SOFTWARE REQUIREMENTS SPECIFICATION

Version 1.0.0

BTI7081 - Software Engineering and Design
Frühlingssemester 2020

Berner Fachhochschule
Abteilung Technik und Informatik

Team Blau:
Timon Borter
Sven De Gasparo
Luca Mühlheim
Marc Muster
Elias Schmidhalter

Klienten:
Prof. Urs Künzler
Prof. Dr. Jürgen Vogel

Inhaltsverzeichnis

Vorwort

Zweck des Dokuments

Die «Software Requirements Specification» (SRS) beschreibt das zu entwickelnde System in einer Auslegung von nicht- und funktionalen Anforderungen. Um der besseren Verständlichkeit beizutragen ist ausserdem ein ausgewähltes Set an Use Cases beigefügt.

Versionierung

Die Versionierung wird laufend, nach Änderungen, vom Autor selbst nachgeführt. Es liegt in seiner Verantwortung und seinem Interesse, die Nachvollziehbarkeit im Dokument aufrecht zu erhalten.

Als Vorbild gilt die im Softwareumfeld bekannte semantische Versionierung (SemVer).

Version	Änderungen
0.0.1	Fertiges Vorwort inklusive dieser Versionierung, Glossar, Einleitung
0.0.2	Anforderungen, Architekturdiagramm, Use Case Diagramm
0.0.3	Detaillierte Use Cases, NFA, Testing
0.0.4	Funktionale Anforderungen, System evolution
0.0.5	System models
1.0.0	1 Version für das Kundenreview

Tabelle 1: Versionierung

Zielgruppe

Dieses Dokument richtet sich an die Klienten in der Rolle als Interessengruppen (Stakeholders).

Projektziel

Das Produkt dieses Projektes ist eine Applikation, die es Menschen mit sozialen Angststörungen ermöglicht, sich besser im Alltag zurecht zu finden. Dies soll durch einen spielerischen Ansatz erreicht werden. Details dazu in den Anforderungen.

Einleitung

Für Personen mit sozialen Angststörungen sind jegliche soziale Interaktionen eine Herausforderung. Bereits das Melden bei einem Therapeuten, um einen Termin zu vereinbaren, ist schwierig für sie. Unsere Applikation setzt genau da an, um Menschen auf spielerischer Weise zu helfen. Die Applikation stellt eine Menge von Herausforderungen für Verfügung, aus welchen die Benutzer auswählen können. Eine danach ausgewählte Herausforderung ist für einen gewissen Zeitraum aktiv, in dem sich die Benutzer bemühen, die Herausforderung zu meistern. Später beurteilen die Benutzer selbst, wie gut sie die Herausforderung gemeistert haben. Als Belohnungssystem kann man Punkte sammeln, mit denen man schliesslich Erfolge erzielen kann.

Die zweite Hauptfunktion ist das Verwalten von Journaleinträgen. Hat der Patient eine aussergewöhnliche Situation erlebt, so kann er einen Journaleintrag erfassen. Dies soll dabei helfen, ihn an gute und schlechte Situationen zu erinnern und daraus zu lernen.

Weiterhin bietet die Applikation Möglichkeiten, Therapeuten zu finden und zu kontaktieren. So kann zum Beispiel ein Termin vereinbart werden.

Glossar

Abkürzung / Wort	Erklärung
FURPS	Das Akronym steht für: Functionality (Funktionalität), Usability (Benutzbarkeit), Reliability (Zuverlässigkeit), Performance (Effizienz) und Supportability (Änderbarkeit)
NFA	Nicht-funktionale Anforderungen.
OpenID Connect	OpenID Connect ist eine Authentifizierung-Layer auf Basis von OAuth 2.0, welches die Autorisierung zur Verfügung stellt.
REST	Representational state transfer, ein Software Architektonischer Style der Vorgaben für Web Services definiert. Sogenannte RESTful Web Services stellen eine Schnittstelle zur Interaktion von Computern im Internet zur Verfügung.
SemVer	Semantische Versionierung („Semantic Versioning“), ist eine Art wie im Softwareumfeld Produkte versioniert werden könne. Für mehr Informationen siehe: https://semver.org.
SRS	Software Requirements Specification; dieses Dokument.

