

Software-Projekt 2 2014

VAK 03-BA-901.02

Anforderungsspezifikation

Schibboleth



Patrick Hollatz	phollatz@tzi.de	2596537
Tobias Dellert	tode@tzi.de	2936941
Tim Ellhoff	tellhoff@tzi.de	2520913
Daniel Pupat	dpupat@tzi.de	2703053
Olga Miloevich	halfelv@tzi.de	2586817
Tim Wiechers	tim3@tzi.de	2925222

Abgabe: TT. Monat JJJJ — Version 1.1

Inhaltsverzeichnis

1	Einleitung (Patrick)	3
1.1	Zweck	3
1.2	Rahmen	3
1.3	Definitionen, Akronyme und Abkürzungen	3
1.4	Referenzen	4
1.5	Übersicht über das Dokument	4
2	Allgemeine Beschreibung	4
2.1	Ergebnisse der Ist-Analyse (Tim E.)	4
2.1.1	Erstes Kundengespräch vom TT.MM.JJJJ	4
2.1.2	Interview mit einem Mitarbeiter der	4
2.1.3	Analyse eines ähnlichem Systems (Daniel Pupat)	5
2.2	Produktperspektive (Tim W. und Tim E.)	15
2.2.1	Systemschnittstellen	15
2.2.2	Benutzerschnittstelle	15
2.2.3	Hardwareschnittstellen	15
2.2.4	Softwareschnittstellen	15
2.2.5	Kommunikationsschnittstellen	15
2.2.6	Speicherbeschränkung	16
2.2.7	Operationen (Betriebsmodi)	16
2.2.8	Möglichkeiten der lokalen Anpassung	16
2.3	Anwendungsfälle (Daniel)	16
2.4	Charakteristika der Benutzer	18
2.5	Einschränkungen (Tim W. und Tim E.)	18
2.5.1	Rahmenbedingungen	19
2.5.2	Gesetzliche Rahmenbedingungen	19
2.5.3	Sicherheitskritische Aspekte	19
2.6	Annahmen und Abhängigkeiten (Tim E.)	19
2.7	Ausblick (Tim E.)	19
3	Detaillierte Beschreibung	20
3.1	Datenmodell (Tobias)	20
3.2	Anwendungsfälle (Tobias und Daniel)	20
3.3	Aktionen	21
3.4	Entwurfseinschränkungen	22
3.5	Softwaresystemattribute	22
3.6	Weitere Anforderungen	22
4	Anhang	22

Version und Änderungsgeschichte

Die aktuelle Versionsnummer des Dokumentes sollte eindeutig und gut zu identifizieren sein, hier und optimalerweise auf dem Titelblatt.

Version	Datum	Änderungen
1.0	21.05.2014	Analyse ähnlicher Systeme.
1.1	22.05.2014	Anwendungsfälle .

1 Einleitung (Patrick)

Dieses Dokument dient als Vorlage für Eure Anforderungsspezifikation. Die Gliederung dieses Dokuments ist an die Struktur des IEEE-Standards 830.1998 angelehnt, weicht jedoch an einigen Stellen davon ab. Die Abweichungen sind im weiteren Verlauf dieses Dokuments dokumentiert. Weitere detaillierte Hinweise finden sich im IEEE-Standard 830.1998, der in Stud.IP beziehungsweise über die Uni-Bibliothek in digitaler Form verfügbar ist ¹.

1.1 Zweck

Was ist der Zweck dieser Anforderungsspezifikation? Wer sind die LeserInnen?

1.2 Rahmen

Dieser Abschnitt soll einen groben Überblick über die zu erstellende Software geben: Welche Produkte sind zu erstellen (mit Namen)? Was tut die Software? Auch: Was tut sie nicht? Wozu soll die Software verwendet werden? (Ziele etc.)

1.3 Definitionen, Akronyme und Abkürzungen

Hier geht es vor allem um Begriffe aus der Anwendungsdomäne, d.h. aus der Welt des Kunden. Aber auch Begriffe, die dem Kunden evtl. fremd oder unklar sind, sollten erläutert werden.

¹Bei <http://ieeexplore.ieee.org> im Suchfeld 'IEEE std 830-1998' eingeben. Funktioniert nur innerhalb des Uni-Netzes.

