

Weekly Meeting - 13.03.2013

Teilnehmer

- HSR/IFS
 - Hans Rudin (HRU)
 - Kevin Gaunt (KGA)
- SA Team
 - Michael Weibel (MWE)
 - Alexandre Joly (AJO)
 - Manuel Alabor (MAL)

Organisation

- Leitung: AJO
- Protokoll: MAL

Agenda

- Letztes Protokoll
- Projektplanung
- Guidelines
 - Coding Guidelines
 - Testing Guidelines
- Software Architecture Document
- Domainmodel & ER Diagramm
- Anforderungsdokumentation
 - funktional & nicht funktionale Anforderungen
- Use Cases
- Architekturprototyp

- Offene Fragen
- Ausblick, nächste Schritte

Letztes Protokoll

- Ist i.O.
- Beschlüsse in einem Abschnitt zusammenfassen

Projektplanung

- Phasenplanung
 - Erklären warum Inception & Elaboration überlappen
- Milestones
 - KGA: Bezeichnung Milestones könnten sprechender sein

Guidelines

- AJO: JavaScript Guideline von Airbnb
- AJO: Testing Guideline bis Ende Woche
- KGA: Warum Airbnb Guideline?
 - MAL: Verbreitet in Community, schon länger Bestand

Anforderungsdokumentation

Funktionale & Nichtfunktionale Anforderungen

- HRU: Einleitung für Requirements
- KGA: F5: WG verlassen statt nur "Verlassen"
- MAL: NF5: Nicht nur ROCA sondern auch Tilkov Erwähnen

Use cases

- HRU: Mapping auf Requirements machen
- AJO: Muss Registrieren & Login auseinandergenommen werden?
 - HRU: Nein, Fallunterscheidung kann im Use Case geschehen

- AJO: Erläuterung wie "Invite with Link" funktioniert
 - HRU: Expliziter erklären dass die Links ausserhalb des Systems verteilt werden
- AJO: Kann "UC10 Benutzer verwalten" so zusammengefasst werden? Enthält ja CRUD & Rechtevergabe
 - HRU: Kein Problem, einfach im Use Case wieder per Fallunterscheidung erwähnen
 - HRU: Casual Style wählen --> wikipedia suche von gestern ;)
- HRU: Es muss klar sein, wie der Link zur Einladung verteilt/kopiert werden kann
- AJO: Aktoren
 - HRU: Facebook fehlt
- HRU: Neue Use Case Formulierung ist gut so
- AJO: Muss extend-Use Case einzeln beschrieben werden?
 - HRU: Ja

Technische Architektur

Domainmodel

- AJO: Deutsche/Englische Begriffe
 - HRU: Übergänge deutsche/englische Begriffe klar definieren
 - HRU: Konzepte Beschreiben
- AJO: Erklärung Rollenkonzept
 - AJO: Admin-Rolle wird als nice to have klassifiziert
 - AJO: Resident & CommunityAdmin sollen umgesetzt werden
 - AJO: User, welcher Community erstellt, wird automatisch CommunityAdmin
 - HRU: Problematik Rollenmodellierung
 - Link Security Patterns: <http://swissmanu.github.com/hsr-apf-2013/patterns.pdf>, Abschnitt 1.2
 - KGA: Naming "CommunityAdmin" unglücklich... Was hat er mit dem normalen Admin zu tun? Besser "WG Verantwortlicher"

- MAL: Pragmatischerer Vorschlag für Rollenkonzept
- AJO: Erläuterung Achievement-System
 - HRU: Braucht es Verbindung zwischen Rule <-> Task?
- HRU: Domainmodell braucht Attribute!

ER Diagramm

- HRU: Multiplicity RoleToUser <-> User falsch
- AJO: Überspringen da Dank neuen Erkenntnissen überholt

Software Layers

- Business Layer:
 - Businesslogic:
 - HRU: Validation gehört eher in Models...?
 - HRU: Was ist API?
 - MWE: Service API, in unserem Falle REST
 - HRU: API/Controllers: Wirklich auf gleicher Ebene?
- MAL: Presentation Layer für Server & Client differenzieren
- KGA: Layerdiagramm sollte Logging & Security nicht als einzelne "Cross Cutting" Komponente darstellen
- HRU: "Tiefere" bitte in UML Notation
 - Komponentendiagramme
 - Sequenzdiagramme für Kommunikation

Architekturprototyp

- MWE: Warum kein Sails?
 - Hauptgrund: ORM unbrauchbar (keine Relations etc.)
 - Testability kein Thema im Moment
 - Schlechte Dokumentation
- MWE: Entscheid am letzten Freitag:
 - express.js

- HRU: Erklären warum kein Sails sondern nun express verwendet wird
- MWE: Warum JugglingDB als ORM?
 - Relations funktionieren
 - Viele DB-Adapter vorhanden
 - Weniger optimal im Moment: PostgreSQL Adapter erstellt keine Foreign Keys
- MWE: Erläuterung "Passport"
 - Authentication mit verschiedensten Identity Providers (Facebook, Twitter, Google, ...)

Offene Fragen

- Urheberrechtsvereinbarung nötig?
 - HRU: Ja, machen wir um einfach sicher zu sein

Ausblick

- Elaboration abschliessen: Siehe Beschlüsse

Varia

- KGA: InfoQ Vortrag Video Mitschnitt von Stefan Tilkov ist interessant!

Beschlüsse

- Projektplanung
 - Kurz Sachverhalt erklären, warum Phasen überlappen
 - Milestones sprechender benennen sobald Inhalt fix ist
- Technische Architektur
 - Zusätzlicher Abschnitt "Node.JS & express.js"-Einführung im Kaptiel technische Dokumentation
 - inkl. Erklärung warum kein Sails sondern nun express verwendet wird
 - Rollenkonzept "pragmatisieren"

- Attribute im Domainmodell hinzufügen!
 - Achievementmodellierung "pragmatisieren"
 - Komponentendiagramme für genauere Abhängigkeiten (spätere Projektphase)
 - Sequenzdiagramme
 - ERM gem. neuem Domainmodell überarbeiten
 - Layerdiagramm sollte Logging & Security nicht als einzelne "Cross Cutting" Komponente darstellen
- Transition zwischen Sails & express.js erklären/verkaufen