HINWEIS: Blauer Text kann gelöscht werden, beziehungsweise soll ersetzt werden

- Anforderungen-

…hier Name des Systems eintragen…

Version: …eintragen…

|  |  |  |
| --- | --- | --- |
| Projektbezeichnung | PRODUKT.PROJEKTBEZEICHNUNG | |
| Projektleiter |  | |
| Verantwortlich | Anforderungsanalytiker (AG) | |
| Erstellt am |  | |
| Zuletzt geändert | 18.03.2014 13:59 | |
| Bearbeitungszustand | X | in Bearbeitung |
|  | vorgelegt |
|  | fertig gestellt |
| Dokumentablage | /media/r0ot/data/studium/softwareprojekt/OTH\_SWP\_SS15/Basisverzeichnis/trunk/01\_Analyse/Anforderungen.odt | |

Weitere Produktinformationen

|  |  |
| --- | --- |
| Mitwirkend | [nicht beteiligt] Projektleiter  [nicht beteiligt] Projektmanager  [nicht beteiligt] Anwender |

Änderungsverzeichnis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Änderung | | | Geänderte Kapitel | Beschreibung der Änderung | Autor | Zustand |
| Nr. | Datum | Version |  |  |  |  |
| 1 |  | 1.1 | Alle | Initiale Produkterstellung |  |  |

Prüfverzeichnis

Die folgende Tabelle zeigt einen Überblick über alle Prüfungen – sowohl Eigenprüfungen wie auch Prüfungen durch eigenständige Qualitätssicherung – des vorliegenden Dokumentes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Datum | Geprüfte Version | Anmerkungen | Prüfer | Neuer Produktzustand |
|  |  |  |  |  |

Inhalt

[1 Einleitung 4](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574146)

[2 Ausgangssituation und Zielsetzung 4](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574147)

[3 Funktionale Anforderungen 5](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574148)

[3.1 Use-Case Übersicht 5](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574149)

[3.2 Use-Beschreibungen 5](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574150)

[3.3 (Sonstige) Funktionalität 6](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574151)

[3.4 Modell des Problembereichs (Konzeptionelles Datenmodell) 7](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574152)

[4 Nicht-Funktionale Anforderungen 7](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574153)

[4.1 Benutzbarkeit (Usability) 7](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574154)

[4.2 Zuverlässigkeit (Reliability) 8](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574155)

[4.3 Leistung (Performance) 8](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574156)

[4.4 Unterstützbarkeit (Supportability) 9](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574157)

[4.5 Sonstige Einschränkungen 9](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574158)

[5 Risikoakzeptanz 10](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574159)

[6 Skizze der Gesamtsystemarchitektur 10](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574160)

[7 Lieferumfang 11](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574161)

[8 Abnahmekriterien 11](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574162)

[9 Abkürzungsverzeichnis 11](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574163)

[10 Literaturverzeichnis 12](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574164)

[11 Abbildungsverzeichnis 12](file:///C:\media\r0ot\data\studium\softwareprojekt\Dateivorlagen\Systemanalyse\HYPERLINK#_Toc382574165)

# 

Einleitung

Dieses Dokument enthält alle an das zu entwickelnde System gestellten Anforderungen. Die Gliederung orientiert sich am Aufbau des V-Modell-XT®[[1]](#footnote-2)-Produkts „Anforderungen (Lastenheft)“, ist jedoch für die Verwendung in der Veranstaltung **„Software-Projekte“** im Studiengang **„Angewandte Informatik“** der **OTH-Amberg-Weiden** angepasst worden (und nicht konform zum V-Modell-XT): Teilnehmer dieser Veranstaltung erhalten von ihrem „Auftraggeber“ lediglich einen Überblick über das gewünschte System (siehe das Thema „Ausgangssituation und Zielsetzung“ in diesem Dokument); die Anforderungen müssen die Teilnehmer dann in enger Abstimmung mit ihrem „Auftraggeber“ selbst erarbeiten und in diesem Dokument niederlegen. Dadurch sollen sie Gelegenheit erhalten, sich in der Herausarbeitung von Anforderungen intensiver zu üben.

Weil der „Auftraggeber“ also nicht wirklich ein Lastenheft liefert, aus dem die „Auftragnehmerseite“ (=Teilnehmer am Software-Projekt) dann ein separates Pflichtenheft ableitet, dient dieses Dokument als Ersatz für Lasten- und Pflichtenheft im Sinne des V-Modell-XT.