1.4 Referenzen

Neben sonstigen Quellen, die Ihr verwendet habt, können dies z.B. das Skript, dieses Beispieldokument, der zugrunde liegende IEEE-Standard und anderes sein

1.5 Übersicht über das Dokument

Was enthält die Anforderungsspezifikation? Wie ist das Dokument organisiert?

2 Allgemeine Beschreibung

2.1 Ergebnisse der Ist-Analyse (Tim E.)

Hier sollten die Ergebnisse Eurer Ist-Analyse kurz zusammengefasst werden. Diese Beschreibung ist hilfreich, um die Motivation für die Anforderungen zu verstehen und um sie später nachzuvollziehen (z.B. dann wenn Anforderungen überarbeitet werden sollen, weil sich ihre Rahmenbedingungen geändert haben).

Mögliche Inhalte:

- *Interview/Beobachtung des Kunden oder der Benutzer*
- *Analyse des bisherigen Systems und dessen Probleme*
- *Analyse ähnlicher Systeme*
- *Auswertung der Benutzerbefragung*
- *Wie sollen die identifizierten Probleme vom neuen System adressiert werden?*

N.B.: Dieser Abschnitt ist im IEEE-Standard nicht vorgesehen, aber dennoch sinnvoll.

2.1.1 Erstes Kundengespräch vom TT.MM.JJJJ

2.1.2 Interview mit einem Mitarbeiter der ...

Falls durchgeführt

2.1.3 Analyse eines ähnlichem Systems (Daniel Pupat)

Als System haben wir uns die App Quizduell ausgesucht. Die App ist im App-Store kostenlos zu erhalten. Wir benutzen hierzu nicht die Premium-Version, da die kostenlose unserem System am nächsten kommt. Wir haben nicht die Rechte eines Administrators, weshalb sich die Analyse nur auf die App bezieht. Die App kann nur verwendet werden, wenn man eine bestehende Internet Verbindung hat.



Abbildung 1: Ladebildschirm

Dies ist der Ladebildschirm, sobald die App gestartet wird.



Abbildung 2: Startbildschirm

Hier ist der Startbildschirm, nachdem man eingeloggt ist. Oben hat man die Funktionen Aktualisieren, Statistiken und Einstellungen. Beim aktualisieren wird die Seite aktualisiert.



Abbildung 3: Statistiken

Hier kann man die Rating Liste aller Nutzer in Deutschland im Bezug auf der Punktzahl und der Anzahl der geschrieben und angenommenen Fragen sehen. Für die eigene Statistik wird eine Premium-Version benötigt.



Abbildung 4: Einstellungen

Unter Einstellungen kann man seine Kontoeinstellungen bearbeiten, wo man den Benutzernamen und Passwort ändern kann. Für Avatar gestalten und Farbschema ändern benötigt man die Premium-Version, welche man unter PREMIUM kaufen erwerben kann. Bei Fragen schreiben, kann man eigene Fragen schreiben, die evtl. dann übernommen werden. Bei Hilfe werden bestimmte Fragen im Hinblick auf die Benutzung beantwortet.



Abbildung 5: Startbildschirm

Weiter Funktionen beim Startbildschirm sind Neues Spiel starten oder ein bereits angefangenes Spiel weiterspielen.



Abbildung 6: Bildschirm wenn man ein neues Spiel starten möchte

Wenn man ein neues Spiel starten möchte, kann man einen neuen Spieler suchen, unter Facebook Freunden suchen, falls man unter Facebook eingeloggt ist, oder gegen einen beliebigen Spieler spielen, der ebenfalls einen gesucht hat.



Abbildung 7: Bildschirm beim neuen oder laufenden Spiel

Hier wird der bisherige Spielstand angezeigt. Insgesamt gibt es 6 Fragen, welche jeder beantworten muss. Die Spieler erhalten jeweils die gleichen Fragen. Es werden immer 2 Fragen abwechselnd gestellt, wobei der Spieler einmal eine Kategorie auswählen darf und einmal die Fragen der Kategorie des Gegners beantworten muss. Die Spieler haben 48 Stunden Zeit die Fragen zu beantworten bis das Spiel automatisch, mit einer Niederlage des Spielers der nicht geantwortet hat, endet.

