test, und zwar einschließlich anschließender Überarbeitung und Iteration bis zur Erreichung der gewünschten Qualität. Anderenfalls würden Sie etwa den Abschluss einer äußerst mangelhaften Implementierung als Erreichung eines Meilensteins zählen, was aber für den Projektfortschritt nicht aussagekräftig ist.

- Neben Qualitätssicherungsmaßnahmen, die als Teilaufgaben zu jedem Arbeitspaket gehören, in welchem Implementierung stattfindet, sollte es für den Gesamttest der in einem Teilprojekt angefertigten Programme ein eigenes Arbeitspaket zur Qualitätssicherung geben.
- Die Teilprojektberichtserstellung und die Planung für das folgende Teilprojekt sind als eigene Arbeitspakete zu definieren.
- Die Definition von Abhängigkeiten zwischen den Arbeitspaketen. Diese erfolgt am anschaulichsten in der Form eines gerichteten azyklischen Graphen, dessen Knoten Arbeitspakete darstellen, und in dem ein Weg von einem Arbeitspaket AP<sub>m</sub> zu einem Arbeitspaket AP<sub>n</sub> anzeigt, dass AP<sub>m</sub> beendet sein muss, bevor AP<sub>n</sub> begonnen werden kann (Netzplan).
- Für jede einzelne Person: Eine Abschätzung der wöchentlich für das Projekt verfügbaren Zeit, in Stunden. Auf der Basis der Aufwandsschätzungen pro Projekt und dieser verfügbaren Zeit ist der Zeitpunkt zu bestimmen, zu dem ein Arbeitspaket beendet sein müsste.

Erfahrungsgemäß sind Planungen, die auf deutlich mehr als 10 Arbeitsstunden pro Woche und Person beruhen, sehr optimistisch.

Bei dieser Ressourcenplanung sind auch Feiertage, Geburtstage, Urlaube, Praktika, Klausurtermine mit ihren Vorbereitungszeiten und andere vorhersehbare Ereignisse einzuplanen.

Für jedes Arbeitspaket müssen Anfangs- und Endtermin und die Ressourcenplanung konsistent sein. Der Plan ist in einer Weise zu dokumentieren, die Inkonsistenzen leicht erkennbar macht, etwa als Gantt-Diagramm.

Die Arbeitspakete dienen während der Arbeit an einem Teilprojekt zur Projektverfolgung. Das Ende jedes Arbeitspaketes bildet einen Meilenstein. Da Verzögerungen daher erst dann klar erkannt werden, wenn Arbeitspakete abgeschlossen sein müssten, sollten Arbeitspakete nicht zu umfangreich sein. In jeder Arbeitswoche sollte die Fertigstellung einiger Arbeitspakete geplant sein, damit Sie in den wöchentlichen Besprechungen feststellen könne, ob Sie vorangekommen sind.

Gewöhnlich gilt: Pläne, die von Entwicklern aufgestellt werden, haben den doppelten Charakter einer möglichst realistischen Erwartung und zugleich einer Selbstverpflichtung. Aber: Als Teilnehmer des Softwarepraktikums haben Sie normalerweise noch keine Erfahrungen mit solchen Planungen. Daher können Ihnen Planungsfehler, die nicht auf Erfahrungen sondern allein auf Spekulation beruhen, nicht persönlich angerechnet werden. Da ohne Übung in diesem Bereich keine Erfahrung gesammelt werden kann, müssen Sie dennoch, basierend zunächst nur auf Spekulation, und in späteren Teilprojekten auf der Grundlage der dann vorhandenen Zeitprotokolle und spekulativer Annahmen, möglichst detaillierte Pläne für Teilprojekte anfertigen. Nach der Durchführung jedes Teilprojektes werden Sie für den Teilprojektbericht tatsächlich benötigte Aufwände und geplante vergleichen, die Unterschiede interpretieren und Konsequenzen für folgende Teilprojekte ziehen.

