

Technische Universität Dresden
Fakultät Informatik
Institut für Software- und Multimediatechnik
Lehrstuhl Softwaretechnologie
Dr.-Ing. Birgit Demuth

Einführung in das Softwaretechnologie-Praktikum SS 2000

1 Inhalt der Lehrveranstaltung

Das Softwaretechnologie-Praktikum im SS 2000 wird für Studenten des Diplomstudiengangs Informatik und des Ergänzungsstudiengangs Softwaretechnologie durchgeführt. Es hat zum Ziel, die im Fach Softwaretechnologie erworbenen Kenntnisse zu festigen und praktisch anzuwenden. Inhaltlich wird auf die im Wintersemester 99/00 durchgeführte Lehrveranstaltung Softwaretechnologie (Prof. Dr. H. Hußmann) Bezug genommen. D.h., es ist ein Softwareprodukt objektorientiert unter Beachtung folgender technischer Randbedingungen zu entwickeln:

- Anwendung der CRC-Karten-Methode
- Anwendung der UML-Notation
- Nutzung von Java 2 als Implementierungssprache
- Verwendung des Frameworks *SalesPoint 2.0*

Die Projektaufgaben sind im Rahmen studentischer Teams (in der Regel 6 Studenten/innen) arbeitsteilig zu lösen.

Die möglichst weitgehende Verwendung des Frameworks wird aus folgenden Gründen bindend vorgeschrieben.

- Durch das Framework ist eine Grundarchitektur vorgegeben. Das erleichtert den für Unerfahrene schwierigen OO Entwurf.

- Die Einarbeitung in bestehende Programmsysteme gehört zum Arbeitsalltag des Software-Entwicklers. Das Framework ist durch Beschreibung und Anpassungsschnittstelle vergleichsweise gut zugänglich.
- Es soll Wiederverwendung bewußt eingeübt werden. Hacker neigen dazu, laufend das Rad neu zu erfinden. Auf diese Weise entstünden unkontrolliert Code-Doubletten, die – weil nicht zusammenhängend – nicht systematisch gewartet (verbessert oder an neue Gegebenheiten angepaßt) werden können. Im Gegensatz dazu erfordert die Wiederverwendung von Bausteinen deren laufende Anpassung und führt dadurch zu immer flexibleren und verlässlicheren Komponenten: die Qualität steigt, die investierte Mühe lohnt sich mittelfristig.

2 Unterstützende Angebote

WWW-Seiten zum Softwarepraktikum. Aufgabenstellungen, Hinweise zum Praktikum sowie zu verfügbaren Tools, aktuelle Informationen, Lehrmaterialien und Verweise zu weiteren fachlichen Informationen sind unter der WWW-Adresse

<http://www-st.inf.tu-dresden.de/sp/>

zu finden.

Tutorial zum Framework. Am 10.04.00 findet in der 5. DS im Hörsaal GRU 172 ein Tutorial zur Einführung in das Framework SalesPoint statt (Herr Steffen Zschaler).

Zur Vorbereitung des Tutorials und zur Durchführung des Praktikums stehen Lehrmaterialien unter der WWW-Adresse

<http://www2.informatik.unibw-muenchen.de/Lectures/SalesPoint/Einstieg.htm>

zur Verfügung.

3 Teamorganisation

Es werden Projektteams gebildet (Gruppen von Studenten), die arbeitsteilig jeweils eine Aufgabenstellung zu bearbeiten und zu lösen haben. Die Organisation der Projektteams erfolgt nach dem Prinzip des **Chefprogrammiererteams**. Einem Studenten wird die Funktion des Chefprogrammierers übertragen, die anderen Gruppenmitglieder fungieren als Mitarbeiter. Der **Chefprogrammierer** übernimmt die Rolle des Teamleiters und ist in erster Linie für die Kommunikation mit dem Auftraggeber (Praktikumsbetreuer) und innerhalb des Teams, die Teamorganisation, für alle wichtigen Entwurfsentscheidungen sowie die qualitäts- und termingerechte Erstellung des Gesamtsystems (Softwareprodukt und Dokumentationen) verantwortlich. Er entwirft und implementiert zugleich zentrale oder kritische Teile des Gesamtsystems. Die anderen Teammitglieder teilen sich in folgende Aufgaben bzw. Rollen:

- Der **Assistent** kann stellvertretend für den Chefprogrammierer alle Entscheidungen treffen und unterstützt den Chefprogrammierer beim Entwurf.
- Der **Sekretär** ist für die gesamte Verwaltungsarbeit zuständig, einschließlich der Verwaltung der Klassenbibliotheken sowie aller Entwicklungsdokumente.
- Die **Programmierer** sind über alle Projektphasen für das ihnen zugeordnete Teilsystem einschließlich der Entwicklungs- und Anwenderdokumentation des Teilsystems zuständig.

