

Abschlussbesprechung Software-Projekt I (Kurzanalyse + Techniken) bei Prof. Knabe im SS 2014

Hier die in der Abschlussbesprechung am 24.07.2014 geäußerten Meinungen der Studierenden, gegliedert nach Themengebieten und Meilensteinen.

Legende:

+ positiv bewertet, – negativ bewertet, ~ teils/teils, ! Wunsch

Meilensteine

V2 Teambildung, Projektstudie, Teamprofil

--	--

V3 Zielbestimmung (Muss, Soll, Kann)

~	Eine Gruppe: War nicht viel Arbeit; WG-App: Doch.
---	---

V4 Fachklassendiagramm: Klassennamen, Attribute, Assoziationen, Kardinalitäten, Technik-Einarbeitungsplan

	Technik-Einarbeitungsplan: WG-App: Mussten diesen Plan später wegen Umfangs revidieren. Tour2Go: Plan war gut. Roboterspiel: Auch mehrere auf dieselbe Technik ansetzen! Muss mit Maven funktionieren! Keyseeker: GWT war nicht gut auf verschiedene Personen aufteilbar. Knabe: Personenzuordnung nicht so wichtig wie Festlegung der zu lernenden Techniken.
--	--

V5 Operationsnamen im KD, git-Leerprojekt

!	Zu git im Unterricht Empfehlung geben: Was versionieren, was nicht.
!	Paketdiagramm aus Abnahme früher einplanen!

V6 Technik-Prototypen

+	Meilenstein an sich gut.
!	Es sollte schon auf später nötige Fassaden hingewiesen werden, um Umbau zu vermeiden.

V7 Build-Werkzeug mit Testsuite darin und in IDE

!	Wäre besser noch vor V6 „Technik-Prototypen“, damit man gleich mavenisierte Tutorien sucht.
!	Ausführen <code>mvn package</code> auf Konsole! (Frage Knabe: Was war damit gemeint?)

V9 Objektverwaltung-UI

-	Tour2Go: Erklären, dass CRUD gemeint ist. Bottom-Up (erst Persistenzschicht) wäre günstiger für Entwicklung.
!	Auf untenstehende Meilensteine Definitionen verweisen!
!	Bei jedem Meilenstein aufführen, was abgenommen wird, z.B. immer Build und Testsuite.

V11 Persistenzschicht, Technikfassaden inklusive Testsuite+Build

?	Überlegen: Objektverwaltungs-UI und Technik-Fassaden inkl. Persistenz parallel!?
---	--

V13 Objektverwaltung, Technikeinbindung inklusive Testsuite+Build

!	Als Vorabnahme kennzeichnen!
!	Transaktionsmanagement in einen Meilenstein aufnehmen! Orientierung geben, nach welchem Muster man Transaktionsmanagement machen soll.
?	Funktioniert DynamicProxy mit @REST-Jersey?

14 Abnahme

!	Abnahmekriterien früher bekannt geben!
---	--

!	Kriterienformular: Statt „Vorführen“ als dozentengesteuerte Abnahme formulieren.
!	Hinweisen: In Abnahme sind alle bisherigen Meilensteine enthalten.

Meilensteine

	Zeitlicher Rhythmus OK?
--	-------------------------

Eingesetzte Techniken

!	Projektgruppen benötigen je eigenen virtuellen Debian-Root-Server!
	1 WG-App: jQuery mobile + Phonegap, Jersey, Jasmine, RestAssured, Hibernate, Maven?
+	JQuery mobile war super.
	2 KeySeeker: GWT-GUI, GWT-RPC, Hibernate, Maven
~	GWT: Theorie sehr gut, Probleme mit Plugins, für Firefox gibt es kein Google Developer Plugin mehr, viele Kompatibilitätsprobleme.
+	GWT-RPC klappte gut, jedoch Probleme mit Plugins.
	4 Tour2Go: Android SDK, Google Maps, Jersey, Gson, JPA/Hibernate, Gradle
+	Android SDK: OK. Benutzten Android Studio, β-Version war etwas schwierig, Nach Update auf neuere Version schwierig.
	5 Terminverwaltung: AngularJS+Phonegap, Jersey, JPA/Hibernate, Maven
+	AngularJS: Gut waren scope, double data binding, eigene Services festlegen, modularisieren.
-	Jasmine-Tests für Angular-Module nicht geschafft.
	6 Roboterspiel: HTML/CSS/JS, com.sun.net.httpserver.HttpServer, JPA/Eclipselink, Maven
+	Gson für Konvertierung Java ↔ JavaScript war gut.
+	Eigenbau-HTTP-Server war interessant und ausreichend.

Unterricht

--	--

Rücksprachen

--	--

Notengebung

	Gerecht im Vergleich zu anderen?
--	----------------------------------

Gruppenarbeit

!	Knabe: Ein Team ist geplatzt. Wie üblich wegen unterschiedlichen Arbeitseinsatzes. Beide Teile müssen dann das geplante System alleine zu Ende bringen.
---	---

Studiengang Medieninformatik

--	--