

**SEPM Gruppe 3950****[[review:requirements]]****REVIEW REPORT: REQUIREMENTS****Review Bereich:** Requirements**Reviewer:** Simon**Review Dauer ca./Feinheit:** 3h/eher grober, struktureller Überblick**erstellt am:** 20.10.2008, 19:45**spezieller Fokus:** Erhebung des status quo, insbesondere Beachtung der Vollständigkeit der Use Cases und der Requirements Dokumentation. Entdecken von hidden requirements. Feststellung des Umfangs der bisherigen Requirements**Liste der gefundenen Punkte**

<b>Name</b>	<b>Beschreibung</b>	<b>Schwere/Seiteneffekte</b>	<b>Aufwand zur Behebung</b>
<b>Dokument Features</b>			
project group	Die erwähnte project group ist nicht genau definiert	Implizite Annahme, scope creep bei unterschiedlichen Auffassungen	gering, keine Nebeneffekte
data pool	siehe oben...		
meta data vs. categories	Es werden metadaten und Kategorien erwähnt, diese aber auch nicht weiter spezifiziert	siehe oben ...	
allgemein	Die Features allgemein sind recht vage gehalten	kein unmittelbares Problem für die Entwicklung solange die restlichen Requirements stimmen	gering
<b>Dokument Functional Requirements</b>			
allgemein	Die functional requirements sind so wie die features sehr vage gehalten. Die requirements werden nur mir sehr allgemeinen Worten beschrieben. Es sollte vielmehr eine genaue Spezifikation der Anforderungen sein, denn eine erweiterte Featureliste.	Die fehlende Genauigkeit ist durchaus als kritisch zu betrachten und kann zu erheblichen scope creep führen. Sie beinhalten sehr viele implizite Information, die explizit gemacht werden muss	10-20h
<b>Dokument Nonfunctional Requirements</b>			
allgemein	Die Kern-nfr werden ausreichend beschrieben, es fehlt jedoch die Messbarkeit der nfr vor allem im Bezug auf Einfachheit, etc.	gering, Die für das Projekt kritischen nfr (keine data corruption, ...) sind ausreichend.	2-5h
<b>Assumptions</b>			
Diskussion	Zum Teil finden sich in den Assumptions Diskussionen, die dort aber nicht hingehören	Aufweichung der "fixen" assumptions	gering
Unsortiert	Die Assumptions sind unsortiert und nicht strukturiert.	wichtige assumptions zu einem Gebiet könnten übersehen werden	gering-mittel
allgemein	Die Assumptions sind sehr gut und vor allem sehr spezifisch und lassen wenig Raum für Interpretationen. Der Großteil der Assumptions sollte aber direkt in den requirements stehen.	Es könnte passieren, dass Information mehrfach redundant in der Doku vorkommt, bzw. die assumptions die requirements ablösen → Einarbeitung der assumptions in die requirements	5-10h

Dokument <b>Use Cases</b>			
Übersicht	Die Übersicht ist ganz OK, Die Zuteilung zu den einzelnen Gruppen macht einigermaßen Sinn		
open Issue	Jeder Use Case hat eine open Issues section. Imho gehören die use cases zu den statischen Dokumenten, die, einmal erstellt, nicht mehr geändert werden, daher macht auch eine open issues section wenig Sinn.	Vermischung statischer und dynamischer Dokumente, Konsistenzprobleme mit anderen Dokumenten	gering
Ungenauigkeiten	Manche use cases sind nicht genau spezifiziert. Z.B. "list all notes". Es wird nicht spezifiziert welche Daten genau angezeigt werden sollen, oder in welcher Sortierung etc.	Die Ungenauigkeit kann zu versteckten Annahmen führen, was zu inkonsistentem Verhalten führen kann.	mittel
Exceptions	Es werden bei fast keinem use case exceptions angegeben. Diese Feld ist meist leer.	Fehler in der Modellierung, wodurch das Modell an Aussagekraft verliert	mittel
soft lock	In vielen use cases wird auf den soft lock verwiesen, der aber nie implementiert wurde.	Inkonsistent mit der Implementierung, Aus der Sicht der use cases nicht unbedingt problematisch, nur sollte man sich darauf einigen, ob man implementiert oder nicht. Übereinstimmung mit dem Requirements	gering
Abstraktionsebene	Die use cases beinhalten zum Teil viele implementierungsspezifische Details (e.g Synchronisation, etc.)	Durch die unsaubere Abstraktion könnte das Modell aufgeweicht werden. Vor allem eine Vermischung/Überschneidung mit den functional requirements könnte problematisch sein.	hoch; Neuaufteilung, Bereinigung der Überschneidungen

### **Zusammenfassung des Reviews**

Es gibt keine richtige SRS (Software Requirements Specification). Die Requirements sind sehr allgemein, und auch nicht ausreichend. Sie beschreiben das Systemverhalten nicht annähernd ausreichend. Die Use Cases sind sehr umfangreich und erscheinen auch vollständig, doch beinhalten sie einige Implementierungsdetails, die besser in den Requirements aufgehoben wären. Requirements und use cases haben eine sehr verschwommene Grenze und überschneiden sich auch zum Teil.

Die Assumptions sind sehr detailliert und entsprechen noch am ehesten einer genauen Spezifikation. Sie sollte aber als Assumptions aufgelöst werden und in die requirements integriert werden.

### **Gesamtzustand**

Der Gesamtzustand ist nicht so gut. Das, was vorhanden ist, ist nicht so schlecht, nur vom Umfang her zu gering und oft zu allgemein. Im Hinblick auf bessere Kontrolle sollte dieser Bereich stark verbessert werden.

### **Aufwand der Verbesserung**

10-30 Stunden, Neustrukturierung der SRS, überarbeiten, vervollständigen.

### **abschließende Kommentare**

Eine stärkere Trennung der einzelnen Bereiche wäre hilfreich.

---

*review/requirements.txt · Last modified: 2008/10/22 14:13 by johannes*