In Ergänzung zu diesem allgemeinen Verständnis wird von jedem Student verlangt, daß er unabhängig von seiner Rolle im Team einen Teil der Implementation eigenständig übernimmt. Im Ergebnis des Entwurfes, an dem ebenfalls alle Studenten beteiligt sein sollen, erfolgt die Festlegung, wer für die Implementation welcher Klassen bzw. Teilsysteme (Packages) verantwortlich ist.

Die konkrete Arbeitsteilung in der Gruppe (Teamorganisation) wird durch die Gruppe festgelegt und im Projektplan festgeschrieben.

4 Durchführung des Praktikums

4.1 Aufgabenstellungen und Praktikumsbetreuung

Im Praktikum sollen auf der Basis des vorhandenen *SalesPoint 2.0*-Frameworks Verkaufsanwendungen in Java erstellt werden.

Jeder Praktikumsgruppe wird eine konkrete Aufgabenstellung und ein Betreuer zugeordnet. Die Betreuer führen sowohl **Pflichtkonsultationen** als auch **wöchentliche Konsultationen** durch. Die Termine (Uhrzeit und Räume) werden in Absprache mit den Betreuern festgelegt. Darüberhinaus sollte regelmäßig ins WWW nach aktuellen Informationen geschaut werden.

4.2 Einschreibung in Projekte

Um einen zügigen Beginn des Praktikums zu gewährleisten, wird die Gruppeneinteilung und Aufgabenzuordnung durch den Lehrstuhl in der ersten Praktikumswoche festgelegt. Es besteht die Möglichkeit, sich bis **spätestens Mittwoch, dem 05.04.2000, im Sekretariat des Lehrstuhles Softwaretechnologie (DUE 26/258)** in Listen mit Gruppen und Aufgabenstellungen einzuschreiben. Jeder Gruppe werden 6 Studenten zugeordnet. Die endgültige Praktikumsgruppeneinteilung wird bis zum Freitag, den 07.04.00, durch Aushang vorm Sekretariat des Lehrstuhls bekanntgegeben.

4.3 Projektphasen

Der Praktikumsablauf wird in fünf Projektphasen einschließlich notwendiger Iterationen unterteilt. Die **projektbegleitende Dokumentation im WWW** ist ausdrücklich erwünscht!

4.3.1 Einarbeitung

Ziel der Einarbeitungsphase (3 Wochen) ist es

1. sich mit dem Framework *SalesPoint 2.0* vertraut zu machen;
2. mit dem Framework eine kleine Beispielanwendung *Gästekonto* zu entwickeln;
3. einen Projektplan mit organisatorischen Festlegungen zu entwerfen.

- zu 1.:** Lesen Sie dazu in der ersten Woche auf der Homepage von SalesPoint 2.0 besonders gründlich die Überblicksseiten zum Framework und das Tutorial.
- zu 2.:** In der zweiten und dritten Woche soll das *Gästekonto* entsprechend der nachfolgenden Spezifikation entwickelt werden. Gehen Sie dabei analog zum Tutorial vor, wo auch alle benötigten Techniken beschrieben sind. Erstellen Sie entsprechende Dokumente (Klassendiagramm, Zustandsdiagramm(e), Quellcode, lauffähige Anwendung). In der Pflichtkonsultation der vierten Woche soll ein Prototyp vorgeführt werden.

Gästekonto:

In Urlaubsgebieten gibt es manchmal Banken, die Gästekonten anbieten. Ein solches Gästekonto richtet man ein, indem man am Bankschalter einen Betrag einzahlt und ein Paßwort für das Konto vereinbart. Am Bankautomaten kann man rund um die Uhr - nach Anmeldung mit Name und Paßwort - den Kontostand erfragen und Geld abheben. Ist die Bank geöffnet, dann kann man am Schalter zusätzlich Geld einzahlen und das Paßwort ändern. Es gibt weder Kontogebühren noch Guthabenzinsen. Ein Konto wird gelöscht:

- sobald es kein Guthaben mehr ausweist oder
- drei Monate nach seiner Eröffnung; das Restguthaben fällt dann an die Bank.

