- Systemanalyse -

Lastenheft für „Geheime Kommunikation mit Quantenkryptographie - eine Lernsoftware zum Verstehen der Grundlagen“

Version: 1.0

|  |  |  |
| --- | --- | --- |
| Projektbezeichnung | QuaKrypto | |
| Projektleiter | Simon Schisslbauer | |
| Verantwortlich | Team | |
| Erstellt am | 30.03.2023 | |
| Zuletzt geändert | 17.04.2023 23:12 | |
| Bearbeitungszustand | X | in Bearbeitung |
|  | vorgelegt |
|  | fertig gestellt |
| Dokumentablage |  | |

Änderungsverzeichnis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Änderung | | | Geänderte Kapitel | Beschreibung der Änderung | Autor | Zustand |
| Nr. | Datum | Version |
| 1 |  | 1.0 | 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](#_Toc34239361)

[2 Ausgangssituation und Zielsetzung 4](#_Toc34239362)

[3 Funktionale Anforderungen 5](#_Toc34239363)

[3.1 Use-Case Übersicht 5](#_Toc34239364)

[3.2 Use-Beschreibungen 5](#_Toc34239365)

[3.3 (Sonstige) Funktionalität 7](#_Toc34239366)

[3.4 Modell des Problembereichs (Konzeptionelles Datenmodell) 7](#_Toc34239367)

[4 Nicht-Funktionale Anforderungen 7](#_Toc34239368)

[4.1 Benutzbarkeit (Usability) 7](#_Toc34239369)

[4.2 Zuverlässigkeit (Reliability) 8](#_Toc34239370)

[4.3 Leistung (Performance) 8](#_Toc34239371)

[4.4 Unterstützbarkeit (Supportability) 9](#_Toc34239372)

[4.5 Sonstige Einschränkungen 9](#_Toc34239373)

[5 Risikoakzeptanz 10](#_Toc34239374)

[6 Skizze der Gesamtsystemarchitektur 10](#_Toc34239375)

[7 Lieferumfang 11](#_Toc34239376)

[8 Abnahmekriterien 11](#_Toc34239377)

[9 Glossar 11](#_Toc34239378)

[10 Abkürzungsverzeichnis 12](#_Toc34239379)

[11 Literaturverzeichnis 12](#_Toc34239380)

[12 Abbildungsverzeichnis 12](#_Toc34239381)

# Einleitung

Dieses Dokument enthält alle an das zu entwickelnde System gestellten Anforderungen. Die Gliederung orientiert sich am Aufbau des V-Modell-XT®[[1]](#footnote-1)-Produkts „Anforderungen (Lastenheft)“, ist jedoch zur Verwendung für die Veranstaltung **„Software-Projekte“** in Informatik-Curricula 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, was ungefähr dem Thema „Ausgangssituation und Zielsetzung“ in diesem Dokument entspricht; die Anforderungen müssen die Teilnehmer dann in enger Abstimmung mit ihrem „Auftraggeber“ selbst erarbeiten und in diesem Dokument niederlegen. Dadurch sollen sie Gelegenheit erhalten, auch Tätigkeiten der System-Analyse intensiver zu üben. Die „Auftraggeberseite“ liefert also nicht – wie im V-Modell-XT vorgesehen - das komplette Lastenheft, aus dem die „Auftragnehmerseite“ ein separates Pflichtenheft ableitet; stattdessen wird das hier vorliegende Dokument vom studentischen Entwicklerteam zur Dokumentation der Analyse-Ergebnisse erstellt und zugleich als Ersatz für die im V-Modell-XT vorgesehenen Dokumente Lasten- und Pflichtenheft verwendet.

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.