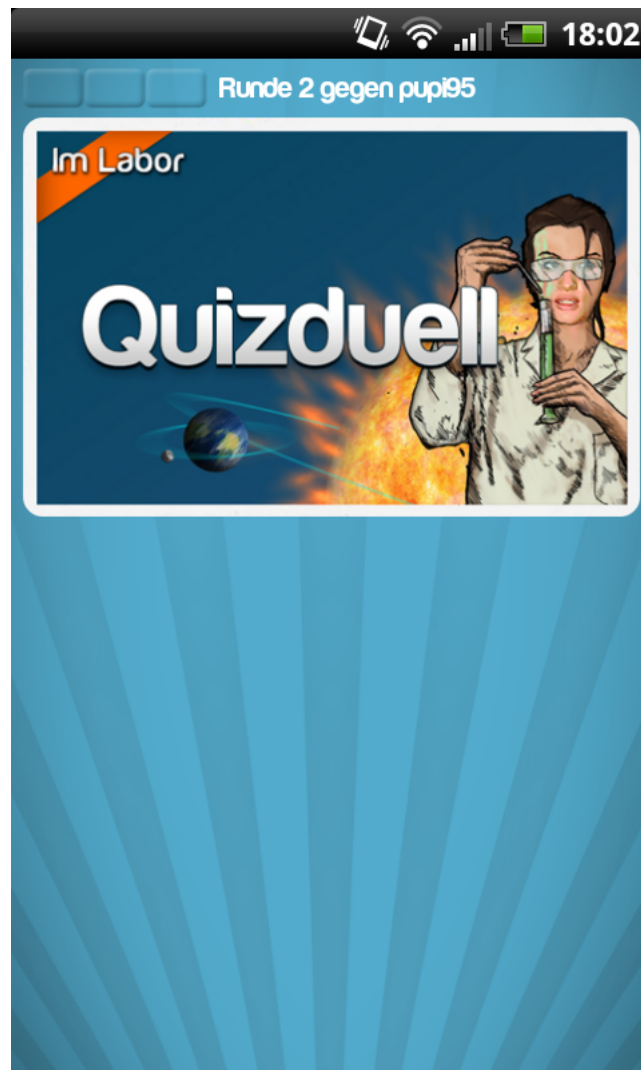


Abbildung 8: Beim Starten der Fragen

Beim Starten einer Frage erscheint erst nur eine Karte mit der Kategorie, bei einer Berührung wird die Karte umgedreht.

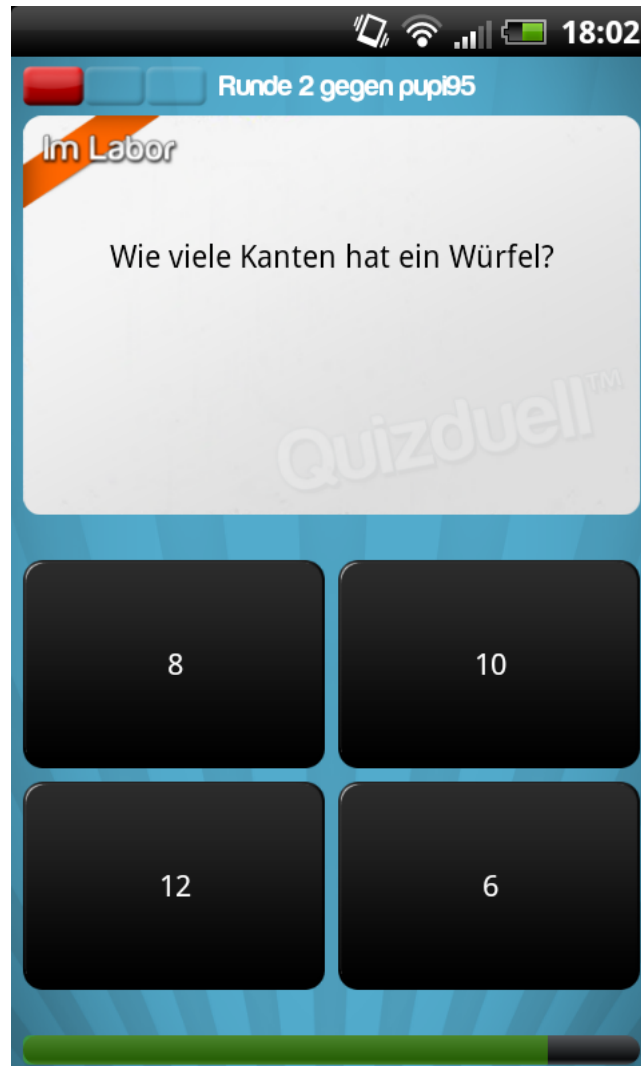


Abbildung 9: Anzeigen der Frage

Nun wird die Frage angezeigt, man hat 20 Sekunden Zeit um die Frage zu beantworten, sonst wird diese als Falsch gewertet.

