

Abschlußbesprechung Software-Projekt II (Projekt-Realisierung) bei Prof. Knabe im SS 2003

Hier die in der Abschlußbesprechung am 18.07.2003 geäußerten Meinungen der Student(inn)en, gegliedert nach Themengebieten und Meilensteinen.

Legende: + positiv bewertet, – negativ bewertet oder soll geändert werden, ~ teils/teils

Analysephase

-	iNews: Aufwand bei Problemen mit Techniken nicht abschätzbar
-	iNews: Analysephase zu umfangreich im Vergleich zur Realisierung
+	Publicity: Einarbeitung in Basistechniken in VL-freier Zeit war gut
-	Späte Implementierung verzögert Kennenlernen der Arbeitsdisziplin der Kollegen
~	UI-Prototyp: Hinweis auf realen Einsatz geben
-	Schon vor Analyse das akademische Bank3Tier-Beispiel mit Klassenvervielfältigung zeigen
-	Skript + SWP II-Webseite möglichst früh bekanntgeben
-	Statt Muß/Kann-Kriterien mehrere Ausbaustufen planen
-	Über vorlesungsfreie Zeit schon Zerreiß-Durchstich für das eigene Projekt als technische Einarbeitung deutlich empfehlen!
?	Wie analysiert man Client/Server-Systeme? Kann man da von idealer Technik ausgehen? Wie vermeidet man Redundanz in der Analyse?

Meilensteine

2 Präsentation Einarbeitungsnachweise

-	Zu weltfremd, gibt falsche Sicherheit.
-	Präsentationen waren zu umfangreich, Papers wären besser
-	Besser schon als Zerreiß-Durchstich für das eigene Projekt als deutliche Empfehlung! Wie die saubere Lösung dann aussieht, wird in den ersten 4 Wochen unterrichtet.

3 Entwurf: Lg -Klassendiagramm + Lg-Operations-Spezifikationen

~	Wurde bei vielen Gruppen 1:1 umgesetzt, wiegte einen jedoch in falscher Sicherheit ob der technischen Probleme
---	--

5 DB-Entwurf + Klassendiagramme Gesamtsystem + Zuständigkeitsverteilung

--	--

7 1. Durchstich: Objektverwaltung Klasse Benutzer über 3 Schichten + Testtreiber

+	Klasse Benutzer gut zum Einstieg geeignet
-	Besser gleich 2 Klassen

9 2. Durchstich: Objektverwaltung anderer Klasse + Testtreiber

--	--

11 3. Durchstich: Eine Objektverbindungsverwaltung 3 Schichten, mit Testtreibern

-	Etwas spät
---	------------

13 Integrationstest

-	War ein zu großer Sprung nach Objektverbindungsverwaltung
---	---

15 Präsentation Gesamtsystem öffentlich

-	Zu kurz für die viele Arbeit
-	Entwickler würden gerne auch auf die technischen Probleme eingehen

17 Abnahme

-	Komplexität der erstellten Software konnte nicht richtig gewürdigt werden.
---	--

Andere Projektmodelle

+	Individuelle Vereinbarung von Meilensteinen bei Gruppe iNews war gut
-	Bei passenden Projekten wären vertikale Komponenten als Meilensteine besser geeignet

Vorlesung

-	Stoffauswahl: Was fehlte? Diskussion von Projektgruppenproblemen im Plenum
-	Bewertungskriterien deutlich machen: Exceptionhandling, Transaktionssicherheit, Fachkonzept als API, Testtreibersuite, Redundanzfreiheit
-	Das Skript sollte keine didaktischen Lücken haben. Bem. Knabe: Nötig für Diskussion und Entwicklung in der Vorlesung.
-	Zeit für Projektpräsentationen, auch der technischen Aspekte, reichte nicht aus
-	Für die Lehrveranstaltung fehlte im Web ein Diskussionsforum

Rücksprachen

-	Zu wenig über Architektur gesprochen
-	Am Anfang jeder Sitzung sollte Frage gestellt werden, welche Probleme die Gruppe hat, damit die Zeit nicht nur für die Qualitätsbewertung draufgeht.

Gruppenarbeit

-	Projektarbeit und Kommunikation wurden in der Vorlesung nicht thematisiert
-	Ansprechpartner für Probleme der Zusammenarbeit wäre gut, Bsp. Herrn Haack (KUP) fragen.
-	Es besteht große Scheu, Gruppenprobleme dem Professor gegenüber zu benennen, daher kommen diese erst sehr spät zum Vorschein.

Techniken

-	Es wäre gut, wenn Studenten eigenen Server im Labor aufstellen und ans Internet anschließen könnten. Sicherheitsprobleme sollten lösbar sein. Integrationsprobleme lassen sich nicht über Laptops lösen!
---	--

Einsatz von CVS

+	Sehr positiv, gut in Eclipse integriert, auch Paket umbenennen
+	Wäre ohne CVS nicht gegangen
-	Bei CVS-Problemen/Spezialwünschen wie Versionierung reichte Support nicht aus.
-	Diskquota auf Labor-CVS-Server reichte nicht aus

Einsatz von Castor und anderen OR-Mappern

+	Gruppe Publicity: Castor OK
-	Castor: Analyse-Klassendiagramm nicht direkt umsetzbar
-	Castor: Keine Polymorphie, rekursive Assoziation schlecht möglich
-	Castor: Dokumentation stellt nur die einfachsten Fälle dar
-	Castor sollte noch nicht empfohlen werden (Version 0.9)
+	Hibernate: gut, auch gut dokumentiert
+	Apache-OJB: gut, Dokumentation weniger

Einsatz von Struts

+	Kann man empfehlen
+	Gut für JSP Model 2-Architektur, Tiles gut als Frame-Ersatz

Einsatz von MulTEx

+	Vorgabe eines Weges zum Umgang mit Ausnahmen war gut
---	--

Einsatz von JUnit-Testtreibersuite

+	War eine große Hilfe
+	Steigert Vertrauen in eigenen Code
+	Half, Seiteneffekte bei Veränderungen aufzudecken
-	Kein Nutzen bei nachträglicher Erstellung der Testtreiber
-	Bei Testen einer Schicht für sich (mittels Mock-Objekten) Riesenaufwand

Studiengang

-	Schon im Grundstudium 3./4. Semester ein Mini-Projekt, Synergie zwischen PR IV und SWE nutzen.
---	--