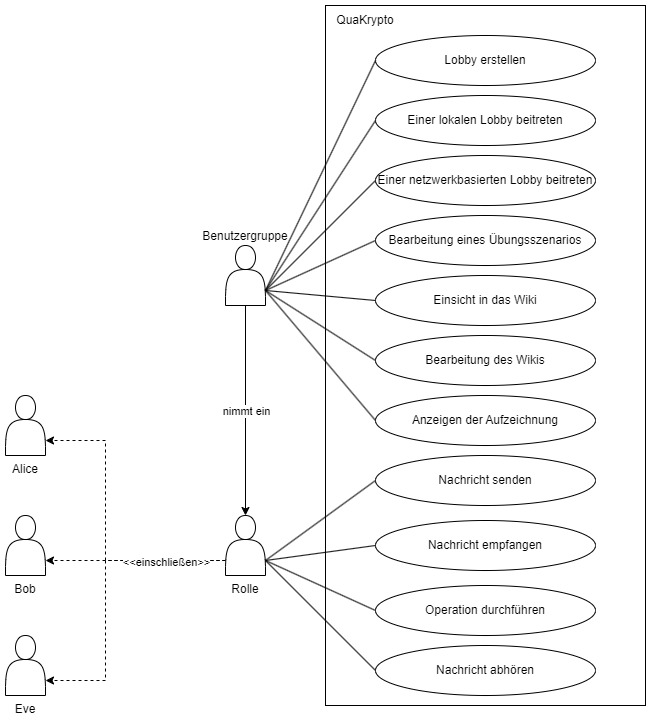
Im Allgemeinen sollten keine technischen Lösungen vorgegeben werden, um Architekten und Entwickler bei der Suche nach optimalen technischen Lösungen nicht einzuschränken.