Den gesetzlichen Vorschriften entsprechend protokolliert die Bank alle Vorgänge mit.

- zu 3.:** Erstellen Sie einen Projektplan mit folgenden Festlegungen:
- WER? (Teamorganisation)
 - WIE? (Durchführungsweise, d.h. teamspezifische Festlegungen wie z.B. zu
 - Codier-Richtlinie und Namenskonventionen für Dateien
 - Kommunikation

- Versionsmanagement
- gemeinsamer Login (beim FRZ zu erhalten)
- Kontrolle des Projektfortschritts durch Protokoll
- Testdurchführung
- WWW-Seiten-Erstellung)
- WOMIT? (eingesetzte Softwarewerkzeuge)

4.3.2 Analyse (OOA)

Zweck dieser Phase (2 Wochen) ist es, das gestellte Thema vollständig zu durchdringen und die Anforderungen des Auftraggebers zu erkennen. Gehen Sie wie folgt vor:

1. Identifizieren Sie anhand der Aufgabenstellung wichtige **Anwendungsfälle!**
2. Spielen Sie konkrete *Szenarien* zu diesen Anwendungsfällen durch (mitprotokollieren!) und stellen Sie dabei **CRC-Karten** auf (ebenfalls aufschreiben!).
3. Suchen Sie nach Aspekten, die in der Aufgabenstellung nur angedeutet sind oder ganz fehlen können. Behandeln Sie diese analog.
4. Ordnen Sie diese Funktionalität in einer **Prioritätenliste** an, wobei Sie aus Anwendersicht unterscheiden sollten,
 - was das zu erstellende Programm auf alle Fälle leisten **muß**
 - was es darüberhinaus leisten **soll**
 - welche Dinge es noch leisten **kann**, wenn dafür die Entwicklungszeit reicht.
5. Legen Sie diese Arbeitsergebnisse (CRC-Karten, Sequenzdiagramme für die Szenarien, Prioritätenliste) etwa nach der ersten Analysewoche ihrem Betreuer vor. Klären Sie mit ihm Zweifelsfälle und verhandeln Sie über Prioritäten, die dann am Ende der Analysephase verbindlich festgelegt werden.
6. Erstellen Sie auf der Basis der bisherigen Ergebnisse

- ein **Anwendungsfalldiagramm**;
- ein **UML Klassendiagramm**, welches die statischen Zusammenhänge der gefundenen Klassen beschreibt;
- zu den kompliziertesten Anwendungsfällen **UML-Sequenzdiagramme**.

7. Überarbeiten Sie alle ihre Dokumente und legen Sie diese ihrem Betreuer zur Zustimmung am Ende der zweiten Analysewoche vor (Pflichtkonsultation OOA).

4.3.3 Entwurf (OOD)

Ziel dieser Phase (2 Wochen) ist es, die Gesamtstruktur der Implementierung zu entwerfen sowie gegebenenfalls ausgewählte Funktionen in einem Prototyp zu implementieren. Dazu ist zunächst ein genauer Abgleich zwischen dem Analyseergebnis und der Beschreibung des Frameworks erforderlich, um festzustellen

- welche der geforderten Funktionen das Framework abdeckt;
- welche weiteren Funktionen es nach Anpassungen (wie genau sehen diese aus?) abdeckt;
- was darüberhinaus zu ergänzen ist.

Die Verantwortlichkeiten der in der Analysephase gefundenen Klassen müssen dabei solchen Klassen des Frameworks zugeordnet werden, die eben dies schon leisten, oder solchen anwendungsspezifischen Unterklassen von Frameworkklassen, die zwecks Anpassung an die konkrete Problemstellung einzurichten sind. Es entstehen neue Klassenbeschreibungen und ein neues Klassendiagramm. (An diesem Diagramm und dem aus der Analysephase werden die Gemeinsamkeiten und Unterschiede zwischen Analysemodell und zu implementierendem Modell deutlich.)

Wichtige Anpassungsschritte sind:

- die Festlegung **konkreter Bestands- und Katalogklassen**;
- die Umsetzung von Anwendungsfällen in **SaleProcesses** mit Hilfe von **Zustandsübergangsdiagrammen** (die auszuarbeiten sind);

- die Darstellung von Katalogen und Beständen in der grafischen Oberfläche; Bereitstellung von Operationen über **Menüpunkte** und **Knöpfe** (Entwurf der Benutzeroberfläche).