Kern dieses Dokuments sind die funktionalen und nicht-funktionalen Anforderungen an das System, sowie eine Skizze des Gesamtsystementwurfs. Der Entwurf berücksichtigt die zukünftige Umgebung und Infrastruktur, in der das System später betrieben wird, und gibt Richtlinien für Technologieentscheidungen. Ebenfalls Teil der Anforderungen ist die Festlegung von Lieferbedingungen und Abnahmekriterien.

Die funktionalen und nicht-funktionalen Anforderungen dienen nicht nur als Vorgaben für die Entwicklung, sondern sind zusätzlich Grundlage der Anforderungsverfolgung und des Änderungsmanagements. Die Anforderungen sollten so aufbereitet sein, dass die Verfolgbarkeit (Traceability) sowie ein geeignetes Änderungsmanagement für den gesamten Lebenszyklus eines Systems möglich sind.

Im Allgemeinen sollten keine technischen Lösungen vorgegeben werden, um Architekten und Entwickler bei der Suche nach optimalen technischen Lösungen nicht einzuschränken.

# Ausgangssituation und Zielsetzung

In diesem Thema werden die Ausgangssituation und der Anlass zur Durchführung des Projekts anschaulich dargestellt. Es wird beschrieben, welche Defizite bzw. Probleme existierender Systeme oder auch der aktuellen Situation zur Entscheidung geführt haben, das Projekt durchzuführen, und welche Vorteile durch den Einsatz des neuen Systems erwartet werden.

Es werden zusätzlich alle relevanten Stakeholder des Projekts benannt und die technische und fachliche Einbettung des zu entwickelnden Systems in seine Umgebung skizziert. Zusätzlich werden erste Rahmenbedingungen für die Entwicklung identifiziert und beschrieben. Rahmenbedingungen können beispielsweise technische Vorgaben oder Vorgaben zur Sicherheit sein.

…hier Ihren Text einfügen…

Funktionale Anforderungen

Funktionale Anforderungen beschreiben die Fähigkeiten eines Systems, die ein Anwender erwartet, um mit Hilfe des Systems ein fachliches Problem zu lösen. Die Anforderungen werden aus den zu unterstützenden Geschäftsprozessen und den Ablaufbeschreibungen zur Nutzung des Systems abgeleitet.

Die Beschreibung der funktionalen Anforderungen erfolgt beispielsweise in Form von Anwendungsfällen (Use Cases). Ein Anwendungsfall beschreibt dabei einen konkreten, fachlich in sich geschlossenen Teilvorgang. Die Gesamtheit der Anwendungsfälle definiert das Systemverhalten. Ein Anwendungsfall kann in einfachem Textformat beschrieben werden, häufig stehen jedoch organisationsspezifische Muster zur Beschreibung zur Verfügung. Für datenzentrierte Systeme wird im Rahmen der funktionalen Anforderungen ein erstes fachliches Datenmodell erstellt, das als Grundlage des späteren Datenbankentwurfs dient. Das fachliche Datenmodell des Systems wird aus den Entitäten des Domänenmodells abgeleitet.

Die funktionalen Anforderungen sind die zentralen Vorgaben für die Systementwicklung.

## Use-Case Übersicht

…hier Use-Case-Diagramm(e) einfügen…

## Use-Case-Beschreibungen Framework

|  |  |
| --- | --- |
| **Graph erstellen** | |
| **Kennung** | G-1 |
| **Priorität** | hoch |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion erstellt ein neues Objekt vom Typ Graph | |
| **Parameter:** | |
| Keine Parameter | |
| **Nachbedingung(en)/Rückgabe:** | |
| Ein gültiges Graph Objekt wurde erstellt. | |

|  |  |
| --- | --- |
| **Graph löschen** | |
| **Kennung** | G-2 |
| **Priorität** | niedrig |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion entfernt alle Graphelemente vom Graph Objekt und versetzt dieses in einen gültigen aber leeren Ausgangszustand. | |
| **Parameter:** | |
| Keine Parameter | |
| **Nachbedingung(en)/Rückgabe:** | |
| Das Graph Objekt besitzt keine Graphelemente mehr und jeglicher Speicher ist zur Freigabe markiert. | |