# Ausgangssituation und Zielsetzung

…hier Ihren Text einfügen…

# Funktionale Anforderungen

## Use-Case Übersicht



## Use-Beschreibungen

|  |  |  |
| --- | --- | --- |
| **Erstellen einer Lobby** | | |
| **Kennung** | | UC-1 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Eine Benutzergruppe wählt:   * das Protokoll, das sie durch das System erlernen will * ob das Übungsszenario lokal oder netzwerkbasiert durchgeführt werden soll * die gewünschte Variante * die gewünschte(n) Phase(n) * den gewünschten Schwierigkeitsgrad   im Lobbykonfigurationsmenü aus. | | |
| **Vorbedingung(en):** | | |
| * Die Benutzergruppe befindet sich im Lobbykonfigurationsmenü | | |
| **Nachbedingung(en):** | | |
| * Eine Lobby wurde erstellt * Das System öffnet die erstellte Lobby | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Benutzergruppe das Protokoll auswählt, welches sie durch das System erlernen will 2. Die Benutzergruppe wählt aus, ob das Übungsszenario lokal oder netzwerkbasiert bearbeitet werden soll 3. Die Benutzergruppe wählt die gewünschte Variante aus 4. Die Benutzergruppe wählt die gewünschte(n) Phase(n) des Protokollablaufs aus 5. Die Benutzergruppe wählt den gewünschten Schwierigkeitsgrad aus 6. Die Benutzergruppe bestätigt ihre Eingaben 7. Das System prüft, ob die Eingaben der Benutzergruppe gültig sind 8. Das System übernimmt die Eingabedaten und erstellt eine Lobby   **Ende**. | |
| **Ablauf-Varianten:** | | |
| 7a | Die Eingaben der Benutzergruppe sind ungültig | |
|  | 1. Das System weist die Benutzergruppe auf die ungültigen Eingaben hin   **Ende.** | |
| 8a | Die Benutzergruppe hat die netzwerkbasierte Bearbeitung für das Übungsszenario gewählt | |
|  | 1. Das System legt eine netzwerkbasierte Lobby an   **Ende.** | |
| 8b | Die Benutzergruppe hat die lokale Bearbeitung für das Übungsszenario gewählt | |
|  | 1. Das System legt eine lokale Lobby an   **Ende.** | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Beitritt zu einer lokalen Lobby** | | |
| **Kennung** | | UC-2 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Die Benutzergruppe tritt einer vorhandenen, lokalen Lobby bei.  Dabei legt sie…   * ihre Rolle(n) * ihren Alias * ihr Passwort   fest.  Jede im Übungsszenario vorhandene Rolle (Alice, Bob, Eve) muss von genau einer Benutzergruppe repräsentiert werden. | | |
| **Vorbedingung(en):** | | |
| * Eine lokale Lobby wurde erstellt * Das System befindet sich in der Lobby | | |
| **Nachbedingung(en):** | | |
| * Die Benutzergruppe ist der Lobby beigetreten | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Benutzergruppe die Aktion ausführt einer vorhandenen Lobby beizutreten 2. Die Benutzergruppe wählt mindestens eine der noch vorhandenen Rollen aus 3. Das System sperrt die Auswahl der gewählten Rolle(n) für andere Benutzergruppen 4. Die Benutzergruppe legt ihren Alias fest 5. Die Benutzergruppe legt ihr Passwort fest 6. Das System prüft, ob die Eingaben gültig sind   **Ende**. | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Beitritt zu einer netzwerkbasierten Lobby** | | |
| **Kennung** | | UC-3 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Die Benutzergruppe tritt einer vorhandenen, netzwerkbasierten Lobby bei.  Dabei legt sie…   * die gewünschte Lobby * ihre Rolle(n) * ihren Alias * ihr Passwort   fest.  Jede im Übungsszenario vorhandene Rolle (Alice, Bob, Eve) muss von genau einer Benutzergruppe repräsentiert werden. | | |
| **Vorbedingung(en):** | | |
| * Im selben Netzwerk wurde mindestens eine netzwerkbasierte Lobby erstellt * Die Benutzergruppe befindet sich im Lobbyauswahlmenü | | |
| **Nachbedingung(en):** | | |
| * Die Benutzergruppe ist der Lobby beigetreten | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Benutzergruppe die Aktion ausführt einer vorhandenen Lobby beizutreten 2. Die Benutzergruppe wählt im Lobbyauswahlmenü die gewünschte Lobby aus 3. Das System prüft, ob die Lobby weiterhin verfügbar ist 4. Die Benutzergruppe wird zur gewünschten Lobby hinzugefügt 5. Die Benutzergruppe wählt mindestens eine der noch vorhandenen Rollen aus 6. Das System sperrt die Auswahl der gewählten Rolle(n) für andere Benutzergruppen 7. Die Benutzergruppe legt ihren Alias fest 8. Die Benutzergruppe legt ihr Passwort fest 9. Das System prüft, ob die Eingaben gültig sind   **Ende**. | |