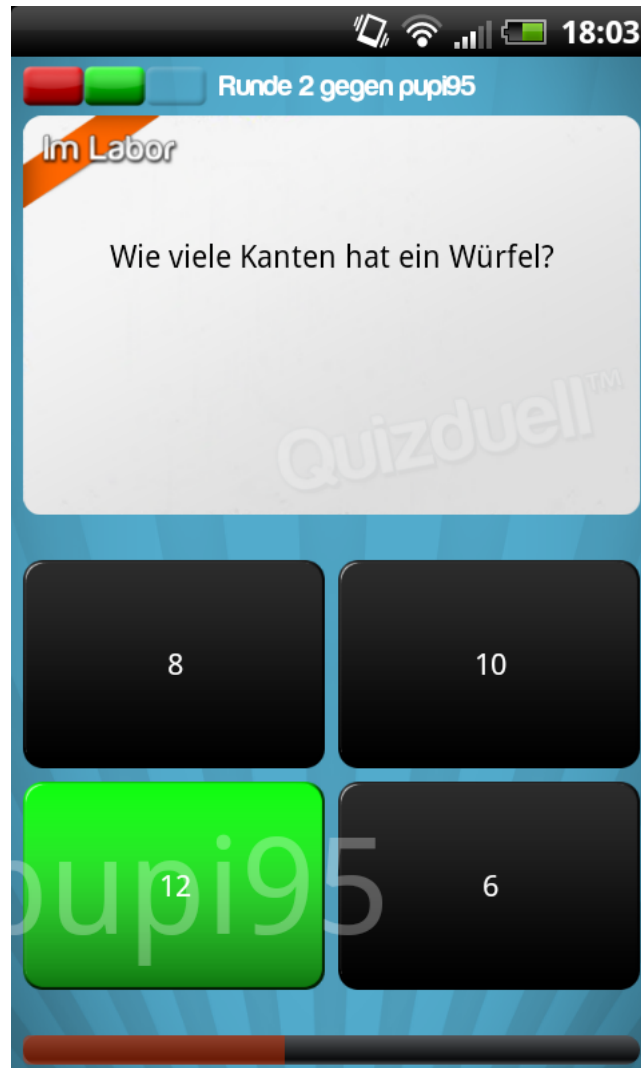


Abbildung 10: Reaktion bei richtiger Antwort

Bei einer richtigen Antwort wird diese mit Grün unterlegt und Zeitgleich wird der Name des Gegners auf seiner Antwort angezeigt, wenn er diese bereits beantwortet hat. Bei einer falschen wird die Antwort mit Rot unterlegt.

Fazit:

Die Quizduell App ist ein gutes System mit welcher man unseres später Vergleichen kann. Es hat viele Funktion, wie gezeigt, welche unser System ebenfalls haben muss. Es gibt allerdings auch Unterschiede, in unsere App wird es nur 3 Runden pro Runde geben und es werden keine eigenen Fragen geschrieben, außerdem wird es keine Premium-Version geben. Zudem werden wir auch eine offline-Funktion einbauen, in welcher der Spieler alleine spielen kann.

2.2 Produktperspektive (Tim W. und Tim E.)

2.2.1 Systemschnittstellen

Schnittstellen zu anderen Systemen, z.B. Datenimport/-export, Konfigurationsdateien, anzubindende externe Dienste und deren Schnittstelle, Anbieten der eigenen Funktionalität als API o.ä.

Als Grundlage dient ein neu aufzusetzendes oder bestehendes System, auf dem die Software installiert werden kann. Als externe Schnittstelle muss dieses mit dem Internet verbunden sein. Der Server der Quizapp kann an einem PC über HTTP mit einem HTML5-unterstützenden Browser angesprochen werden. Mobiler Zugriff erfolgt auf Android-Geräten zudem mit einer eigenen App.