Tabelle 2: Glossar

User and System Requirements Specification

Akteure

Im Rahmen der Projektziele (PZ, NZ und ZM, «Design Thinking») gibt es drei Akteure. Es sind folgende:

Akteur	Typ	Beschreibung	Anwendung
Patient (Patient)	Primär	Ein Patient ist eine Person, die unter einer sozialen Angststörung leidet.	Der Patient benutzt das System, um seine Ängste mit Hilfe von verschiedenen Herausforderungen spielerisch zu bekämpfen. Anhand von Belohnungen kann er seinen Fortschritt messen. Er könnte über die Applikation auch mit Therapeuten Kontakt aufnehmen.
System Administrator (System Administrator)	Auswertung	Der System Administrator ist ein Inhaber der Applikation. Er hat Administrativen Zugriff auf sämtliche Systeme.	Als Administrator kann man Auswertungen über die Benutzerbasis sowie deren Aktivitäten erstellen.
Therapeut (Therapist)	Hintergrund	Ein Therapeut ist ein Doktor oder Psychologe, welcher sich im Gebiet der sozialen Angststörungen auskennt und Patienten entsprechend Hilfe anbieten kann.	Als Therapeut kann ich mich beim Inhaber der Applikation melden, um darin mit Kontaktdaten aufgenommen zu werden. Patienten in der Nähe können dann den Kontakt via Applikation auf einem bequemen Weg suchen.

Tabelle 3: Liste der Akteure

Übersicht gemäss den Storyboards

Gemäss den Story Boards wurden die relevanten UseCases genauer beschrieben. Die Software Requirements decken nur die blau Markierten UseCases ab: Diese werden im Rahmen dieses Projekts umgesetzt.

