

Hinweis: Die Arbeitsergebnisse sollen auf den Rechnern im Pool der Fachgruppe vorgestellt werden. Die Aufgaben selber können auch an Privat-Rechnern bearbeitet werden. Es empfiehlt sich der Einsatz einer Unix-Plattform (z. B. Solaris, Linux, etc.).

Hinweis: Literaturquellen zu den einzelnen Aufgaben sind jeweils im Anschluß an die Aufgabe angegeben.

Die Aufgaben als PDF zum Ausdrucken:

<http://pi.informatik.uni-siegen.de/lehre/2002w/seu/scm/aufgaben.pdf>

## 1 Szenario

Sie arbeiten in einem jungen Softwarehaus als Entwickler in einem Team mit zwei Kollegen. Ihr Team hat von Ihrem Chef den Auftrag bekommen, die Überarbeitung der bereits existierenden Software PI-SET/TT zur Erfassung von Arbeitsaufwänden vorzubereiten. Die Software wurde für den Single-User Betrieb erstellt. Daraus resultiert die Speicherung der Daten in Dateien. Ihr Auftraggeber wünscht, daß die Software mehrbenutzerfähig wird. Um der Lohnabteilung einen Überblick über die Arbeitszeiten der Mitarbeiter zu geben, sollen die Daten aller Mitarbeiter zentral gesichert werden. Im Zuge dessen soll die Datenhaltung auf eine Datenbank umgestellt werden.

### Lernziele dieses Vorlesungsteils

- Grundlagen und Einsatzbereiche von Werkzeugen erkennen
- Vor- und Nachteile von Werkzeugen erkennen
- Erkenntnisse praktisch erproben
- Werkzeuge: Rational Rose, cvs, ant, bash

## 2 Aufgabe: Reengineering (50 Punkte)

*Zeitschätzung: 9h*

**Hinweis: In diesem Aufgabenteil müssen keine Implementierungsarbeiten geleistet werden!**

Vom Auftraggeber haben Sie den Quellcode von PI-SET/TT erhalten, dieser ist in einem Versionsmanagement-System (CVS) gespeichert. Sie haben den Auftrag erhalten, sich den Quellcode anzusehen und einen Vorschlag für eine Umstrukturierung des Codes zu machen, da die alte Struktur für die neuen Anforderungen (mehrbenutzerfähig und Speicherung in einem DBMS) nicht mehr geeignet ist.

1. Machen Sie ein Check-Out des Quell-Text.
2. Nachdem Sie den Quelltext vorliegen haben, verschaffen Sie sich einen Überblick über die existierende Struktur, indem Sie ein Werkzeug zur Erstellung von UML-Klassendiagrammen nutzen, mit dem Reengineering möglich ist.
3. Anschließend sollen Sie mit Hilfe des Werkzeugs die vorliegende Struktur in eine 5-Schichten Architektur überführen. Überlegen Sie sich im Zuge dessen, wie Sie die Klassen auf die einzelnen Schichten verteilen können und wie die Datenhaltung in die neue Architektur integriert werden kann.

Material: 6.2

## 3 Aufgabe: Versionsmanagement (100 Punkte)

*Zeitschätzung: 24h*

Zur Weiterentwicklung von PI-SET/TT soll der Quellcode in einem anderen Versionsmanagement-System (CVS) gespeichert werden. Da auch externe Programmierer eingesetzt werden, die nicht innerhalb des Firmennetzwerks arbeiten, soll ein entfernter Zugriff auf CVS möglich sein.

1. Informieren Sie sich, wie man ein CVS-Repository für den entfernten Zugriff einrichtet. Ihr Chef wünscht, daß sich alle Mitarbeiter mit einem Login-Namen und Passwort bei CVS anmelden müssen, bevor sie darauf zugreifen können.