2.2.2 Benutzerschnittstelle

GUI-Design-Richtlinien und Interaktionsmechanismen (nicht Screenshots aller Dialoge — die werden in Kapitel 3 gezeigt — aber evtl. ein Screenshot, der einen groben Überblick und Eindruck des GUI-Designs gibt). Die GUI der Website weist je nachdem, ob sich ein normaler Nutzer oder ein Admin anmeldet, unterschiedliche Funktionalitäten auf.

2.2.3 Hardwareschnittstellen

Schnittstellen zu vorgegebenen Hardwarekomponenten (Name, Version).

Die HTML-Website ist plattform-unabhängig nutzbar. Smartphones mit kompatibler Android-Version (welche?) können die Quizapp nutzen.

2.2.4 Softwareschnittstellen

Softwarebibliotheken und -rahmenwerke (Frameworks), die benutzt werden sollen, mit Versionsnummer, Hersteller, Quelle etc. Dazu gehört auf jeden Fall Java.

Name	Version	Hersteller	Quelle
Java Runtime	6 Update 37	Oracle	http://java.com
JUnit			
libGDX			
Android			
...			

2.2.5 Kommunikationsschnittstellen

Anforderungen an und Bandbreite von Kommunikationsnetzwerken, öffentliche oder auch private IP-Adressen?

Das System muss mit einer öffentlichen IP-Adresse und Domain von außen ansprechbar sein. Die Bandbreite sollte groß genug sein um mehreren Nutzern parallel den Zugang zur Software zu gewähren, wobei jeder Nutzer einzeln nur eine sehr geringe Bandbreite benötigt.

2.2.6 Speicherbeschränkung

min./max. verfügbarer Hauptspeicher und Festplattenplatz, knappe Begründung wie Ihr zu der hier angegebenen Einschätzung gekommen seid

Die App wird voraussichtlich bis zu 30mb Speicherplatz beziehen. (Arbeitsspeicher?)
Das Softwaresystem

2.2.7 Operationen (Betriebsmodi)

Welche Betriebsmodi gibt es? Warum? Welche Benutzerklasse darf was in welchem Betriebsmodus (Rechte)? Was ist der Zusammenhang zwischen Betriebsmodus und Sicherung/Wiederherstellung von Daten?

Wir unterscheiden zwischen dem normalen Betriebsmodus und einem Wartungsmodus, in dem z.B. Backups aufgespielt werden können. Während des Wartungsmodus ist kein öffentlicher Zugriff vorgesehen. Im normalen Betriebsmodus wird unterschieden zwischen Admins, registrierten Benutzern und Gästen. Jeder hat das Recht zu spielen, wobei registrierte Nutzer Vorteile wie Statistiken etc. genießen. Der Admin kann das System über die Website verwalten, z.B. Nutzernamen ändern. In der App ist kein Admin-Modus vorhanden.

2.2.8 Möglichkeiten der lokalen Anpassung

Was kann bei Auslieferung des Systems alles konfiguriert werden? Z.B. Pfade, Datenbankname, Server-IP usw. Hier ist nicht Internationalisierung gemeint! Es handelt sich bei dem System um ein lauffähiges Gesamtpaket. Es muss keine Datenbank o.Ä. eingerichtet werden. Lediglich eine IP-Adresse muss eingerichtet werden. Somit ist keine lokale Anpassung nötig.

2.3 Anwendungsfälle (Daniel)

- **1. App starten**

Die App wird gestartet.

- **2. Offline-Modus starten**

Die App wird im Offline-Modus gestartet. Keine Anmeldung und Internet Verbindung notwendig.

