

Abschlussbewertung für das Projekt LESSENTROPY

David Kunz (s713203) Alexander Metzner (s712719)
René Nicklas (s712773)

Projekt im Rahmen der Veranstaltung Softwareprojekt II
an der Technischen Fachhochschule Berlin
im Sommersemester 2005
Prof. Knabe

Inhaltsverzeichnis

1	Beschreibung des Projekts LESSENTROPY	1
2	Aufgabenverteilung der Gruppenmitglieder	1
3	Verwendete Technologien	2
4	Verwendete Software	3
5	Umsetzungsstatus gemäß Pflichtenheft	3
6	Erfahrungen	3
7	Abschließendes Fazit	4

1 Beschreibung des Projekts LESSENTROPY

Das Projekt LESSENTROPY ist eine Client-Server basierte Software, die es ermöglicht, Datenmedien (CDs, DVDs, USB Massenspeicher, externe Festplatten, Smarccards) zu indizieren und diese Indices in einer zentralen Datenbank zu speichern.

Die Software stellt eine komfortable GUI zur Verfügung, die diese Indices durchsuchen kann. Der Benutzer kann auf verschiedenste Art und Weise die Suche beeinflussen sowie die Ergebnismenge einschränken.

Des Weiteren können die Indices in ein XML Format exportiert werden und später (in eine weitere Installation) wieder importiert werden.

2 Aufgabenverteilung der Gruppenmitglieder

Tabelle 1 gibt die Verantwortlichkeiten der einzelnen Gruppenmitglieder wieder.

Name	Matrikelnr.	Komponenten (Fachschichten)	Basistechnologien
David Kunz	s713203	Kommunikation (client- und serverseitig), Testtreiber (Fachlogikschicht)	RMI, JUnit, XSLT
Alexander Metzner	s712719	Persistenzschicht, Fachlogikschicht, Testtreiber (Fachlogik und Persistenz), Supportframeworks (Konfiguration, Stringmanagement, Exception utilities), Wiring, Deployment	JDBC, Reflection, XML (dom4j), Ant, JUnit
René Nicklas	s712773	UI Schicht, Fachlogikschicht, Testtreiber (Fachlogik)	Swing, JUnit

Tabelle 1: Verantwortlichkeiten der Gruppenmitglieder

3 Verwendete Technologien

Java 5 Standard Edition Java 5 vor allem, um die Vorteile von Generics nutzen zu können. Wir haben uns dennoch bemüht, alle weiteren Neuerungen gegenüber Java 1.4 ebenfalls zu verwenden (Annotations, Enums, Autoboxing)

Swing Da Java verwendet wurde, bot sich die Verwendung von Java Swing für die GUI Schicht ebenfalls an.

JDBC Da auch JDBC den Standard für den Datenbankzugriff unter Java darstellt, haben wir dieses API ebenfalls verwendet.

Reflection Im Zuge der Erstellung eines eigenen OR Mappers brachte der Einsatz von Reflection eine große Vereinfachung, da viele Aufgaben generisch in einer Klasse zu lösen waren.

RMI Ebenfalls Bestandteil des J2SE und daher gut für die Java - Java Kommunikation geeignet.

XML/ XSLT Um Daten plattformunabhängig darzustellen, ist XML mittlerweile die erste Wahl. In Kombination mit XSLT ergab sich für uns nicht nur ein mächtiges Werkzeug für die Codegenerierung (von externen Strings) sondern auch für die Erstellung der Dokumentation.

Ant Build und deployment Werkzeug für Javaapplikation, das sehr gut erweitert werden kann (bspw. JUnit).

JUnit Das Standard Unittest Framework für Java.

CVS Da dieses Projekt als Sourceforgeprojekt¹ realisiert wurde, waren wir an das CVS von Sourceforge gebunden. Die Wahl war dennoch gut, vor allem durch die hervorragenden Eclipseintegration von CVS.

¹siehe auch <http://www.sourceforge.net/projects/lessentropy>

4 Verwendete Software

Poseidon For UML – Community Edition 3.0 Gutes freies UML Werkzeug für das Forwardengineering. Leider keine gute Unterstützung für Reverse Engineering (in der gegebenen Version). Keine Unterstützung für Generics und Enums.

Eclipse 3.1 Sehr gute, anerkannte und freie (open source) Entwicklungsumgebung für Java.