|  |  |
| --- | --- |
| **Knoten hinzufügen** | |
| **Kennung** | K-1 |
| **Priorität** | hoch |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion fügt dem Graphen ein Knoten Objekt hinzu. | |
| **Parameter:** | |
| Der hinzuzufügende Knoten. | |
| **Nachbedingung(en)/Rückgabe:** | |
| Der Graph besitzt das übergebene Knoten Objekt. | |

|  |  |
| --- | --- |
| **Knoten löschen** | |
| **Kennung** | K-2 |
| **Priorität** | niedrig |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion entfernt einen im Graph befindliches Knoten Objekt. | |
| **Parameter:** | |
| Ein Index, der den zu entfernenden Knoten eindeutig identifiziert. | |
| **Nachbedingung(en)/Rückgabe:** | |
| Der Graph besitzt nicht länger den identifizierten Knoten. | |

|  |  |
| --- | --- |
| **Knoten suchen** | |
| **Kennung** | K-3 |
| **Priorität** | mittel |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion sucht einen Knoten im Graph und liefert im Erfolgsfall dieses zurück. | |
| **Parameter:** | |
| Ein Index oder ein Attribut, der den zu findenden Knoten eindeutig identifiziert. | |
| **Nachbedingung(en)/Rückgabe:** | |
| Eine Liste mit Knoten, die die Suchkriterien erfüllen. Falls kein Knoten gefunden werden konnte, liefert die Funktion einen Null-Wert zurück. | |

|  |  |
| --- | --- |
| **Knoten erstellen** | |
| **Kennung** | K-4 |
| **Priorität** | hoch |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion erstellt ein neues Objekt vom Typ Knoten. | |
| **Parameter:** | |
| Keine Parameter | |
| **Nachbedingung(en)/Rückgabe:** | |
| Ein gültiges Knoten Objekt wurde erstellt. | |

|  |  |
| --- | --- |
| **Attribut erstellen** | |
| **Kennung** | A-1 |
| **Priorität** | hoch |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion erstellt ein neues Objekt vom Typ Attribut. | |
| **Parameter:** | |
| Der Typ, den das Attribut repräsentieren soll. | |
| **Nachbedingung(en)/Rückgabe:** | |
| Ein gültiges Attribut Objekt wurde erstellt. | |

|  |  |
| --- | --- |
| **Attribut hinzufügen** | |
| **Kennung** | A-2 |
| **Priorität** | mittel |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion fügt einem Graphenelement ein Attribut hinzu. | |
| **Parameter:** | |
| Das zu hinzuzufügende Attribut. | |
| **Nachbedingung(en)/Rückgabe:** | |
| Das Graphenelement besitzt das Attribut. | |

|  |  |
| --- | --- |
| **Attribut löschen** | |
| **Kennung** | A-3 |
| **Priorität** | mittel |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion ändert einen Wert des Attributs | |
| **Parameter:** | |
| Keine Parameter | |
| **Nachbedingung(en)/Rückgabe:** | |
| Das Graphenelement besitzt nicht länger das Attribut | |

|  |  |
| --- | --- |
| **Attribut ändern** | |
| **Kennung** | A-4 |
| **Priorität** | mittel |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion löscht ein Attribut eines Graphenelements. | |
| **Parameter:** | |
| Keine Parameter | |
| **Nachbedingung(en)/Rückgabe:** | |
| Der Wert des Attributs ist geändert. | |

|  |  |
| --- | --- |
| **Attribut abfragen** | |
| **Kennung** | A-5 |
| **Priorität** | mittel |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion sucht ein Attribut eines Graphenelements und gibt dieses im Erfolgsfall zurück. | |
| **Parameter:** | |
| Der Typ, den das Attribut besitzen muss. | |
| **Nachbedingung(en)/Rückgabe:** | |
| Eine Liste mit gefundenen Attributen. Falls kein Attribut gefunden werden konnte, liefert die Funktion einen Null-Wert zurück. | |

|  |  |
| --- | --- |
| **Kante erstellen** | |
| **Kennung** | KA-1 |
| **Priorität** | hoch |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion erstellt ein neues Objekt vom Typ Kante. | |
| **Parameter:** | |
| Keine Parameter | |
| **Nachbedingung(en)/Rückgabe:** | |
| Ein gültiges Kante Objekt wurde erstellt. | |