- **3. Benutzer registrieren**
Benutzer registriert sich mit Benutzernamen und Passwort
- **4. Benutzer anmelden**
Der Benutzer meldet sich mit Benutzernamen und Passwort an. Die Daten werden gespeichert, sodass kein nochmaliges anmelden nötig ist.
- **5. Benutzer abmelden**
Benutzer wird abgemeldet und muss sich beim nächsten Start wieder anmelden.
- **6. Online Modus starten**
Der Online Modus wird gestartet. Einmaliges anmelden und Internet Verbindung erforderlich.
- **7. Neues Spiel starten**
Eine Liste aller Spieler die online sind wird angezeigt.
- **8. Gegner herausfordern**
Aus der Liste kann ein Spieler angeklickt und herausgefordert werden.
- **9. Herausforderung annehmen**
Bei einer Herausforderung kann der Gegner annehmen oder ablehnen.
- **10. Spielrunde starten**
Eine neue Spielrunde wird gestartet, die erste Frage erscheint.
- **11. Frage beantworten**
Der Spieler hat 20 Sekunden Zeit zu antworten.
- **12. Einstellungen ändern**
Unter Einstellungen kann der Benutzername und das Passwort geändert werden
- **13. Punktzahl anzeigen**
Rangliste mit den Punktzahlen aller Mitspieler wird angezeigt
- **14. Website wird aufgerufen**
Website wird angezeigt.
- **15. Admin anmelden**
Benutzer meldet sich als Admin an. Die Website ist nur für Administratoren verwendbar.
- **16. Admin abmelden**
Benutzer meldet sich als Admin ab.
- **17. Frage hinzufügen**
Eine neue Frage wird hinzugefügt
- **18. Frage bearbeiten**
Eine vorhandene Frage wird bearbeitet
- **19. Frage löschen**
Eine vorhandene Frage wird gelöscht

- **20. User löschen**
Bestimmter User wird gelöscht

2.4 Charakteristika der Benutzer

Beschreibt hier Eure typischen Benutzer. Benutzt dazu die in der Vorlesung vorgestellten Personas. Zur Erinnerung: Ihr beschreibt konkrete Personen, die Repräsentanten der verschiedenen Benutzertypen sind (mit Name, evtl. Wohnort, Tätigkeit, Alter, Bild, ...). Diese sollten eine gewisse Motivation haben, bestimmte Anwendungsfälle durchzuführen (und dort auch eingesetzt werden!).

2.5 Einschränkungen (Tim W. und Tim E.)

Dieser Abschnitt gibt einen groben Überblick über alles, was den Entwurf einschränken könnte. Details folgen dann in Abschnitt 3 („Detaillierte Beschreibung“). Allgemeine Sicherheitsanforderungen könnten also hier kurz beschrieben werden, Details dann im Abschnitt 3.5 zu Systemattributen.

Beispiele für mögliche Einschränkungen könnten sein (die Liste ist nicht vollständig und nicht alle dieser Punkte sind in jedem möglichen Projekt relevant):

- feste Vorgaben (z.B. Policies)
- gesetzliche Rahmenbedingungen (z.B. Datenschutzaspekte)
- Hardwarebeschränkungen
- festgelegte Schnittstellen zu anderen Anwendungen
- parallele Operationen (z.B. Multithreading)
- zu unterstützende Prüffunktionen (auch: Audit-Funktionen; z.B. „alle Buchungen im System müssen protokolliert und von einem Wirtschaftsprüfer nachvollziehbar sein“)
- Steuerungsfunktionen (z.B. „Das System kann von einem Techniker von Ferne gewartet werden“)
- Anforderungen an die zu verwendende Programmiersprache
- Verlässlichkeit; die Verfügbarkeit eines technischen Systems ist die Wahrscheinlichkeit oder das Maß, dass das System bestimmte Anforderungen innerhalb eines vereinbarten Zeitrahmens erfüllt. Eine Beispielanforderung könnte sein, dass das System an jedem Tag der Woche 24 h lang in Betrieb sein muss und im Jahr höchstens 2 Tage nicht verfügbar sein darf.
- Zuverlässigkeit; Software-Zuverlässigkeit wird definiert als die Wahrscheinlichkeit einer fehlerfreien Software-Anwendung über eine spezifizierte Zeitdauer und unter spezifizierten Umgebungsbedingungen. Hier könnte zum Beispiel eine zu errei-

chende Testabdeckung in Bezug auf ein Abdeckungsmaß gefordert werden: „90 % Anweisungsabdeckung bei den Komponententests“.

- *Kritikalität (Gefährlichkeit eines Versagens) der Anwendung*
- *Sicherheit*

2.5.1 Rahmenbedingungen

2.5.2 Gesetzliche Rahmenbedingungen

2.5.3 Sicherheitskritische Aspekte

2.6 Annahmen und Abhängigkeiten (Tim E.)

Faktoren, deren Änderung zwangsläufig zu Änderungen an der Anforderungsspezifikation führen würde.