HSQldb Freie (open source) java basierte Embedded Datenbank, die vor allem für die Entwicklung enorme Vorteile bringt.

MySQL Bekannte freie (open source) relationale Datenbank, die für nicht allzu große Datenmengen sehr gut skaliert.

5 Umsetzungsstatus gemäß Pflichtenheft

Durch das frühzeitige Ausscheiden eines vierten Teammitgliedes konnte die Umsetzung aller geforderten Kriterien des Pflichtenheftes nicht eingehalten werden. Dennoch wurde ein Großteil realisiert.

Folgende Abweichungen sind festzustellen:

- Die Pluginschnittstelle sowie das ZIP Archiv Plugin wurden nicht realisiert.
- Es wurden nur zwei der geforderten Datenbanken angebunden. Die Anbindung weiterer, Datenbanken ist jedoch mit wenig Aufwand verbunden.
- Es wurden keine nativen Startup Applikation für die geforderten Betriebssysteme erstellt.
- Die geforderten Performancevorgaben können nur bedingt eingehalten werden.

Alle weiteren Vorgaben, Funktionalitäten und Kriterien wurden gemäß Pflichtenheft umgesetzt.

6 Erfahrungen

Die Analyse zeigte sich als äußerst gut und vollständig, so dass keine weiteren Änderungen an den erstellten Modellen vorzunehmen war. Der Entwicklungsprozess wurde klar eingehalten.

[dk]

Die relativ engen Projektvorgaben, die im Hinblick auf die abschliessende Bewertung sicher auch ihre Berechtigung hatten, führten, für das zweite Semester, leider zu einem relativ strikten Projektplan. Um diesen einzuhalten, konnte nicht intensiv nach alternativen Lösungen gesucht werden.

Bei fehlender Vorkenntnis, einer Technologie, konnte schnell viel Zeit fehlinvestiert werden, wenn sich diese Technologie im Nachhinein als ungeeignet herausstellte.

Besonders lehrreich war der vielfältige Einsatz von Java Technologien im Zusammenspiel.

[am]

Die Implementierung eines eigenen OR Mappings stellte eine große Herausforderung dar. Das entstandene Mapping skaliert relativ gut für den gegebenen Kontext, ist jedoch sicherlich nicht als generelle Lösung für andere Projekte einsetzbar.

Insgesamt war das Projekt mit über 200 Klassen und mehr als 14.000 Zeilen Quellcode zu groß für drei Mann in einem Semester. Allerdings ist es auch schwer vorstellbar, ein noch kleineres Projekt zu realisieren so dass dabei sowohl technisch- fachlicher Anspruch wie auch persönliche Motivation vorhanden sind.

[rn]

Bei der Arbeit an dem Projekt LessEntropy hatte ich die Möglichkeit mich mit vielen neuen Techniken zu beschäftigen. (JUnit, dom4J, Java 1.5, ...). Zum Anderen konnte ich mich in bereits bekannte Themen tiefer einarbeiten und so verschiedene Eigenarten der Techniken besser kennenlernen (Swing, XML, ...). Das Einteilen des zeitlichen Rahmens war zusätzlich eine Aufgabe, welche immer höhere Priorität erlangte, je mehr die Abgabe des Projekts näher rückte.

7 Abschließendes Fazit

Das Projekt konnte unsere Meinung nach zufriedenstellend abgeschlossen werden. Dennoch lief nicht alles optimal. Probleme mit schlecht dokumentierter oder gar nicht funktionierender Software Dritter kostete viel Zeit.

Rückblickend stellen wir fest, dass die strikte Trennung von Analyse und Realisierung auf zwei Semester ungünstig ist. Viel Zeit für die Analyse aber dann doch recht wenig Zeit für die Realisierung machen es schwer, konstant hohe Qualität zu produzieren, vor allem bei zeitlich nicht immer gut einschätzbaren Quellcodearbeiten.

Auf einige Aspekte wurde seitens des Dozenten genaustens geachtet (Transaktionssicherheit, Persistenz, Exceptionhandling) andere blieben jedoch gänzlich unbeachtet (GUI, Kommunikation, Nebenläufigkeit), so dass vielen Arbeiten nur wenig Beachtung und Weiterentwicklung zugute kam.

Die beiden Semester (Softwareprojekt I und II) haben viele Erfahrungen in Sachen Teamarbeit, große Projekt und längere Planung mit sich gebracht, die allesamt als positiv zu bewerten sind.