|  |  |
| --- | --- |
| **Kante löschen** | |
| **Kennung** | KA-2 |
| **Priorität** | mittel |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion entfernt die Kante. | |
| **Parameter:** | |
| Keine Parameter | |
| **Nachbedingung(en)/Rückgabe:** | |
| Die Kante ist gelöscht. | |

|  |  |
| --- | --- |
| **Kante ändern** | |
| **Kennung** | KA-3 |
| **Priorität** | hoch |
| **Kurzbeschreibung:** | |
| Der Aufruf dieser Funktion ändert die mit der Kante verbundenen Knoten | |
| **Parameter:** | |
| Keine Parameter | |
| **Nachbedingung(en)/Rückgabe:** | |
| Die Knoten der Kante sind geändert. Der Zustand der Kante ist weiterhin gültig. | |

## Use Case-Beschreibungen Tool

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case-Dokument Importieren** | | | | |
| **Kennung** | | | | UC-1 |
| **Priorität** | | | | hoch |
| **Kurzbeschreibung:** | | | | |
| Der Anwender möchte ein vorliegendes Word Dokument, welches UseCases beinhaltet, in das Tool importieren. | | | | |
| **Vorbedingung(en):** | | | | |
| Das Dokument (.docx) hat das richtige Format und ist nicht beschaedigt. | | | | |
| **Nachbedingung(en):** | | | | |
| Die UseCases existieren als Datenstruktur und können weiterverarbeitet werden. Es liegt eine Liste aller erfolgreich eingelesenen UseCases vor. | | | | |
| **Normaler Ablauf:** | | | | |
|  | | | 1. Der Anwender betätigt den Button zum Dokument einlesen. 2. Der Anwender wählt den Pfad zu seinem UseCaseDokument. 3. Der Anwender wählt sein Dokument aus und bestätigt den Dialog. 4. Das Tool gibt eine Rückmeldung, ob Importvorgang erfolgreich. 5. Das Tool zeigt eine Liste von eingelesenen UseCases. 6. Der UseCase endet. | |
| **Ablauf-Varianten:** | | | | |
| 2a | | Der Anwender bricht den Dialog zur Dateiauswahl ab | | |
|  | | 1. Der UseCase endet | | |
| 3a | | Der Anwender bricht den Dialog zur Dateiauswahl ab | | |
|  | | 1. Der UseCase endet | | |
| 4a | | Ein Ringschluss zwischen UseCases wurde vom Tool erkannt | |
|  | | 1. Der Anwender erhält negatives Feedback – Das Tool informiert über die Ursache des Fehlers (Ringschluss wurde erkannt) 2. Das Tool überspringt den UseCase und fährt mit dem nächsten UseCase fort. | |
| 4b | | UseCase-Bezeichnung kommt mehrmals vor | |
|  | | 1. Der Anwender erhält negatives Feedback – Das Tool informiert über die Ursache des Fehlers (kann WORD-Dokument nicht öffnen, UseCases haben falsches Format) 2. Das Tool überspringt den UseCase und fährt mit dem nächsten UseCase fort. | |
| 4c | | Fehler in der Dokumentstruktur der Word-Datei (.docx) | |
|  | | 1. Der Anwender erhält negatives Feedback – Das Tool informiert über die Ursache des Fehlers (kann WORD-Dokument nicht öffnen, Fehler in der Dokumentenstruktur oder einfach nur leeres Dokument) 2. Der UseCase endet. | |
| **Spezielle Anforderungen:** | | | | |
| keine | | | | |
| **Zu klärende Punkte:** | | | | |
| Soll der Anwender mehrere Dateien auswählen können, die eingelesen werden sollen? | | | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Graph visualisieren** | | |
| **Kennung** | | UC-SM-1 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Das Programm visualisiert mithilfe des Systems und vorher importierten UseCases den Graphen. | | |
| **Vorbedingung(en):** | | |
| Mindestens ein Use Case muss importiert sein | | |
| **Nachbedingung(en):** | | |
| Ein Use Case Graph wurde visualisiert | | |
| **Normaler Ablauf:** | | |
|  | 1. Dieser Anwendungsfall beginnt, sobald der User ein UseCase aus der Liste selektiert. 2. Die Zeichenfläche des Use Cases wird angezeigt 3. Das Tool analysiert den Graphen des entsprechenden Use Cases und zeichnet alle Knoten und Kanten (mithilfe des Graphvisualisierer?)auf die Zeichenfläche. 4. Sobald der Graph fertig gezeichnet wurde, endet der Use Case | |
| **Ablauf-Varianten:** | | |
|  |  | |
|  |  | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Sequenzgraph visualisieren** | | |
| **Kennung** | | UC-SM-1 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Das Programm visualisiert mithilfe des Systems und vorher importierten Use-Cases den Sequenzgraphen. | | |
| **Vorbedingung(en):** | | |
| Mindestens ein Use Case muss importiert sein und mindestens eine Sequenz wurde vom System aus diesem analysiert. Ein Use Case muss aus einer Liste selektiert sein. Für jede Sequenz muss eine Farbe definiert sein | | |