| **Ablauf-Varianten:** | | |
| 3a | Die ausgewählte Lobby ist nicht mehr existent | |
|  | 1. Das System weist die Benutzergruppe darauf hin, dass die angegebene Lobby nicht gefunden werden konnte   **Rückkehr nach**: 2 | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Bearbeitung eines Übungsszenarios** | | |
| **Kennung** | | UC-4 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Alle vorhandenen Rollen durchlaufen die ihnen zugeordneten Handlungsschritte. | | |
| **Vorbedingung(en):** | | |
| * Eine Lobby wurde erstellt * Jede in der Lobby vorhandene Rolle ist von genau einer Benutzergruppe belegt | | |
| **Nachbedingung(en):** | | |
| * Eine Aufzeichnung der Bearbeitung des Übungsszenarios wurde erstellt | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn der Lobbyhost das Übungsszenario startet 2. Das System initialisiert das Übungsszenario und die Aufzeichnung des Übungsszenarios 3. Der Rolle werden dem Zug entsprechende Handlungsschritte angezeigt 4. Die Rolle führt die gewünschten Handlungsschritte aus 5. Das System speichert die Bearbeitung des Übungsszenarios in einer Aufzeichnung ab 6. Die Rolle bestätigt das Ende ihres Zuges durch einen entsprechenden Handlungsschritt 7. Das System gibt die Steuerung für die nächste Rolle frei 8. Dieser Vorgang (Schritte 3 bis 8) wird so lange wiederholt, bis die aktuelle Phase abgeschlossen wurde 9. Dieser Vorgang (Schritte 3 bis 9) wird so lange wiederholt, bis die letzte durch den Lobbyhost ausgewählte Phase abgeschlossen wurde   **Ende**. | |
| **Ablauf-Varianten:** | | |
| 3a | Es handelt sich um ein lokales Übungsszenario, wobei mindestens zwei Benutzergruppen existieren | |
|  | 1. Die Rolle entsperrt vor ihrem Zug ihre Informationen und ihre Handlungsschritte mit ihrem Passwort   **Rückkehr nach**: 4 | |
| 6a | Es handelt sich um ein lokales Übungsszenario, wobei mindestens zwei Benutzergruppen existieren | |
|  | 1. Das System sperrt die Informationen und Handlungsschritte der jeweiligen Rolle   **Rückkehr nach**: 7 | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Einsicht in das Wiki** | | |
| **Kennung** | | UC-5 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Um ihr Wissen und ihre Kenntnisse zu vertiefen, soll die Benutzergruppe ein Wiki mit Informationen über das Protokoll einsehen können. | | |
| **Vorbedingung(en):** | | |
| * Es existiert ein Wiki | | |
| **Nachbedingung(en):** | | |
| * Das Wiki ist geöffnet | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Benutzergruppe die Aktion zum Öffnen des Wikis ausführt 2. Das System zeigt der Benutzergruppe das Wiki mit allen Wiki-Einträgen an 3. Die Benutzergruppe wählt einen Wiki-Eintrag der angezeigten Wiki-Einträge aus 4. Das System zeigt der Benutzergruppe ihren ausgewählten Wiki-Eintrag an   **Ende**. | |
| **Ablauf-Varianten:** | | |
|  |  | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Bearbeitung des Wikis** | | |
| **Kennung** | | UC-6 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Die Benutzergruppe soll das vorhandene Wiki bearbeiten können. | | |
| **Vorbedingung(en):** | | |
| * Das Wiki ist geöffnet | | |
| **Nachbedingung(en):** | | |
| * Die Wiki-Einträge wurden gemäß der Benutzereingaben verändert * Die Änderungen am Wiki wurden abgespeichert | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Benutzergruppe die Aktion zum Bearbeiten des Wikis ausführt 2. Die Benutzergruppe wählt aus, ob sie einen Wiki-Eintrag bearbeiten, löschen oder hinzufügen will 3. Das System speichert die von der Benutzergruppe getätigten Änderungen am Wiki   **Ende**. | |
| **Ablauf-Varianten:** | | |
| 2a | Die Benutzergruppe will einen Wiki-Eintrag bearbeiten | |