## 6 Entwurfsdokumente

Die Entwurfsdokumente beschreiben das zu implementierende (Teil-)Programm in problemangemessener Weise. Sowohl Statik als auch Dynamik der Software ist in verständlicher Form zu beschreiben. Hierzu können

etwa, soweit angemessen, ER-Diagramme, Objekt- und Klassendiagramme, Datenflussdiagramme, Interaktions- und/oder Zustandsdiagramme oder auch andere selbstgewählte, angemessene Notationen benutzt werden.

Entwurfsdokumente sind die wichtigste Grundlage für die Gruppendiskussion über die Strukturen der Software, die angefertigt werden soll. Als solche können sie natürlich nicht nach der Implementierung angefertigt werden. Ein Entwurfsdokument für das Gesamtprojekt wird in den ersten zwei Wochen der Projektlaufzeit angefertigt; dieses wird im Laufe der einzelnen Teilprojekte dann korrigiert und ergänzt.

## **7 Stilfibel**

Eine Stilfibel enthält Hinweise für die Gestaltung des Quellcodes oder auch der Entwurfsbeschreibungen. Eine Quellcode-Stilfibel legt unter anderem fest:

- Nach welchen Kriterien der Quellcode auf Dateien aufzuteilen ist und welche Namenskonventionen für Dateinamen zu nutzen sind.
- Welche Konventionen für Zeilenlängen, Prozedurlängen, Kommentierung, Bezeichnerwahl, Einrückung und etwa zu vermeidender Sprachkonstrukte der benutzten Programmiersprache einzuhalten sind.

Weitere Punkte können hinzukommen. Sie dürfen eine existierende Stilfibel an Ihre Bedürfnisse anpassen.

## **8 Testpläne und Testprotokolle**

Der Wert von Tests, die nicht nach einem vorher erdachten und dokumentierten Plan durchgeführt werden, ist gering, weil die Auswahl der ausgeführten Testfälle hier nur willkürlich erfolgt. Ein systematischer Test verlangt also, dass die auszuführenden Testfälle ausgewählt und dokumentiert werden, bevor die Tests durchgeführt werden. Bei der Ausführung der Tests wird dann für jeden Testfall des Testplans das Testergebnis protokolliert, damit die Testergebnisse nachvollziehbar werden.

Ein Testplan enthält eine Auflistung der durch die Testfälle abzudeckenden Fragestellungen und eine Anzahl von detailliert, d.h. für einen Außenstehenden nachvollziehbar beschriebenen Testfällen, deren Ausführung diese Fragestellungen in hinreichender Weise klären. Ein Testfall besteht aus einer genauen Beschreibung der Ausführungsumgebung des zu testenden Programms oder Programmteils sowie des erwarteten Programmverhaltens.

Ein Testprotokoll beschreibt Datum, Teilnehmer und Ergebnisse eines Tests, also das Systemverhalten, das bei den Testfällen eines Testplans aufgetreten ist.

Zum Abschluss jedes Teilprojektes sind Testpläne und Testprotokolle anzufertigen, die nachweisen, dass das Produkt den Anforderungen entspricht oder nicht. Bei Nichterfüllung der Anforderungen ist natürlich entsprechende Nacharbeit und eine Wiederholung des systematischen Tests ggf. im nächsten Teilprojekt nötig.

## **9 Reviews und Reviewprotokolle**

Reviews sind ein sehr leistungsfähiges Mittel der Qualitätssicherung, das Arbeitsergebnisse verschiedenster Art zu prüfen gestattet. Sie beruhen darauf, dass die zu prüfenden Arbeitsergebnisse einigen Gutachtern vorgelegt werden, die sie jeweils für sich gemäß den Punkten einer Checkliste prüfen, und die dann in einer gemeinsamen Sitzung ihre Ergebnisse abgleichen. Wir empfehlen, dass Sie bei Ihrem Review die Gutachter aus anderen Praktikumsgruppen rekrutieren und sich entsprechend auch selbst bereit erklären, in Reviews anderer Gruppen die Gutachterrolle zu übernehmen.