| **Nachbedingung(en):** | | |
| Ein Sequenzgraph wurde visualisiert | | |
| **Normaler Ablauf:** | | |
|  | 1. Dieser Anwendungsfall beginnt, sobald der User eine Sequenz aus einer Liste selektiert. 2. Die ausgewählte Sequenz wird dem Tool (an den Graphvisualisierer) übergeben 3. Alle Farben des Use Case Graphen werden zurückgesetzt 4. Alle Knoten und Kanten der Sequenz werden in einer zuvor festgelegten Farbe auf der Zeichenfläche eingefärbt (Sequenz wird dem Graphvisualisierer übergeben). 5. Sobald der Sequenzgraph fertig gezeichnet wurde, endet der Use Case | |
| **Ablauf-Varianten:** | | |
|  |  | |
|  |  | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Elementdetails anzeigen** | | |
| **Kennung** | | UC-SM-1 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Das Programm visualisiert mithilfe des Systems alle Details des zuvor ausgewählten Elements. | | |
| **Vorbedingung(en):** | | |
| Mindestens ein Element muss selektiert sein | | |
| **Nachbedingung(en):** | | |
| Die Details des selektierten Elements werden angezeigt | | |
| **Normaler Ablauf:** | | |
|  | 1. Dieser Anwendungsfall beginnt, sobald der User ein Element (Use Case oder Sequenz aus der Liste oder ein Element auf der Zeichenfläche) auswählt. 2. Das ausgewählte Element wird dem Tool (Dialog View Model) übergeben 3. Alle Eigenschaften des Elements werden dem User in einem separaten Bereich des Tool angezeigt und visualisiert 4. Sobald alle Eigenschaften angezeigt werden, endet der Use Case | |
| **Ablauf-Varianten:** | | |
|  |  | |
|  |  | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Szenario-Matrix exportieren** | | |
| **Kennung** | | UC-SM-1 |
| **Priorität** | | hoch |
| **Kurzbeschreibung:** | | |
| Der User erstellt mithilfe des Systems und einer vorher importierten Use-Cases die daraus resultierende Szenario-Matrix. Diese wird als Graph dargestellt und kann anschließend als XML-Datei wieder exportiert werden (siehe US-SM-2) | | |
| **Vorbedingung(en):** | | |
| Ein Use Case muss bereits importiert sein, es gibt ein Tabellenlayout | | |
| **Nachbedingung(en):** | | |
| Ein Graph der Szenario-Matrix wurde erstellt | | |
| **Normaler Ablauf:** | | |
|  | 1. Dieser Anwendungsfall beginnt, wenn der User über die Benutzer-System-Schnittstelle die Option „Szenario-Matrix exportieren“ auswählt 2. Der Benutzer wählt über einen entsprechenden Dialog einen Pfad aus, in dem die exportierte Tabellen-Datei abgelegt werden soll 3. Der Benutzer wird gebeten eine Zahl n anzugeben. Die Szenarien werden anschließen auf jene beschränkt, die nicht mehr als n Varianten kombinieren 4. Das Tool erfasst alle Szenarien („Pfade durch den Use-Case“) durch eine geeignete Graph-Traversierung. Pfade, die mehr als n Varianten kombinieren, werden nicht berücksichtigt 5. Pro Szenario wird der Start gespeichert und alle möglichen Ablaufvarianten 6. Anhand eines, im System definierten Tabellenlayout, wird eine „Szenario-Matrix“-Tabelle erstellt. Die Spalten sind in diesem Falle die ID’s, Szenario-Beschreibung, der Startpunkt, die Ablaufvarianten, die Bemerkung, etc. (siehe Tabellenlayout). Eine Reihe ist ein Szenario 7. Das System erzeugt nun anhand der im Speicher liegenden Matrix und dem Tabellenlayout eine XML-konforme Tabellendatei. 8. Der Ordner, in welchem die exportierte Datei liegt, wird im Explorer geöffnet. Der Use Case endet hier | |
| **Ablauf-Varianten:** | | |
| 3a | Die Benutzereingabe war falsch | |
|  | 1. Der Eingabedialog wird erneut geöffnet 2. Weiter bei Punkt 2 | |
| 4a | Es werden keine Pfade erkannt | |
|  | 1. Das System signalisiert, dass keine Pfade und somit keine Szenarien gefunden wurden und der Use-Case evtl. unvollständig ist  2. Es wird keine Szenario-Matrix erstellt. Der Anwendungsfall endet hier | |
| 4b | Es werden Ringschlüsse gefunden | |
|  | 1. Das System beschränkt sich auf alle Szenarien die den Teilabschnitt überhaupt nicht, genau einmal, genau zweimal durchlaufen | |
| 4c | Es wurde keine Szenario-Beschreibung gefunden | |
|  | 1. Das System signalisiert, dass der Use Case wahrscheinlich unvollständig ist 2. Die Szenario-Matrix wird nicht exportiert 3. Der Anwendungsfall endet. | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