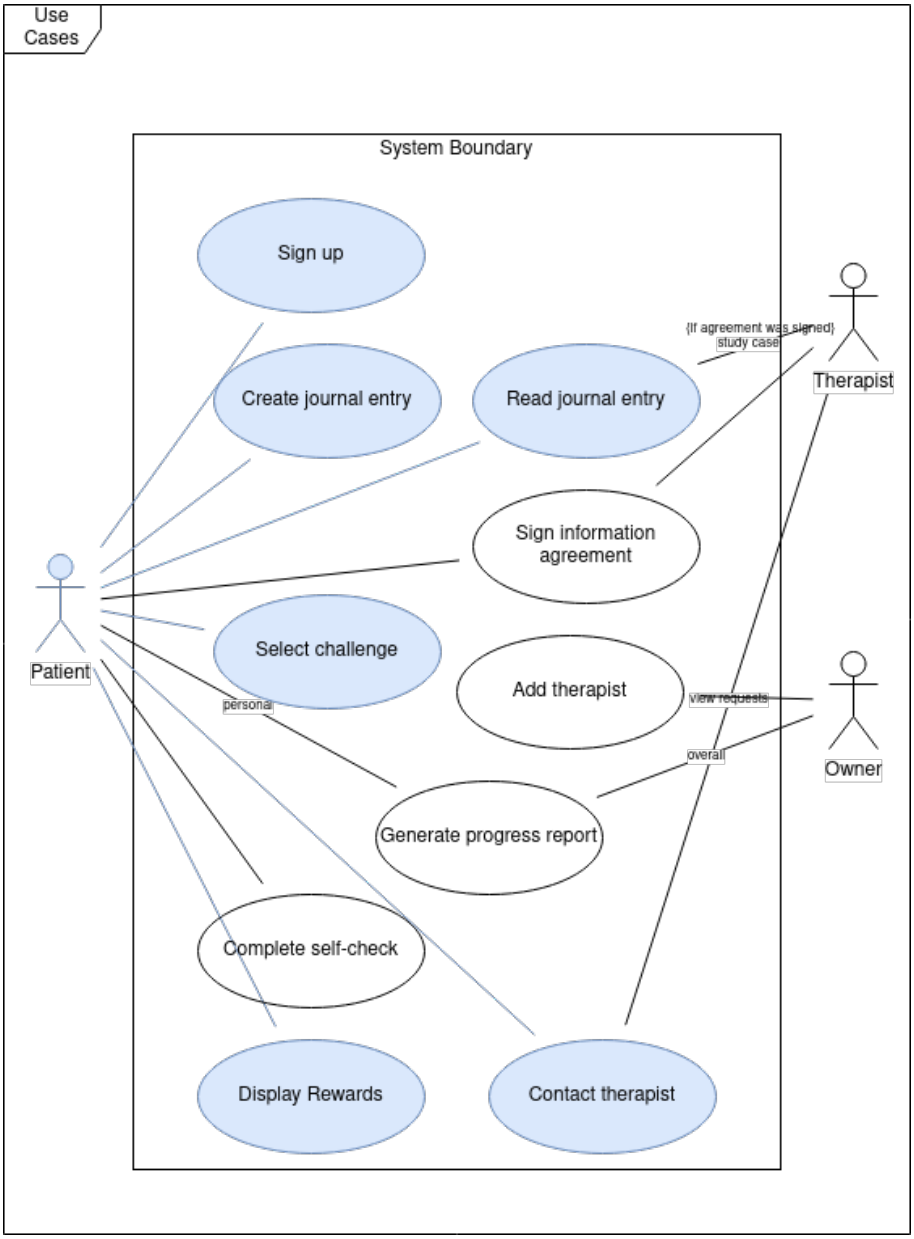

Abbildung 1: Übersicht der Use Cases gemäss den Storyboards

Anwendungsfallbeschreibung

Scenario 1, Journaleintrag erfassen

Nr. und Name:	**Szenario 1 Journal**
Szenario:	Journaleintrag erfassen
Kurzbeschreibung:	Der Patient erfasst eine erlebte Situation ins Journal.
Beteiligt Akteure:	Patient
Auslöser / Vorbedingung:	Hat eine aussergewöhnliche Situation erlebt, die er besonders gut/schlecht bewältigt hat.
Ergebnisse / Nachbedingung:	Die Situation ist in der Webapplikation als Journaleintrag erfasst und kann wieder aufgerufen werden.

Tabelle 4: Scenario 1, Journaleintrag erfassen

Ablauf

Nr.	**Wer**	**Was**
1	Patient	Er erlebt im Alltag eine aussergewöhnliche Situation
2	Patient	Er loggt sich auf der Plattform ein
3	Patient	Er navigiert auf die Journaleinträge
4	Patient	Über einen Button startet er das Erfassen eines neuen Eintrags
5	Patient	Er erfasst sein Erlebtes im neuen Journaleintrag
6	Patient	Er speichert den Journaleintrag

Tabelle 5: Scenario 1, Ablauf

Ausnahmen, Varianten

Nr.	**Wer**	**Was**
2.1	Patient	Der Patient muss sich registrieren, falls er noch kein Konto besitzt.

Tabelle 6: Scenario 1, Ausnahmen, Varianten

Scenario 2, Herausforderung abschliessen

Nr. und Name:	**2, Herausforderung abschliessen**
Szenario:	Herausforderung abschliessen
Kurzbeschreibung:	Der Patient schliesst eine Herausforderung ab.
Beteiligt Akteure:	Patient
Auslöser / Vorbedingung:	Der Patient ist bereits auf der Plattform eingeloggt.
Ergebnisse / Nachbedingung:	Der Patient erhält eine Belohnung.

Tabelle 7: Scenario 2, Herausforderung abschliessen

Ablauf

Nr.	Wer	Was
1	Patient	Er erhält eine Auswahl von Herausforderungen
2	Patient	Er akzeptiert eine Herausforderung aus der Auswahl
3	Patient	Er setzt die Herausforderung in seinem Alltag um
4	Patient	Er navigiert zu der soeben umgesetzten Herausforderung
5	Patient	Über einen Button schliesst er die Herausforderung ab
6	Patient	Er beurteilt seinen Erfolg bei der Tätigkeit
7	Patient	Er erhält abhängig vom Schwierigkeitsgrad der Herausforderung und seiner Beurteilung Punkte

Tabelle 8: Scenario 2, Ablauf

Ausnahmen, Varianten

Nr.	Wer	Was
7.1	Patient	Wenn er seine Leistung bei der Herausforderung schlecht beurteilt, erhält er Tipps für das nächste Mal.