2. Nachdem Sie sich mit der Dokumentation beschäftigt haben, legen Sie ein neues CVS-Repository für eine Gruppe von drei Programmierern an, denen Sie einen Passwortzugang nach CVS einrichten. Nutzen Sie hierfür das pserver-Protokoll von CVS. Das CVS-Repository muß im Verzeichnis \$HOME/cvshome liegen, damit der Zugriff mittels pserver-Protokoll funktioniert. Der Zugriff erfolgt über die Maschine pi72.informatik.uni-siegen.de.
3. Anschließend erstellen Sie die neue Struktur für die Software, die Sie sich in Aufgabe 2.3 überlegt haben. Um den Quellcode aus dem alten CVS-Repository unter Beibehaltung der Versionshistorie in das neue Repository zu überführen, gibt es mindestens zwei Ansätze. Beschreiben Sie die beiden Ansätze und diskutieren Sie die Vor- und Nachteile. Realisieren Sie einen Ansatz.
4. Markieren Sie alle aktuellen Versionen, um später den Ausgangszustand der Änderungen wieder rekonstruieren zu können.
5. Wie Sie bei der Umstrukturierung sicherlich festgestellt haben, basiert PI-SET/TT auf der Komponente genform. An dieser soll unabhängig entwickelt werden können, ohne immer den ganzen Quelltext von PI-SET/TT mit auschecken zu müssen. Nutzen Sie hierzu die Möglichkeiten von CVS.
6. Ihr Chef hat von einem Bekannten gehört, der ebenfalls Chef eines kleinen Softwarehauses ist, daß es mit CVS möglich sein soll, sich über Änderungen an einzelnen Dateien informieren zu lassen. Er möchte, daß Sie CVS so konfigurieren, daß er über alle Commits per E-Mail informiert wird.
7. Desweiteren möchte Ihr Chef, daß Sie eine Möglichkeit finden, damit er sich darüber informieren kann, welcher Programmierer wann welche Zeile in einer bestimmten Datei geändert hat und wie der aktuelle Zustand einer Datei im Workspace ist.
8. Ihr Chef möchte den Programmierern ein Werkzeug zur Verfügung stellen, mit dem es möglich ist, Unterschiede zwischen zwei Datei-Versionen anzuzeigen und zu mischen. Er nannte Ihnen tkdiff aus dem tkcvs-Paket und Guiffy. Begutachten Sie beide Werkzeuge. Welche Unterschiede gibt es und welche Gemeinsamkeiten?
9. Während Sie das CVS-Repository vorbereiten, hat ein Freelancer einige bahnbrechende Funktionen implementiert (PI-SET/TT – jetzt noch besser). Sie wissen, daß der Freelancer auf der Version vom 1.1.2001 aufgesetzt hat. Deswegen sollen Sie den Zustand des Repository vom 1.1.2001 auschecken und darauf einen Nebenentwicklungszweig anlegen. Der Quellcode vom Freelancer soll dann in den Nebenentwicklungszweig eingespielt werden.
10. Mischen Sie den Quellcode des Nebenentwicklungszweigs in den Hauptentwicklungszweig ein. Beachten Sie, daß Sie evtl. auftretende Konflikte nicht lösen, sondern sich nur eine Lösungsstrategie überlegen.

Material: 6.3

## 4 Aufgabe: Releasemanagement (50 Punkte)

Zeitschätzung: 9h

In dem Softwarehaus, in dem Sie arbeiten, war es bisher üblich, daß ein Entwickler bei einem Release den gesamten Quelltext übersetzt und zusammengepackt hat.

Ein anderer Bekannter teilte Ihnen später mit, daß es mit einem Maketool wie *ant* noch einfacher ist, alles zu übersetzen. Desweiteren würde es *ant* auch ermöglichen, nicht immer alle Dateien zu übersetzen. Es würde nur das übersetzt, was sich wirklich geändert hat:



Im Beispiel würde eine Änderung in Alpha nur zu einem Übersetzen von Alpha führen. Bei einer Änderung z.B. an der Schnittstelle von Bravo würden hingegen Bravo und Alpha übersetzt.

Um die Aufgabenstellung zu lösen, müssen Sie zuerst das Werkzeug *ant* installieren. Alle Aufgaben müssen Sie dann auf dem Nebenzweig des CVS-Repository durchführen, damit Sie den Quelltext compilieren können. Schreiben Sie dann eine Steuerdatei für *ant*, mit der Sie:

- den Quelltext von PI-SET/TT mit Berücksichtigung der Abhängigkeiten zwischen den Java-Klassen übersetzen können

- bei Bedarf alle generierten Dateien wieder löschen können
- die Dokumentation mit Hilfe von JavaDoc generieren können
- alle übersetzten Klassen in ein Jar-Archiv packen können
- das Jar-Archiv auf einem ftp-Server ablegen können

Wie würden Sie die geforderte Funktionalität ohne *ant* oder ähnliche Werkzeuge realisieren?