## (Sonstige) Funktionalität

Hier werden funktionale Anforderungen erfasst, die sich nicht vernünftig durch einen einzelnen Use-Case beschreiben lassen, weil sie z.B. „Querschnitt“-Funktionalität betreffen, die sich über viele Use-Cases erstreckt. Beispiele könnten sein:

|  |  |  |
| --- | --- | --- |
| **ID** | **Beschreibung** | **Querverweise** |
| FR-001 | Das System soll jede Fehlersituation dauerhaft protokollieren |  |
| FR-002 | Nutzung des Systems ist generell nur nach vorheriger Authentifizierung möglich |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Modell des Problembereichs (Konzeptionelles Datenmodell)

Hier wird ein Modell des Problembereichs („konzeptuelles Datenmodell“) in Form eines oder mehrerer UML-Klassendiagramme eingefügt. Das konzeptuelle Datenmodell wird oft nicht streng getrennt von den Anforderungen erarbeitet, weil ein genaues Verständnis z.B. der Use-Cases nicht ohne Verständnis der „Konzepte“ aus dem Problembereich möglich ist. Aus diesem Grund wird in den Software-Projekten für den Studiengang „Angewandte Informatik“ das konzeptionelle Modell an dieser Stelle zusammen mit den Use-Cases dokumentiert.

# Nicht-Funktionale Anforderungen

Nicht-funktionale Anforderungen beschreiben Anforderungen an das System, die nicht-fachlicher Natur sind, jedoch entscheidend zur Anwendbarkeit des Systems beitragen. Sie definieren beispielsweise Qualitätsanforderungen, Sicherheitsanforderungen oder Performanceanforderungen.

Nicht-funktionale Anforderungen definieren grundlegende Eigenschaften eines Systems, die im Architekturentwurf berücksichtigt werden müssen. Sie können zur Abschätzung der Entwicklungskosten herangezogen werden und sollten, soweit möglich, messbar beschrieben sein.

Zur einfachen Strukturierung der Anforderungen werden diejenigen Anforderungen, die nicht eindeutig zu den funktionalen Anforderungen gehören, den nicht-funktionalen Anforderungen zugeordnet.

Die hier verwendete Einteilung unterscheidet verschiedene Arten von Anforderungen nach dem „FURPS“-Schema (**F**unctionality, **U**sability, **R**eliability, **P**reformance, **S**upportability), das auf Hewlett-Packard zurückgeht. Das FURPS-Schema ist hier noch um die Kategorie „Sonstige Einschränkungen“ erweitert. Anforderungen zur Funktionalität sind bereits im Kapitel 3 dokumentiert, in diesem Kapitel folgen lediglich alle restlichen Anforderungen.

Bei Bedarf kann dieses Schema zur Einteilung der Anforderungen auch durch ein anderes Schema (z.B. nach DIN ISO) ersetzt werden – wichtig ist nur, dass bei der Erfassung der Anforderungen überhaupt ein erprobtes Schema verwendet wird. Dies soll eine bessere Übersicht bieten und dazu beitragen, dass keine Anforderungen vergessen werden.