|  | 1. Die Benutzergruppe wählt den zu bearbeitenden Wiki-Eintrag aus 2. Das System lädt den zu bearbeitenden Wiki-Eintrag 3. Die Benutzergruppe tätigt ihre Änderungen an dem gewählten Wiki-Eintrag 4. Die Benutzergruppe bestätigt die getätigten Änderungen am gewählten Wiki-Eintrag   **Rückkehr nach**: 3 | |
| 2b | Die Benutzergruppe will einen Wiki-Eintrag löschen | |
|  | 1. Die Benutzergruppe wählt den zu löschenden Wiki-Eintrag aus 2. Die Benutzergruppe bestätigt die Löschung des gewählten Wiki-Eintrags   **Rückkehr nach**: 3 | |
| 2c | Die Benutzergruppe will einen Wiki-Eintrag hinzufügen | |
|  | 1. Das System lädt einen neuen Wiki-Eintrag 2. Die Benutzergruppe fügt Informationen im neuen Wiki-Eintrag ein 3. Die Benutzergruppe bestätigt die Erstellung des neuen Wiki-Eintrags   **Rückkehr nach**: 3 | |
| **Spezielle Anforderungen:** | | |
| Die Benutzergruppe soll die Bearbeitung jederzeit abbrechen können | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Anzeigen der Aufzeichnung** | | | |
| **Kennung** | | | UC-7 |
| **Priorität** | | | Gering |
| **Kurzbeschreibung:** | | | |
| Das System soll die Aufzeichnung des Übungsszenarios anzeigen. | | | |
| **Vorbedingung(en):** | | | |
| * Die Bearbeitung eines Übungsszenarios ist abgeschlossen * Das System hat die Bearbeitung des Übungsszenarios aufgezeichnet | | | |
| **Nachbedingung(en):** | | | |
| * Der Benutzergruppe wurden die aufgezeichneten Handlungsschritte der Bearbeitung des Übungsszenarios angezeigt | | | |
| **Normaler Ablauf:** | | | |
|  | 1. Der Anwendungsfall beginnt, wenn das Übungsszenario abgeschlossen ist 2. Das System zeigt den Benutzergruppen die Aufzeichnung an 3. Die Benutzergruppe schließt die Aufzeichnung   **Ende**. | | |
| **Ablauf-Varianten:** | | | |
| 2a | | Es handelt sich um ein lokales Übungsszenario | |
|  | | 1. Allen Benutzergruppen wird abschließend zum Übungsszenario die Aufzeichnung an einem Rechner angezeigt   **Rückkehr nach**: 3 | |
| 2b | | Es handelt sich um ein netzwerkbasiertes Übungsszenario | |
|  | | 1. Allen Benutzergruppen wird abschließend zum Übungsszenario die Aufzeichnung am jeweiligen Rechner angezeigt   **Rückkehr nach**: 3 | |
| **Spezielle Anforderungen:** | | | |
|  | | | |
| **Zu klärende Punkte:** | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **Nachricht senden** | | |
| **Kennung** | | UC-8 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Die Rollen sollen Nachrichten in Form von Photonen oder Bitfolgen senden können. | | |
| **Vorbedingung(en):** | | |
| * Es wird ein Übungsszenario bearbeitet * Die Rolle, welche eine Nachricht senden möchte, ist am Zug | | |
| **Nachbedingung(en):** | | |
| * Die Nachricht wurde gesendet und ist auf dem entsprechenden Übertragungskanal gespeichert und kann ausgelesen werden | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Rolle den Handlungsschritt zum Senden einer Nachricht ausführt 2. Die Rolle wählt die zu sendende(n) Information(en) aus 3. Die Rolle wählt den Empfänger der Nachricht aus 4. Das System sendet die gewählte(n) Information(en) in Form einer Nachricht auf dem jeweiligen Übertragungskanal   **Ende**. | |
| **Ablauf-Varianten:** | | |
|  | | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nachricht empfangen** | | |
| **Kennung** | | UC-9 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Die Rollen sollen Nachrichten in Form von Photonen oder Bitfolgen empfangen können. | | |
| **Vorbedingung(en):** | | |
| * Es wird ein Übungsszenario bearbeitet * Die Rolle, welche eine Nachricht empfangen möchte, ist am Zug | | |
| **Nachbedingung(en):** | | |
| * Die Nachricht wurde empfangen und wird angezeigt * Die empfangene Nachricht wurde aus dem Übertragungskanal entfernt | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Rolle den Handlungsschritt zum Empfangen einer Nachricht ausführt 2. Das System zeigt an, auf welchem Übertragungskanal Informationen verfügbar sind 3. Die Rolle wählt aus, von welchem Übertragungskanal sie Informationen empfangen will 4. Die Nachricht wird gemäß der Benutzereingaben ausgelesen 5. Die Nachrichten werden abgespeichert   **Ende**. | |