Die bei einem Review Mitwirkenden haben verschiedene Rollen. Diese sind:

- Autor: Dies ist die Person, die das zu prüfende Arbeitsergebnis angefertigt hat.
- Moderator: Dieser koordiniert die Verteilung von Arbeitsergebnis und Checklisten an die Gutachter, ordnet den Gutachtern Checklistenpunkte zu, legt einen Termin für die Review-Sitzung fest und moderiert den Ablauf.

- Gutachter: Diese untersuchen den Prüfling anhand der ihnen zugeteilten Punkte der Checkliste, halten die Urteile fest und vergleichen ihre Urteile später in der Review-Sitzung.
- Protokollführer: Dieser führt während der Review-Sitzung das Protokoll.

Die Rolle des Protokollführers kann vom Autor oder vom Moderator übernommen werden.

Der Ablauf eines Reviews ist folgender:

1. Ein zu prüfendes Arbeitsergebnis (der „Prüfling“) wird ausgewählt. Es sollte einen Umfang haben, so dass es in einer Review-Sitzung von höchstens zwei Stunden gründlich besprochen werden kann.
2. Checklisten werden ausgewählt oder angefertigt, die den Gutachtern dabei helfen sollen, keine wichtigen Punkte bei der Begutachtung zu vergessen.
3. Der Prüfling und die Checklisten werden an die Gutachter verteilt. Nicht jeder Checklistenpunkt muss von jedem Gutachter behandelt werden; aber *jeder Checklistenpunkt* sollte von *wenigstens zwei Gutachtern* behandelt werden.
4. In Vorbereitung auf die Review-Sitzung geht jeder Gutachter für sich die ihm zugeteilten Checklistenpunkte für den Prüfling durch und hält sein Urteil schriftlich fest. Er hält auch fest, was ihm über die ihm zugeteilten Punkte hinaus am Prüfling aufgefallen ist.
5. Nach einer Zeit, die es jedem Gutachter ermöglicht, seine Aufgabe gründlich zu erfüllen (wenigstens ein paar Tage), treffen sich der Autor des Prüflings, die Gutachter, ein Moderator und der Protokollführer zur maximal zweistündigen Review-Sitzung.

Der Moderator fragt alle Gutachter, ob sie dazu gekommen sind, den Prüfling zu untersuchen. Wenn dies für einen Gutachter nicht gilt, wird die Sitzung abgebrochen und vertagt.

Anderenfalls geht der Moderator die Punkte der Checkliste nacheinander durch und fragt die Gutachter nach Ihrer Einschätzung. Abschließend fragt der Moderator die Gutachter nach weiteren Einschätzungen des Prüflings. Zu jedem Punkt wird unter den Gutachtern nach einem Konsens darüber gesucht, ob ein Mangel vorliegt, ob Nacharbeit nötig ist, und ob das Review für diesen Prüfling nach der Nacharbeit wiederholt werden sollte. Das Ergebnis wird zu jedem Punkt im Review-Protokoll festgehalten.

Die Diskussion von Verbesserungsmöglichkeiten in dieser Sitzung führt erfahrungsgemäß zu ausufernden Sitzungen und ist daher zu unterlassen. Der Autor rechtfertigt seine Entwurfs- und Codier-Entscheidungen nicht, sondern beantwortet nur Informationsfragen von Seiten der anderen Sitzungsteilnehmer.

Die Diskussion von Verbesserungsmöglichkeiten findet allenfalls im Anschluss an die offizielle Review-Sitzung statt.