## Benutzbarkeit (Usability)

Hier werden Anforderungen erfasst, die die Benutzbarkeit („usability“ = Benutzbarkeit / Benutzerfreundlichkeit / Gebrauchstauglichkeit) des Systems betreffen. Hierzu zählen insbesondere Anforderungen zur (Software/Hardware)-Ergonomie („human factors“).

|  |  |  |
| --- | --- | --- |
| **ID** | **Beschreibung** | **Querverweise** |
| UR-001 | Das System soll… |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Zuverlässigkeit (Reliability)

Hier werden Anforderungen erfasst, die die Zuverlässigkeit („reliability“ = Zuverlässigkeit) des Systems betreffen. Hierunter fallen insbesondere Anforderungen an die Wiederherstellbarkeit („recoverability“) und die Verfügbarkeit („availabilty“) des Systems. Die Wiederherstellbarkeit betrifft die Fähigkeit, bei Ausfall oder Störung das Leistungsniveau wieder zu erreichen und betroffene Daten wieder zu gewinnen. Eine Kenngröße in diesem Zusammenhang ist MTTR („**m**ean **t**ime **t**o **r**epair“), die die mittlere Zeit bis zur Wiederinbetriebnahme des Systems bei einem Ausfall angibt. Die Verfügbarkeit kann als Verhältnis zwischen der Zeit, in der das System funktionsfähig ist, und der Gesamtzeit angegeben werden:

V=MTBF / (MTBF+MTTR)

MTTR: mean time to repair (s.o.)

MTBF: **m**ean **t**ime **b**etween **f**ailures (also die mittlere Zeit zwischen zwei Ausfällen)

|  |  |  |
| --- | --- | --- |
| **ID** | **Beschreibung** | **Querverweise** |
| RR-001 | Das System soll… |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Leistung (Performance)

Hierunter fallen Anforderungen an die Leistung („performance“) des Systems. Die Anforderungen beziehen sich insbesondere auf das Zeitverhalten (Ausführungsgeschwindigkeit, Antwortzeiten, Durchsatz) sowie auf das Verbrauchsverhalten (Anzahl der belegten Betriebsmittel und Dauer der Betriebsmittelbelegung).

|  |  |  |
| --- | --- | --- |
| **ID** | **Beschreibung** | **Querverweise** |
| PR-001 | Das System soll… |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Unterstützbarkeit (Supportability)

Hierunter fallen Anforderungen, die Bereiche wie Anpassbarkeit („adaptability“), Testbarkeit („testability“), Wartbarkeit („maintainability“), Erweiterbarkeit („extensibility“), Lokalisierbarkeit („localizability“ = Anpassbarkeit an verschiedene Sprach- und Kulturräume) betreffen.

|  |  |  |
| --- | --- | --- |
| **ID** | **Beschreibung** | **Querverweise** |
| SR-001 | Das System soll… |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Sonstige Einschränkungen

In manchen Fällen können von vorneherein Einschränkungen („constraints“) für Entwurf, Implementierung, Schnittstellen und Hardware des geplanten Systems bestehen, die ebenfalls als Anforderungen zu berücksichtigen sind und das bisherige „FURPS“-Schema zu „FURPS+“ (vgl. [Lar], S. 88) erweitern.

### Schnittstellen

### Implementierung

### Entwurf

Risikoakzeptanz

Für sicherheitskritische Systeme werden in diesem Thema Vorgaben für die Behandlung der Systemsicherheit festgelegt. Es wird aufgezeigt, welche Risiken im Rahmen des Systembetriebs bestehen, welche Schäden, oder auch welche Klassen von Schäden, mit welcher Wahrscheinlichkeit auftreten können und inwieweit das Eintreten eines Schadensfalls toleriert wird bzw. nicht mehr akzeptabel ist.

Die Risikoakzeptanz für die identifizierten möglichen Schadensfälle wird beispielsweise in Form einer Risikoakzeptanzmatrix dokumentiert. Die Matrix ist eine Vorgabe des Auftraggebers, in der er festlegt, bei welcher Schadensklasse und welcher Eintrittswahrscheinlichkeit er welche Risikoklasse akzeptiert.

Bei vielen Aufgabenstellungen in der Veranstaltung „Software-Projekte“ kann dieses Thema weggelassen werden.

