Abschlussbesprechung Software-Projekt II (Projekt-Realisierung) bei Prof. Knabe im WS 2014

Hier die in der Abschlussbesprechung am 11.02.2015 geäußerten Meinungen der Student(inn)en, gegliedert nach Themengebieten und Meilensteinen.
Legende: + positiv, – negativ bewertet, ! Wunsch, ~ teils/teils

Vorsemester Analysephase/Durchstiche

+	Gut 1. Sem. Infrastruktur, dieses Sem. Funktionalität

Meilensteine

4 Zentrales Ausnahmemelden

+	Template Method Pattern war gut.
+	Half bei der Fehlerdiagnose.
	MulTEx lief nicht auf Android vermutlich, da dort Reflection nicht so funktioniert. Anwendung stürzte in Folge ab.

5 2. Durchstich: Objektverbindungsverwaltung + Suchen Objektmenge + Testsuite

~	WG-App: Problem mit Kaskadierung trat dabei noch nicht zutage.
-	Terminverwaltung: Problem gegenseitiger Abhängigkeit zwischen User und Group (m:n) trat dabei noch nicht auf.
!	Es sollte möglichst eine komplizierte Assoziation für diesen Meilenstein ausgewählt werden, um späte Überraschungen zu vermeiden.

7 Freier Meilenstein

+	WG-App: Sehr gut, da individuell aufs Projekt abstimmbar.
+	WG-App: Allerdings auch gut, dass eine Projektstruktur mit bestimmten Meilensteinen vorgegeben wurde.
!	Terminverwaltung: Aufwand von Funktionalität, die durch alle Schichten hindurch implementiert werden muss, wird zu leicht unterschätzt.
+	Tour2Go: Positiv war Chance, eigene Möglichkeiten selbst einzuschätzen.

9 Qualität: Spezifikationen, Komponentenorientierung, Richtlinien

+	Terminverwaltung: Idee war gut, noch mehr Code Review wäre gut.
-	WG-App: Problem aus Vorsemester bzgl. Datenbank-Singleton wurde erst dieses Semester beanstandet.
~	Rich Domain Model wurde nicht richtig umgesetzt. Fehlen Beispiele im Internet.
~	Roboter: Beim Logging war das Singleton-Vermeiden sehr umständlich, ansonsten schon OK.

11 Freier Meilenstein

	s.o.

13 Integrationstest

~	Knabe: Wurde dieses Semester nicht als Fertigstellen der Gesamtapplikation verstanden. Der Meilenstein sollte umbenannt werden nach **Systemtest**.

14 Präsentation Gesamtsystem öffentlich

~	Problematisch beim Präsentieren von Mobile Apps. Gelang für Android mit ChromeCast.
!	Muss geplant werden mit ihrem Aufwand. Projektion der Mobile-Oberfläche wäre auch für die Rücksprachen gut.
+	Sehr interessant, die Leistungen und Probleme der anderen Teams zu sehen. Sollte öfter stattfinden, um Austausch über Probleme anzustoßen.
!	Forum in Moodle für Austausch wäre gut.

!	Im Plenum Austausch über Probleme mit anderen Gruppen.
!	Es gab gemeinsame technische Herausforderungen bei allen Gruppen, z.B. mit JPA/Hibernate und Ausnahmebehandlung.

15 Abnahme

Unterricht

	Sinnvoll?
	Stoffauswahl: Was interessant? Zentrales Ausnahmemelden
	Stoffauswahl: Was überflüssig? Rest Knabe: In welcher Schicht Transaktionsgrenzen definiert werden müssen und Begründung für Komponentenorientierung sind unverzichtbar.
!	Programmierrichtlinien waren schon bekannt. Sollten auf höheres Semester umgestellt werden. Sollte zu Projektanfang ausgegeben werden.
	Stoffauswahl: Was fehlte? Hinweis auf andere Asynchronitätsansätze wie Aktoren/Futures.
	Unterrichtsstil?
	Materialien auf meiner SWP-II-Seite? Teils schwer aufzufinden, da es keine Navigation auf der Seite gibt.

Rücksprachen

+	Meilensteinrhythmus alle 2 Wochen? War OK.
+	Anwesenheitspflicht bei Rücksprachen? War im Prinzip gut, da dann alle da sind. Dadurch kann sofort über Probleme in allen Schichten geredet werden.

Gruppenarbeit

+	Selbstauswahl der Teammitglieder?
+	Gruppengröße 4-5? Sinnvoll, auch wenn dies die Freiheit der Gruppenwahl einschränkt. Je größer die Gruppen werden, desto schwieriger ist es, Gruppensitzungen abzuhalten.
	Projektleiter? Teams kamen ohne diesen aus.

Techniken

Versionierung und Projektkommunikationssoftware

+	Alle: git
	Roboter: Issue Tracking versucht, aber nicht dauerhaft benutzt.
	Tour2Go: Bitbucket für Issue Tracking usw. angefangen, aber nicht dauerhaft benutzt.
	Kommunikation über WhatsApp + E-Mail
	Terminverwaltung: gitlab-Issue-Verwaltung vor Endstress benutzt. Nach Rücksprache Arbeitsaufträge als Issues erfasst.
	Branching? Tour2Go: Nur auf Master-Branch mit klarer Arbeitsteilung. Terminverwaltung: Intensiver Branching-Einsatz für Issues. Schichtspezifische Branches.

Einsatz von JPA bzw. Hibernate (OR-Mapper)

!	Es wäre gut, ein Referenzbuch zum Vorgehen empfohlen zu bekommen.
!	Objektzustandsentwicklung, z.B. detached, unklar. Update-Operationen von verschachtelten Objekten sind nicht trivial.
!	Knabe: Habe immer unidirektionale Assoziationen von viele zu eins empfohlen, aber kein

	Team hat sich daran gehalten! (Ich hab's Ihnen ja gesagt.)

Einsatz eines Web-Frameworks / UI-Frameworks

	WG-App: jQuery mobile: sehr gut.
	Tour2Go: Android: Viele Lösungsmöglichkeiten für ein Problem. Beispiele meist trivial. Hoher Einarbeitungsaufwand. Erst spät eigenes Vorgehen mit vielen Activities als „Bad Practice“ erkannt. Dann Umstellung auf Fragments nötig.
	Terminverwaltung: Angular.js: Zufrieden, aber lernaufwändig. Einfache Sachen einfach, aber fortgeschrittene Sachen sehr schwierig.

Einsatz von MulTEx

Einsatz von JUnit-Testtreibersuite

+	Terminverwaltung: War sehr positiv, hätte man noch mehr machen können. War ein wichtiges Werkzeug, um auslieferbares Produkt zu erzeugen.

Einsatz von

	Build-Werkzeug: Roboter: Praktisch, da alles durchgeneriert wird. Terminverwaltung: Praxisrelevant.

Studiengang