Material: 6.4

## 5 Aufgabe: Shell-Programmierung (100 Punkte)

*Zeitschätzung: 16h*

Immer häufiger hören Sie Meldungen von Kunden, die PI-SET/TT nicht starten können oder bei denen sich die Anwendung sofort wieder beendet. Sie finden heraus, daß in der Umgebung dieser Kunden oft die Voraussetzungen zum Einsatz von PI-SET/TT nicht gegeben sind. Entweder setzen diese Kunden eine zu alte oder zu neue Version des JDK ein oder benötigte Bibliotheken sind nicht installiert oder nicht über den Klassenpfad erreichbar. Sie beschließen, PI-SET/TT um ein Shell-Script zu erweitern, das sicherstellt, das die Anwendung die erforderliche Umgebung vorfindet und diese dann startet. Aus den verschiedenen Shell-Derivaten wählen Sie die Bourne-Again-Shell (bash), da diese auf vielen Plattformen vorhanden ist und einige Konstrukte anbietet, welche die Entwicklung von Skripten vereinfachen. Strukturieren Sie das Skript, indem Sie z. B. Funktionen einsetzen.

Das Skript muß folgende Überprüfungen vornehmen:

- die JDK-Version: 1.1.8, 1.3.x laufen
- swingall.jar, xerces.jar, pisett.jar müssen im Klassenpfad oder im Installationsverzeichnis vorhanden sein. Im letzten Fall ist der Klassenpfad zu erweitern.
- das Script teilt fehlende Voraussetzungen mit und bricht ab
- der Schalter *-force* startet die Anwendung trotzdem mit dem Hinweis auf fehlenden Support (Der Kunde weiß schon was er tut. Finger weg von unserem PMS!)
- der Schalter *-help* bietet einen kurzen Hilfetext zur Bedienung des Wrappers

Material: 6.5

## 6 Material

### 6.1 Ansprechpartner

Dirk Ohst, <mailto:ohst@informatik.uni-siegen.de>

Michael Welle, <mailto:welle@informatik.uni-siegen.de>

### 6.2 Material zu Reengineering

Skript zur Vorlesung ST I:

[http://pi.informatik.uni-siegen.de/kelter/lehre/02w/lm/lm\\_sar\\_20030318\\_info.html](http://pi.informatik.uni-siegen.de/kelter/lehre/02w/lm/lm_sar_20030318_info.html)

Werkzeuge:

<http://pi.informatik.uni-siegen.de/welle/tmp/rose/rose.htm>

Repository:

<http://pi.informatik.uni-siegen.de/lehre/2002w/seu/scm/repository.tar.gz>

## 6.3 Material zu CVS

Cederqvist: Version Management with CVS

- branching / merging Kap.5 8S.
- tagging Kap.4 5S.
- create/remote Kap.2 20S. (2-2.2.1 (4), 2.9, 2.9.1-3.3 (5))
- status/annotate Kap.10 2S.
- notification Kap.10.6 4S.
- import/module Kap.3 3S.

ca. 40 Seiten

Onlinedokumentation:

Cederqvist:

<http://pi.informatik.uni-siegen.de/lehre/2002s/propra/cederqvist-1.11.1p1.pdf>

CVS-Buch:

<http://pi.informatik.uni-siegen.de/lehre/2002s/propra/cvsbook.ps.gz>

CVS-Home:

<http://www.cvshome.org/docs/manual/cvs.html>

CVS-Film:

<http://pi53.informatik.uni-siegen.de:8180/gCVS/index.html>

Erweiterungen:

<http://pi.informatik.uni-siegen.de/lehre/2002w/seu/scm/freelancer.tar.gz>

## 6.4 Material zu Releasemanagement

<http://jakarta.apache.org/ant/manual/index.html>

Bei der Umstrukturierung der Java-Quellen müssen ggfs. die *package*- und *import*-Anweisungen der Java-Klassen angepasst werden. Diese Aufgabe übernimmt das Skript `makepackages.pl`: <http://pi.informatik.uni-siegen.de/lehre/2002w/seu/scm/makepackages.pl>

## 6.5 Material zur Shell-Programmierung

kurze Einführungen:

[http://pi.informatik.uni-siegen.de/kelter/lehre/02w/lm/lm\\_ksp\\_19981201\\_info.html](http://pi.informatik.uni-siegen.de/kelter/lehre/02w/lm/lm_ksp_19981201_info.html)

<http://www.linuxfocus.org/Deutsch/September2001/article216.shtml>

<http://www.tu-berlin.de/zrz/information/paper/instant/sh/>

ausführliches Skript:

<http://123.koehntopp.de/kris/artikel/unix/shellprogrammierung/>