…hier Ihren Text einfügen…

# Skizze der Gesamtsystemarchitektur

Das reine Aufstellen von Anwenderanforderungen ohne Überlegungen zu möglichen Lösungsräumen birgt die große Gefahr, unrealistische Anwenderanforderungen zu definieren. Für die Einordnung, Systematisierung, Kategorisierung und auch Priorisierung von Anwenderanforderungen ist ein Koordinierungsrahmen hilfreich, um die Visualisierung der Anwenderanforderungen zu erleichtern.

Diese Aufgabe kann eine Gesamtsystemarchitektur leisten, die die Sichtweise des Anwenders repräsentiert und nicht die technische Sichtweise des Systemanalytikers beziehungsweise des Systemarchitekten. Das heißt, es ist eine funktionale Systemarchitektur mit Einbettung in die funktionalen Abläufe von Nachbarsystemen zu erstellen. Eine technische Systemarchitektur ist in dieser frühen Phase kaum möglich.

Des Weiteren sind die Besonderheiten der Einsatzumgebung des neuen Systems zu beschreiben, um vor allem die Anforderungen an die Systemsicherheit berücksichtigen zu können.

…hier Ihren Text einfügen…

Lieferumfang

Die folgende Tabelle enthält alle Arbeitsergebnisse, die in der Veranstaltung „Software-Projekte“ zu dem vom Team zu liefernden „End-Produkt“ gehören – für die individuell von jedem Projektteilnehmer zu liefernden Ergebnisse lesen Sie bitte im Projektleitfaden bzw. im Projektkalender nach. Die Benotung erfolgt nicht nur auf Grundlage des lauffähigen Programms, sondern bezieht die Qualität der Analyse, des Entwurfs und des Systemtests mit ein.

|  |  |  |  |
| --- | --- | --- | --- |
| **Lfd. Nr.** | **Was?** | **Art des Dokuments** | **Bemerkungen** |
| **Ergebnis der System-Analyse** | | | |
| 1 | Das Dokument „Anforderungen(XYZ)“ (also dieses Dokument) mit funktionalen, nicht-funktionalen Anforderungen und konzeptionellem Datenmodell. | * Siehe Vorlage. * Wird bei Projektbeginn mit einer Überblick gebenden Systembeschreibung an das Team ausgegeben. Das Dokument ist vom Team weiterzuführen und wieder abzugeben. | * Bitte auf Abgabetermin während des Semesters achten (s. Projektkalender). * Rechtzeitig vor Abgabe auf Qualitätssicherung achten (Review) * Die Dokumentation ist für die Weiterentwicklung in künftigen SW-Projekten besonders wichtig (Qualität!) |
| **Dokumentation des Systementwurfs** | | | |
| 2 | Das Dokument „Systementwurf(XYZ)“. | * Siehe Vorlage. | * Bitte auf Abgabetermin während des Semesters achten (s. Projektkalender) * Rechtzeitig vor Abgabe auf Qualitätssicherung achten (Review) * Die Dokumentation ist für die Weiterentwicklung in künftigen SW-Projekten besonders wichtig (Qualität!) |
| **Implementierung** | | | |
| 3 | Lauffähiger und getesteter Quellcode (incl. Entwicklerdokumentation) |  | Abgabe am Semesterende   * Die Dokumentation ist für die Weiterentwicklung in künftigen SW-Projekten besonders wichtig (Qualität!) |
| **Test** | | | |
| 4 | Testspezifikation Systemtest | * Siehe Vorlage | * Endgültige Abgabe am Semesterende; zur Vorbereitung des Abnahmetests ist die Aufstellung der in den Abnahmetest einbezogenen Testfälle früher vorzulegen (Termin im Projektkalender) |
| 5 | Testprotokoll Systemtest | * Siehe Vorlage | * Abgabe am Semesterende |

# Abnahmekriterien

In der Veranstaltung „Software-Projekte“ werden vom „Auftraggeber“ (in Absprache mit den Teilnehmern) rechtzeitig vor Semesterende Systemtestfälle ausgewählt, die das System dann am Tag der Abnahme ohne Beanstandung „überstehen“ muss.

# Abkürzungsverzeichnis

|  |  |
| --- | --- |
| Abkürzung | Erklärung |
|  |  |

# Literaturverzeichnis

**[Lar]** Larman Craig, *Applying UML And Patterns. An Introduction to Object-Oriented Analysis And Design,* Prentice Hall, 2nd ed., 2002

# Abbildungsverzeichnis

1. V-Modell® ist eine geschützte Marke der Bundesrepublik Deutschland. [↑](#footnote-ref-2)