Im Ergebnis des Entwurfes muß festgelegt werden, welches Teammitglied für die Implementation welcher Klassen zuständig ist (und wann Fertigstellung, Ergänzung des Projektplanes).

4.3.4 Implementation und Test

In dieser Phase (3 Wochen) wird die Anwendung durch folgenden Einzelaktivitäten implementiert:

- Codierung des Entwurfes in Java 2;
- Realisierung einer funktionsfähigen Anwendung;
- ausgiebige Klassen- und Integrationstests mit anschaulichen Testdaten (können dann auch in der Vorführung benutzt werden);
- Vervollständigung und Aktualisierung der Ergebnisse aus der letzten Phase;
- Erstellung einer **javadoc**-Dokumentation.

Dabei sind in der Regel die Rubriken *muß* und *soll* der Prioritätenliste vollständig zu erfüllen. Abweichungen davon bitte frühzeitig mit dem Betreuer absprechen.

4.3.5 Wartung und Pflege

In dieser Phase (3 Wochen) sind folgende Aufgaben zu erfüllen:

- Stabilisierung/Korrektur des Programms
- ggf. Leistungsverbesserung
- Anpassung an Ergänzungswünsche des Kunden (Praktikumsbetreuer)
- Erstellung der (endgültigen) Anwender- und Entwicklerdokumentation

Durch das Erfüllen zusätzlicher Anforderungen des Kunden soll nachgewiesen werden, daß das erstellte Programm so entworfen und implementiert wurde, daß Änderungen ohne großen Aufwand möglich sind.

4.4 Pflichtkonsultationen

Die Ergebnisse jeder Phase werden im Rahmen von Pflichtkonsultationen mit den Betreuern besprochen und in einer Entwicklungsdokumentation pro Projektteam zusammengefaßt. Jedes Teammitglied hat an den Pflichtkonsultationen teilzunehmen. Zweimaliges Fehlen hat das Ausscheiden aus dem Team und dem Praktikum zur Folge. In der ersten Pflichtkonsultation (nach der Einarbeitungsphase) wird die Qualifikation der einzelnen Teammitglieder hinsichtlich **Java-Programmierkenntnisse** und ihrer **Kompetenz zur Beispielanwendung** geprüft. Teammitgliedern mit ungenügenden Kenntnissen wird die weitere Mitarbeit am Praktikum (nicht zuletzt aus Fairness gegenüber den anderen Teammitgliedern) verwehrt. Gegebenenfalls müssen nach dieser Phase wegen Ausschluß von StudentInnen vom Praktikum die Aufgabenstellungen reduziert bzw. Praktikumsgruppen neu formiert werden. Die Teilnahme an der Abschlußpräsentation kann nur nach allen erfolgreichen Pflichtkonsultationen eines Teams erfolgen. In Pflichtkonsultationen erteilte Auflagen sind spätestens in der nächsten Pflichtkonsultation abzurechnen.

4.5 Beleg und Abschlußpräsentation

Das Praktikum schließt mit einer Präsentation des Beleges einschließlich der Vorführung des entwickelten Softwareprodukts ab.

4.5.1 Beleg

Der Abschlußbeleg muß sowohl eine Entwicklungsdokumentation (mit entstandenen Dokumenten während des Praktikums) als auch eine Anwenderdokumentation enthalten. Bei der Erstellung des Beleges ist Wert auf einen guten äußeren Eindruck (Rechtschreibung/Grammatik/Ausdruck, Gliederung der Dokumentationen) zu legen. Die vollständige oder teilweise Bereitstellung der Dokumentationen im WWW ist gewünscht.

Entwicklungsdokumentation. Die Entwicklungsdokumentation soll im wesentlichen die Dokumente aus den einzelnen Projektphasen zusammenfassen.

- Projektplan

- Dokumente aus der Analysephase (Prioritätenliste, CRC-Karten, Anwendungsfalldiagramm, Sequenzdiagramme für alle wichtigen Szenarien, statisches Modell)
- Entwurfsdokumente mit Begründung von Entwurfsentscheidungen (für SalesPoint 2.0 angepaßtes Klassendiagramm und Klassenbeschreibungen, Zustandsübergangsdiagramme zu den Anwendungsfällen)
- Java-Quellcode
- API-Dokumentation in HTML-Format (Nutzung von *javadoc*)
- Bewertung der Lösung (Erreichtes versus Plan, Qualität)
- Bewertung des Projektverlaufs (Was haben die einzelnen Teammitglieder geleistet? Änderungen in der Organisationsstruktur? Techniken des Projektmanagements?)
- beantworteter Fragebogen einschließlich sorgfältiger Ermittlung der geforderten Metriken

Achten Sie dabei darauf, daß

- Analyse-, Entwurfs- und Implementierungsentscheidungen begründet sind;
- bei erfolgten Iterationen im Lebenszyklus des Softwareprodukts alle Modelle aktualisiert werden;
- die Übereinstimmung der Implementation mit den objektorientierten Modellen gegeben ist;
- die Implementation ausreichend kommentiert ist.