Tabelle 9: Scenario 2, Ausnahmen, Varianten

Funktionale Anforderungen

Nummer	Inhalt
FA1	Die Patienten müssen sich registrieren und einloggen können.
	Die Patienten müssen ihre persönlichen Daten bearbeiten können.
	Die Patienten müssen einen Journaleintrag erfassen, lesen, bearbeiten und löschen können.
	Die Patienten müssen eine Herausforderung aus einer gegebenen Auswahl wählen können.
	Die Patienten müssen ihre Herausforderungen bewerten können.
	Die Patienten müssen bereits erzielte und nicht erzielte Erfolge einsehen können.
	Die Applikation hat ein Erfolgssystem. Patienten können durch Selbstbewertung ihrer Herausforderungen Punkte verdienen. Beim erfolgreichen Abschliessen von Herausforderungen können die Patienten Erfolge verdienen. Ebenfalls gibt das Erreichen von diversen Anzahl Punkten Erfolge.
	Die Applikation gibt immer zu jeder Herausforderung Tipps und Tricks, wie die Situation bewältigt werden kann.

Tabelle 10: Funktionale Anforderungen

Nicht-Funktionale Anforderungen

Die nicht-funktionalen Anforderungen werden nach dem FURPS+ Model kategorisiert, wobei die funktionalen Anforderungen ausgelassen werden (siehe Anwendungsfälle).

Benutzerfreundlichkeit (Usability)

Nummer	Inhalt
U1	Die Applikation lässt sich selbsterklärend benutzen.
U2	Der Anwender soll mit maximal 3 Klicks zum gewünschten Inhalt kommen.

Tabelle 11: NFA; Benutzerfreundlichkeit

Verlässlichkeit (Reliability)

Nummer	Inhalt
R1	Die Applikation zeigt gespeicherte Daten zuverlässig an.
R2	Im Falle eines Datenverlustes können die Informationen wiederhergestellt werden.
R3	Jegliche (personenbezogene) Daten, die nach aussen gelangen (zum Beispiel durch irgendwelche Auswertungen), sind immer anonymisiert.

Tabelle 12: NFA; Verlässlichkeit

Performance (Performance)

Nummer	Inhalt
P1	Das System muss eine grosse Anzahl an Patienten handhaben können. Es Unterstützt mindestens 2'000 Patienten pro Jahr.
P2	Suchanfragen müssen innerhalb der folgenden Spezifikation sein: In 90% der Abfragen ist die Antwortzeit weniger als 2 Sekunden.

Tabelle 13: NFA; Performance

Unterstützung (Supportability)

Nummer	Inhalt
S1	Das System läuft sowohl auf Unix/GNU-Linux als auch auf Microsoft Windows Plattformen.

Tabelle 14: NFA; Unterstützung

System architecture

Die Applikation wird den Benutzern als Website zur Verfügung gestellt. Das Backend stellt die Applikationsdaten bereit. Das Frontend kommuniziert nach dem initialen Ladevorgang mithilfe einer REST Schnittstelle mit dem Backend. Das Backend beinhaltet Validierungs- und Businesslogik. Das Backend speichert die persistenten Daten in einer Datenbank ab. Falls nötig kann das Backend mit weiteren Services, wie zum Beispiel einem Mailservice kommunizieren.

Die auf der nachfolgenden Grafik Blau hinter malten Komponenten gehören zum Umfang dieses Projekts. Gegebenenfalls wird die Authentisierung ebenfalls in der Applikation abgehandelt andernfalls kann ein externer Authentisierungsprovider (mithilfe von OpenID Connect) verwendet werden.