Ein Review-Protokoll enthält wenigstens folgende Informationen:

- Datum, Anfangszeitpunkt, Endzeitpunkt, Anwesende mit ihren Review-Rollen, Name des Protokollführers
- Kennzeichnung des Prüflings
- Einschätzung des Prüfobjektes durch die Reviewer
- Auflistung nötiger Nachbearbeitungen und eine Entscheidung, ob der Prüfling einem weiteren Review unterzogen werden muss
- Im Anhang: Die bearbeiteten Checklisten der Reviewer

## 10 Checklisten

Checklisten werden von den Gutachtern eines Reviews benutzt, um das Ergebnis einer individuellen Begutachtung festzuhalten. Die Punkte einer Checkliste sollten möglichst konkret und überprüfbar sein. Beispielsweise sollte in einer Code-Checkliste der Punkt „der Code ist verständlich“ nicht allein auftauchen, sondern in konkretere Unterpunkte wie „die Methoden sind nicht zu lang“, „die Bezeichner sind gut gewählt“ und noch ein paar andere aufgelöst werden.

Checklisten enthalten Felder für das Datum, den Namen des Gutachters, Angaben zum Prüfling. Der Hauptteil ist eine Liste mit den zu überprüfenden Punkten, in der vermerkt werden kann, ob der Punkt für den Prüfling zutrifft oder nicht.

Sie müssen die von Ihnen zu benutzenden Checklisten nicht neu erfinden. Sie können auch vorhandene Checklisten an Ihre Erfordernisse anpassen.

## 11 Gruppen- und Einzelreflexionspapiere

Reflexionspapiere gibt es in zwei Formen. Das Gruppenreflexionspapier stellt einen Konsens der Gruppe dar und legt konkrete Konsequenzen für Veränderungen der Arbeitsweise des Teams insgesamt fest. Die Einzelreflexionspapiere stellen persönliche Erfahrungen und konkrete Konsequenzen für die Veränderung der Arbeitsweise einzelner Personen dar.

Das **Gruppenreflexionspapier** enthält verschiedene Teile, für deren Erstellung die verschiedenen Rollenträger in der Gruppe verantwortlich sind:

- Der Besprechungskordinator sammelt die Besprechungsprotokolle des Teilprojektes; er reflektiert, wie gut Kooperation und Kommunikation während des Teilprojektes funktioniert haben, und er zieht konkrete Konsequenzen daraus für das nächste Teilprojekt.
- Der Planungsmanager und Projektverfolger reflektiert den Projektablauf hinsichtlich der Planerfüllung. Diese Reflexion enthält folgende Punkte:
  - Für jedes Arbeitspaket: eine Gegenüberstellung von Planung und wirklichem Ablauf, mit einer Interpretation der Unterschiede durch Bezug auf die einzelnen Teilaufgaben und den daraus für die Arbeit am nächsten Teilprojekt zu ziehenden Konsequenzen.
  - Für jeden Teilnehmer: eine Gegenüberstellung von Aufwandsplanung und wirklich erbrachtem Aufwand, wiederum mit einer Interpretation der Unterschiede und den daraus zu ziehenden Konsequenzen.
  - Eine absolute und eine prozentuale Aufstellung über angefallene Aufwände bezüglich verschiedener Tätigkeitstypen, wenigstens aufgeschlüsselt nach „Besprechungen“, „Entwurf“, „Implementierung“, „Test“, „Review“, „Debugging“, „Dokumentation“, „Reflexion“.
- Der Anforderungsmanager berichtet, inwiefern sich die Anforderungen weiter geklärt haben und liefert die aktuelle Version des Pflichtenheftes.
- Der Entwurfsmanager liefert den aktuellen Entwurf des Gesamtsystems und diskutiert die wichtigsten Entwurfsentscheidungen respektive die wichtigsten Änderungen gegenüber der früheren Version.
- Der Codemanager diskutiert den Zustand des Quellcodes, also Umfang, Strukturierung, Einhaltung des Coding-Standards und die Nutzung eines Versionsverwaltungssystems.
- Der Qualitätsmanager berichtet über die durchgeführten Qualitätssicherungsmaßnahmen und über deren Erfolg.
- Der Reflexionsmanager berichtet (ab TP2), inwiefern Erfahrungen aus früheren Teilprojekten in den Ablauf des nun zu besprechenden Teilprojektes eingeflossen sind.

