

Protokoll BA Meeting 06.10.14

Datum, Zeit	Mo 06.10.14., 13:00 - 14:30
Ort	Rapperswil, HSR, 6.111
Teilnehmende	Olaf Zimmermann, Tobias Blaser, Laurin Murer, Lukas Wegmann
Protokoll	Tobias Blaser, Laurin Murer

Traktanden

1. Aufgabenstellung abschliessen
2. Bestehender Server CDAR -> Betrieb, Abschaltungstermin Frühling 2015
3. Änderungsnachweis im technischen Bericht: Git Log?
4. Feedback "Architektur Draft"
5. REST-Schnittstelle zu CDAR

Aufgabenstellung

Wurde unterschrieben und von den Studenten eingescant.

Server CDAR

- Server 80.30 gehört dem Institut und wird nicht abgeschaltet

Änderungsnachweis Technischer Bericht

Entfernen

Feedback Architektur

- wenn kein Kommentar -> gut
- liest sich sehr gut, z.T. kommen noch Umgangssprachliche (UGS) Sätze vor
- Weiterer Input: <http://www.ifs.hsr.ch/Technical-Writing-Advice.13194.0.html>
- Konsistenz: Für Englische Begriffe auch Deutsche Grossschreibung verwenden
- Leerzeichen vor jeder Referenz (also nicht: Buch[3] sonder Buch [3])
- Kunde überall ersetzen durch "Vertreter der Anforderungsgruppe"

Anforderungen

- Ein Satz vor dem Unterabschnitt einfügen
- Einschränkungen: Schreiben, woher das fehlende Wissen kommen kann/sollte (<http://www.ifs.hsr.ch/Cloud-Knowledge-Sources.12060.0.html>)

- Personas: Präzisere Angaben (nicht gut oder viel sondern konkrete Zahlen), Thomas: Projektplaner oder technische Führungskraft? Projektgröße beschreiben (KMU?, 3-5 Entwickler)
- Story 23 zu viel UGS ("...weiter zu kommen...")
- Anpassbarkeit: "keine Plattformspez. Config" statt "ausergewöhnliche Config"
- NFA Portability->Anpassbarkeit: nur auf Betriebssystem beschränken, Testinstallation bei Webhostern machen (--> Verschieben in Testing-Dokumentation)

Architektur/Technologie

- Template Motivation: Keine Quality Attributes sondern warum die Entscheidung wichtig ist
- Zusätzlicher Grund für Play: Experte im IFS (Mirko Stocker)
- Patterns unbedingt benennen! (--> sieht besser aus)

REST-Schnittstelle zu CDAR

- Kein Zurückschreiben von Daten von EEPPI zurück zum CDAR -> Interessenten wie EA sollen die EEPPI API nutzen.
- gewünschte Data-Source API für EEPPI definieren (test driven -> API- Vorschlag machen). Datengrundlage: JSON
- Authentifizierung in EEPPI machen (wenig Zeit zu investieren)

TODOs

Studenten:

- Definieren was die Data-Source API liefern muss

LW:

- Baut eine Demo-API auf

ZIO:

- Stellt ein Paar Folien zusammen zum EA Plugin und schickt sie den Studenten.

Beschlüsse

- Versionshistorie ist bei Abgabe nicht mehr nötig, während der Arbeit reicht Git Log (externer Zugriff, nicht im Dokument)
- Security ist KEINE Anforderung (sollte grundlegende Funktionen enthalten, hat aber keine Priorität)