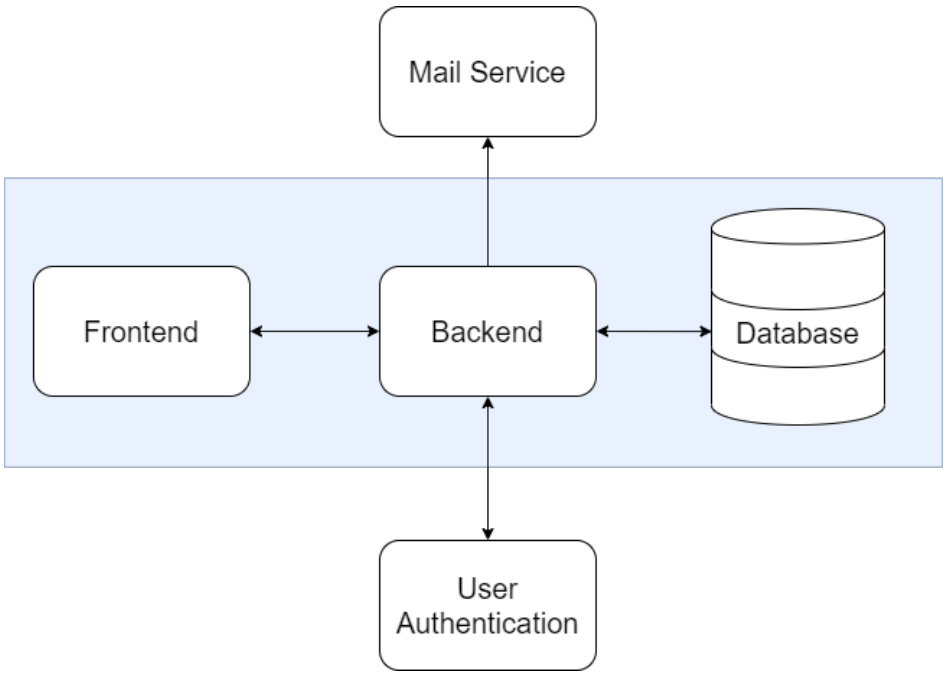

Abbildung 2: System Architektur

System models

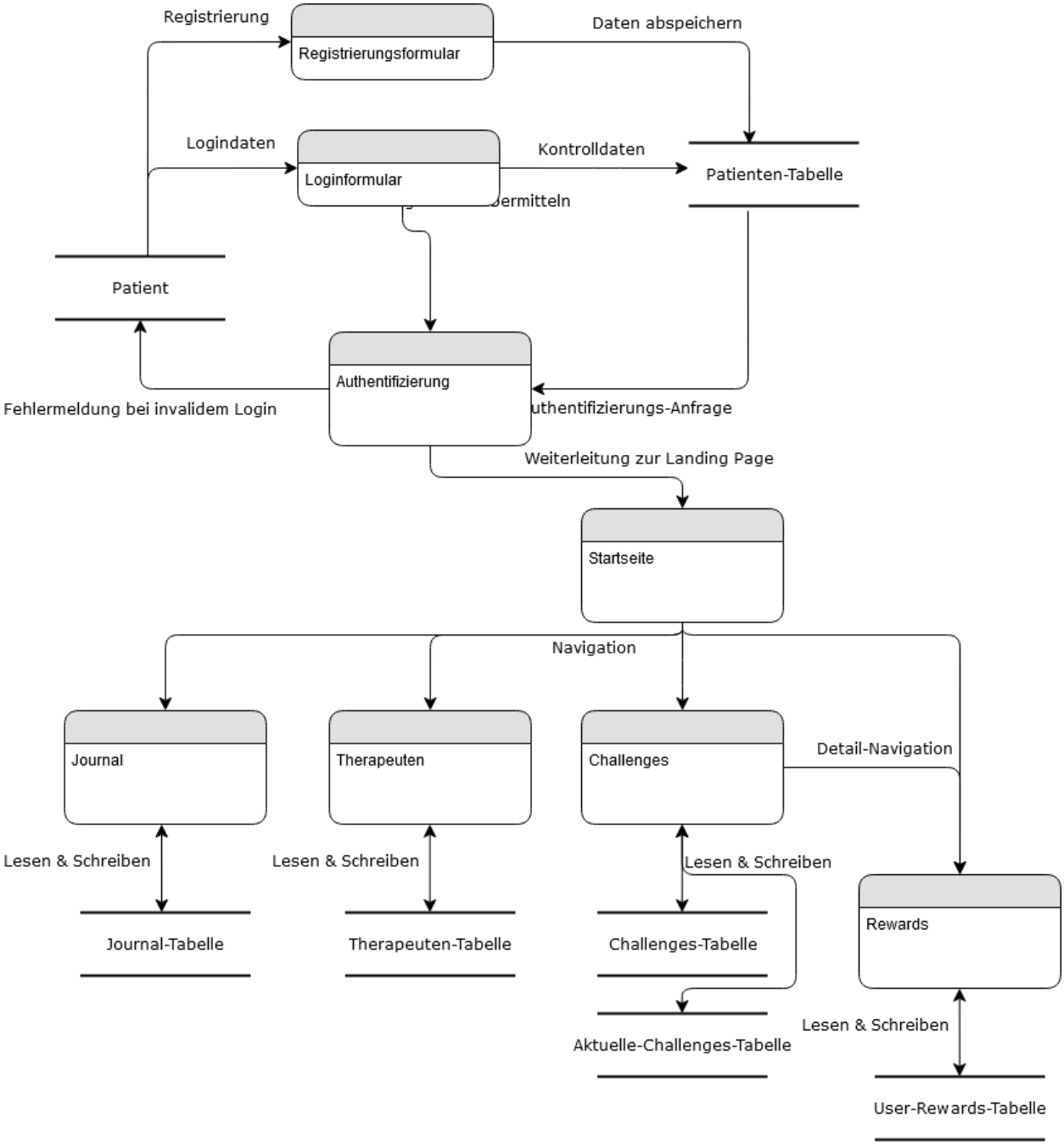

Abbildung 3: Data Flow Model

System evolution

Im Rahmen dieses Projekts wird eine Software erstellt. Diese ist nicht an ein System (Hardware oder Betriebssystem Version) gekoppelt. Dadurch können das unterliegende System getrennt von der Applikation weiterentwickelt und aktualisiert werden.
Vorerst ist keine horizontale Skalierung auf mehrere Server vorgesehen. Falls dies dennoch nötig wäre, müssten insbesondere die Datenhaltung (Datenbank, Sessions) angepasst werden. Das Backend kommuniziert mit der Datenbank über ein definiertes Protokoll, welches eine Aufteilung auf mehrere Server unterstützt.

Testing

Im Front- und Backend wird die Funktionalität jedes komplexeren Feature mit Unit- und wo notwendig mit Integrationstests abgedeckt. Diese Tests werden auf einem Build System kontinuierlich ausgeführt, um die Qualität zu gewährleisten.

Automatisierte System- und Akzeptanztests würden den Rahmen der Proof of Concept Applikation sprengen. Ebenfalls werden keine automatisierten Lasttests umgesetzt.

Unit Testing

Eine Ebene des Softwaretestprozesses, auf der einzelne Einheiten einer Software getestet werden. Der Zweck besteht darin, zu überprüfen, ob jede Einheit der Software die vorgesehene Leistung erbringt.

Integration Testing

Eine Ebene des Softwaretestprozesses, bei der einzelne Einheiten kombiniert und als Gruppe getestet werden. Der Zweck dieser Teststufe besteht darin, Fehler in der Interaktion zwischen integrierten Einheiten aufzudecken.

System Testing

Eine Ebene des Softwaretestprozesses, auf der ein vollständiges, integriertes System getestet wird. Der Zweck dieses Tests besteht darin, die Übereinstimmung des Systems mit den angegebenen Anforderungen zu bewerten.

Akzeptanz Testing

Eine Ebene des Softwaretestprozesses, auf der ein System auf Akzeptanz getestet wird. Der Zweck dieses Tests besteht darin, die Übereinstimmung des Systems mit den Geschäftsanforderungen zu bewerten und zu bewerten, ob es für die Lieferung akzeptabel ist.

Index

Abbildungsverzeichnis

Tabellenverzeichnis