In den **Einzelreflexionspapieren** wertet jeder einzelne Projektteilnehmer und jede einzelne Projektteilnehmerin persönliche Erfahrungen aus und beschreibt persönliche Konsequenzen. Es enthält wenigstens:

- persönliche Interpretation der objektiven erhobenen Daten (Arbeitsaufwände, aufgeschlüsselt nach Teilaufgaben und Tätigkeiten, auch im Vergleich zum Plan, wie im Gruppenreflexionspapier vom Planungsmanager beschrieben), mit Angabe konkreter Konsequenzen;
- eher subjektive Einschätzungen des Projektgeschehens bezüglich der Kooperation und Kommunikation in der Gruppe;
- Einschätzungen der Ausfüllung der zugewiesenen Projektrollen durch die eigene Person und durch alle anderen Projektteilnehmer

## 12 Gesamtprojektbericht

Der Projektbericht zum Projektschluss enthält alle während der Projektlaufzeit gesammelten Dokumente und eine Auswertung des Gesamtprojektes. Der Gesamtprojektbericht wird folgendermaßen strukturiert:

- Es gibt ein Inhaltsverzeichnis mit Seitenangaben für Kapitel und Unterkapitel.
- Ein erstes Kapitel gibt einen Überblick über das Projekt und bündelt Dokumente, die das Gesamtprojekt betreffen: (1) Pflichtenheft (wenn es existiert), (2) Stilfibel, (3) Review-Checklisten, (4) Nutzerdokumentation des Endproduktes



- Das zweite Kapitel bündelt die Ergebnisse der Phase des initialen Entwurfs, also (1) das initiale Entwurfsergebnis, (2) den initialen Plan für das Gesamtprojekt, (3) die Aufwände der Gruppenmitglieder in der Phase des initialen Entwurfs (aus Zeitprotokollen) und (4) Besprechungsprotokolle.
- Es gibt je ein Einzelkapitel für jedes einzelne Teilprojekt. Jedes dieser Einzelkapitel hat die folgenden Unterkapitel: (1) Ergebnisprotokolle der Gruppensitzungen; (2) anfängliche Aufwands- und Ressourcenplanung für das Teilprojekt; (3) Zeitprotokolle der Mitarbeiter; (4) Entwürfe und Dokumentation; (5) Testpläne und Testprotokolle; (6) ggf. Review-Protokolle; (7) Reflexionspapiere der Gesamtgruppe und der einzelnen Gruppenmitglieder
- Es gibt ein Abschlusskapitel, in dem der gesamte Projektablauf auf höherer Abstraktionsebene beschrieben und ausgewertet wird und in dem konkrete Konsequenzen für künftige Projekte gezogen werden, insbesondere bezüglich der Organisation der Gruppenarbeit. Hier erfolgen auch quantitative Auswertungen bezüglich des Gesamtprojektes, mit Vergleich von Plänen und tatsächlichen Aufwänden (1) pro Teilprojekt (nicht pro Arbeitspaket!) und (2) pro Person über die Gesamtprojektlaufzeit, wiederum inklusive Interpretation und Diskussion konkreter Konsequenzen für künftige Projekte.
- Der Quellcode wird auf einer Diskette oder CD beigelegt, und zwar in einer Form, so dass zur Lektüre keine spezifische Softwareentwicklungsumgebung nötig ist.

### 13 Literatur

- [Bal96] Helmut Balzert. *Lehrbuch der Software-Technik/Band 1*. Spektrum Akademischer Verlag, Heidelberg, 1996.
- [Bal98] Helmut Balzert. *Lehrbuch der Software-Technik/Band 2*. Spektrum Akademischer Verlag, Heidelberg, 1998.
- [FLS91] K. Frühauf, J. Ludewig und H. Sandmayr. *Software-Prüfung*. Verlag der Fachvereine/B.G.~Teubner, 1991.
- [Pre97] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, New York, 1997.
- [Som92] Ian Sommerville. *Software Engineering*. Addison-Wesley, Reading/Massachusetts, 1992.