2.7 Ausblick (Tim E.)

Beschreibt hier knapp, welche Änderungen und Erweiterungen zukünftig (d.h. nach Auslieferung des Systems) zu erwarten sind. Diese Information ist wichtig für den Entwurf, um mögliche Änderungen frühzeitig im ersten Entwurf berücksichtigen zu können. Der Entwurf kann dann so gestaltet werden, dass die zukünftigen Anforderungen leicht realisierbar sind. Die zukünftigen Anforderungen sollten realistisch sein, ansonsten könnte ein unnötig allgemeiner und damit zu komplizierter Entwurf die Folge sein. Auch dieser Abschnitt ist im IEEE-Standard nicht vorgesehen – zumindest nicht explizit in Form eines eigenständigen Abschnitts. Dennoch handelt es sich um wertvolle Information, von der der Entwurf profitieren kann.

3 Detaillierte Beschreibung

Die externen Schnittstellen werden grob in Abschnitt 2 beschrieben. Wenn die grobe Beschreibung dort nicht genügt, kann sie hier detaillierter ausgeführt werden (wie vom IEEE-Standard vorgesehen).

3.1 Datenmodell (Tobias)

Das Datenmodell im Kontext des Pflichtenhefts ist „die Darstellung von Informationen und deren Beziehungen in einem fachlogischen Konzept“. Es soll hier gezeigt werden, welche Einheiten für das existierende System relevant sind und welche Beziehungen zwischen diesen Einheiten gelten. Es handelt sich hierbei noch nicht um ein Datenbankschema oder eine Spezifikation von Klassen für die Implementierung (Entwurf), sondern um die Modellierung der realen Welt. Das Datenmodell ist leitend für den Entwurf (weil alles darin beschrieben sich auch in der Software wiederfinden wird), aber nimmt den Entwurf nicht schon vorweg.

Das Datenmodell soll als UML-Klassendiagramm angegeben werden. Wichtig ist hierbei die korrekte Verwendung der UML: Klassen, Attribute, Generalisierung, Assoziation, Aggregation, Komposition, Multiplizitäten. Außerdem sollte das Diagramm sinnvoll und gut lesbar sein. Dazu gehört weiterhin eine kurze Beschreibung des Modells mit ergänzenden Informationen, insbesondere wenn die Relationen durch ihren Namen nicht selbsterklärend sind. Gebt unbedingt ein Mengengerüst für die Daten an: Wie viele Instanzen der wichtigsten Klassen werden erwartet? Erwartet Ihr Änderungen im Datenvolumen in der Zukunft?

3.2 Anwendungsfälle (Tobias und Daniel)

*Dieser Teil enthält die **funktionalen Anforderungen** an das System. Diese werden durch Anwendungsfälle beschrieben. Insofern müssen die Anwendungsfälle die Funktionalität des Systems vollständig abdecken. Daher müssen auch Varianten von Standardabläufen sowie das Verhalten im Fehlerfall behandelt werden.*

In den Anwendungsfällen beschreibt Ihr, wie Eure Personas mit dem System interagieren, wenn sie ein bestimmtes Ziel erreichen wollen. Dabei sollte der Anwendungsfall zum Profil der Persona passen, also eine typische Anwendung seiner Personengruppe sein. Ihr solltet die Anwendungsfälle textuell beschreiben (im unten aufgeführten Schema) und im Fall von komplexen Anwendungsfällen zusätzlich Sequenzdiagramme verwenden, um durch graphische Darstellung das Verständnis zu erleichtern. Stellt sicher, dass die Mindestanforderungen auf jeden Fall erfasst sind. Weiterhin sollen hier noch keine Implementierungsdetails festgelegt werden, um keine Entwurfsentscheidungen vorwegzunehmen.