Anwenderdokumentation. Das Handbuch für den Anwender des erstellten Softwareprodukts soll unter Beachtung folgender Anforderungen knapp geschrieben sein:

- Die angebotene Information muß fehlerfrei und vollständig sein.
 - Was tut die Software?

- Wie können die Anwender erreichen, daß sie es tut? (Handlungsweisen)
 - Vor welchen Fehlern müssen Sie den Anwender bewahren?
 - Welches Hintergrundwissen brauchen die Anwender über die Funktionsweise?
- Die Anwender müssen die Information gut finden können.
 - Die Dokumentation muß ansprechend gestaltet sein.

4.5.2 Abschlußpräsentation

Die Präsentation der Praktikumsergebnisse jedes Projektteams findet in der letzten Praktikumswoche statt. Eine Verlängerung der Bearbeitungszeit ist **nicht** möglich! Es werden Termine festgelegt, zu denen jeweils drei Projektteams ihr Softwareprodukt und ihren Beleg im Rahmen von jeweils 30 min zu verteidigen haben (ca. 10 min zum Projektverlauf, 10 min Vorführung, 10 min Diskussion). Die Teilnahme aller Teammitglieder ist Pflicht!

4.6 Rahmenzeitplan

Der Ablauf des Praktikums wird als Rahmenzeitplan vorgegeben. Die Projektteams können diesen Zeitplan in gewissen Grenzen entsprechend ihren konkreten Bedingungen in Absprache mit dem Praktikumsbetreuer modifizieren (Festlegung im Projektplan). Der **späteste Abgabetermin** von Beleg und Softwareprodukt ist der **07.07.2000, 15 Uhr** beim Betreuer oder im Sekretariat des Lehrstuhles (DUE 26/258)!

Der Zeitrahmen für die Absolvierung des Praktikums ist damit sehr begrenzt. Für die effektive Bearbeitung der Aufgabenstellung stehen 13 Wochen zur Verfügung. **Eine konzentrierte und gut organisierte Bearbeitung der Aufgabe ist für den erfolgreichen Abschluß des Praktikums unbedingt erforderlich!**

Semester- woche	Termin/ Zeitraum/Ort	Einzelaktivitäten
1. EINARBEITUNG	03.04.-07.04.	
	03.04., 5.DS GRU 172	Einweisung ins Praktikum und Einschreibung
	bis 05.04./ Schr. DUE26/258	Einschreibung ins Praktikum
	07.04./ Aushang	Bekanntgabe der Gruppen mit Zuordnung der Aufgabenstellungen (Aufgabenstellungen siehe WWW-Seiten des Lehrstuhls)
2.-3. EINARBEITUNG	10.04.-20.04.	Projekt <i>Gästekonto</i> Erstellung des Projektplanes
	10.04., 5. DS GRU 171	Tutorial zum Framework
	20.4.	Abgabe der Dokumente - Prototyp <i>Gästekonto</i> - Projektplan
4.-5. OOA	25.04.-05.05.	Analyse
	25.04.-28.04.	Pflichtkonsultation zur Einarbeitung (Test auf weitere Teilnahme)
	05.05.	Abgabe der OOA-Dokumente
6.-7. OOD	08.05.-19.05.	Entwurf
	08.05.-12.05.	Pflichtkonsultation zur Analyse
	19.05.	Abgabe der OOD-Dokumente
8.-10. IMPLEMENTATION	22.05.-09.06.	Implementation und Test
	22.05.-26.05.	Pflichtkonsultation zum Entwurf
	09.06.	Abgabe der lauffähigen Anwendung
11.-13. WARTUNG	19.06.-07.07.	Wartung und Pflege
	19.06.-23.06.	Pflichtkonsultation zur Implementierung
	07.07.	Abgabe des Beleges
14. ABSCHLUSS	10.07.-14.07.	Abschlußpräsentation