Verwendet die Screenshots oder digitalisierten Bilder Eures Papierprototypen, um die

Benutzungsführung in den Anwendungsfällen zu illustrieren und die konkrete Benutzeroberfläche, die es zu implementieren gilt, zu spezifizieren. Die Bilder sollten im Text an der entsprechenden Stelle referenziert werden, um das Verständnis für die Abläufe zu gewährleisten. Die Beschreibung muss so genau sein, dass klar ist, wie welche Aktionen ausgelöst werden und was das für Folgen hat (Beispiel: „Benutzer startet die Suche“ – wie macht er das? „... durch Drücken des Buttons ‚Suche‘“). Die Spezifikation, die die Navigation zwischen Screens und Dialogen beschreibt, nennt man das Navigationsmodell. Es kann zum Beispiel in der Notation eines UML-Zustandsdiagrammes ausgedrückt werden, wobei jeder Screen/Dialog als Zustand aufgefasst wird, Benutzerinteraktionen und sonstige Ereignisse als Transitionen dargestellt werden.

Die Struktur der textuellen Beschreibung sollte sein:

- 1. eindeutiger Name des Anwendungsfalls mit eindeutiger Nummer*
- 2. Akteure: welche externen Instanzen interagieren mit dem System in diesem Anwendungsfall?*
- 3. Vorbedingungen: Ausgangszustand, der vor Beginn des Anwendungsfalls gelten muss – hier sollte auch das Ziel des Akteurs genannt werden*
- 4. Regulärer Ablauf: Abfolge von Aktionen der Akteure und Reaktionen des Systems*
- 5. Varianten: mögliche Abweichungen vom regulären Ablauf, z.B. Auslassen oder Wiederholen von Aktionen*
- 6. Nachbedingung: Endzustand und dann mögliche Folgeaktionen*
- 7. Fehler-/Ausnahmefälle mit deren Nachbedingung; z.B. wie wird auf ungültige Eingaben reagiert?*

3.3 Aktionen

Hier sollten die gleichen Aktionen wie in den Anwendungsfällen genannt und genauer beschrieben werden. Mit anderen Worten: Die Anwendungsfälle müssen vollständig durch Ausführung von Aktionen aus dieser Liste durchführbar sein. Im Prinzip muss es z.B. für jeden Button/Menüpunkt/Link eine Aktion geben. Dabei ist zu beachten:

- Die Namen sollten sinnvoll und eindeutig sein.*
- Die Parameter der Aktionen sollen angegeben werden. Hier sollen sprechende Namen verwendet werden. Eventuell müssen die Parameter auch genauer erläutert werden.*
- Es müssen maximale Ausführungszeiten für jede Operation angegeben werden.*
- Die Gruppierung und Sortierung sollte sinnvoll sein (z.B. alphabetisch).*

*Wenn Ihr z.B. irgendwo in Eurer GUI ein Suchfeld habt, in das Ihr den Namen eines Kunden eintragen könnt, und einen Button, welcher die Suche startet, dann wird es vermutlich eine Aktion **Kunde suchen(name)** geben. Dies ist eine Funktion, die*

*Euer System bereitstellt und die durch Anklicken des Buttons ausgelöst wird. Der Anwendungsfall **Kunde suchen** verwendet dann diese Aktion, enthält aber zusätzlich die Beschreibung der Interaktion mit dem System.*

Dieser Abschnitt ist im Standard im Prinzip vorgesehen, weil hierzu grundsätzlich eine Aussage gemacht werden muss. Die Aktionen sind letztlich die Produktfunktionen, während die Anwendungsfälle die Interaktion zwischen Akteuren und System beschreiben.

3.4 Entwurfseinschränkungen

Wurde bereits in 2.5 behandelt und muss daher hier nicht wiederholt werden. Falls aber eine detailliertere Beschreibung notwendig wäre, wäre hier der geeignete Ort.

3.5 Softwaresystemattribute

Hier werden die sogenannten „nichtfunktionalen Anforderungen“ spezifiziert. Dazu gehören beispielsweise:

- *Performanz*
- *Zuverlässigkeit (Korrektheit, Robustheit, Ausfallsicherheit)*
- *Verfügbarkeit*
- *Sicherheit*
- *Wartbarkeit*
- *Portabilität*

Die spezifizierten Systemattribute müssen hinreichend konkret und überprüfbar formuliert werden.

3.6 Weitere Anforderungen

In diesem Abschnitt können weitere relevante Anforderungen beschrieben werden, die in keine der oben genannten Abschnitte passen.

4 Anhang

Hier können weitere detailliertere Ergebnisse aus der Ist-Analyse oder andere Informationen, die zur Erstellung der Spezifikation gedient haben (z.B. Papierprototypen), angefügt werden.