| **Ablauf-Varianten:** | | |
| 3a | Die Rolle möchte Informationen vom Photonenübertragungskanal empfangen | |
|  | 1. Die Rolle wählt vorhandene Polarisationsschemata als Detektor für die Photonenfolge aus   **Rückkehr nach**: 4 | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nachricht abhören** | | |
| **Kennung** | | UC-10 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Eve soll Nachrichten in Form von Photonen oder Bitfolgen abhören können. | | |
| **Vorbedingung(en):** | | |
| * Es wird ein Übungsszenario bearbeitet * Eve ist am Zug | | |
| **Nachbedingung(en):** | | |
| * Die Nachricht wurde abgehört und angezeigt * Die abgehörte Nachricht wurde an den ursprünglichen Empfänger weitergesendet | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn Eve den Handlungsschritt zum Abhören einer Nachricht ausführt 2. Das System zeigt an, auf welchem Übertragungskanal Informationen verfügbar sind 3. Eve wählt aus, auf welchem Übertragungskanal sie Informationen abhören will | |
| **Ablauf-Varianten:** | | |
| 3a | Eve möchte Informationen vom Photonenübertragungskanal abhören | |
|  | 1. Eve wählt vorhandene Polarisationsschemata als Detektor für die Photonenfolge aus 2. Die Nachricht wird gemäß der Benutzereingaben ausgelesen 3. Die interpretierten Nachrichten werden abgespeichert 4. Das System sendet die abgehörten Photonen an den ursprünglichen Empfänger   **Ende.** | |
| 3b | Eve möchte Bitfolgen vom Bitübertragungskanal abhören | |
|  | 1. Die Nachrichten auf dem Bitübertragungskanal werden abgespeichert 2. Das System zeigt an, welche Bitfolgen abgehört wurden 3. Das System sendet die abgehörten Bitfolgen an den ursprünglichen Empfänger   **Ende.** | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Generierung einer Information** | | |
| **Kennung** | | UC-11 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Die Rolle soll Informationen generieren können. | | |
| **Vorbedingung(en):** | | |
| * Die Rolle, welche Informationen generieren möchte, ist am Zug * Die Rolle verfügt über die für die Generierung benötigten Informationen | | |
| **Nachbedingung(en):** | | |
| * Die generierten Informationen werden der Rolle angezeigt * Die generierten Informationen sind abgespeichert | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Rolle die Aktion zur Generierung der gewünschten Information ausführt 2. Das System generiert die gewünschte Information 3. Die Rolle benennt die generierte Information | |
| **Ablauf-Varianten:** | | |
| 2a | Die zu generierende Information ist eine Bitfolge | |
|  | 1. Die Rolle wählt eine beliebige Zahl aus, mit der eine zufällige Bitfolge generiert werden soll 2. Das System generiert eine zufällige Bitfolge   **Rückkehr nach**: 3 | |
| 2a1a | Die Rolle gibt die Bitfolge an | |
|  | 1. Die Rolle gibt die Bitfolge ein 2. Das System gibt der Rolle die Möglichkeit die eingegebene Bitfolge anzupassen   **Rückkehr nach**: 3 | |
| 2b | Die zu generierende Information sind Polarisationsschemata | |
|  | 1. Die Rolle wählt eine beliebige Zahl aus, mit der zufällige Polarisationsschemata generiert werden soll 2. Das System generiert zufällige Polarisationsschemata   **Rückkehr nach**: 3 | |
| 2b1a | Die Rolle gibt die Polarisationsschemata an | |
|  | 1. Die Rolle gibt die Polarisationsschemata ein 2. Das System gibt der Rolle die Möglichkeit die eingegebenen Polarisationsschemata anzupassen   **Rückkehr nach**: 3 | |
| 2c | Die zu generierende Information sind Photonen | |
|  | 1. Die Rolle wählt vorhandene Polarisationsschemata als Emitter für die Photonenfolge aus 2. Die Rolle wählt die Bitfolge, aus welcher die Photonen generiert werden sollen, aus 3. Das System berechnet anhand dieser beiden Eingaben die zu sendende Photonenfolge   **Rückkehr nach**: 3 | |
| 2d | Die zu generierende Information ist eine Bit-Maske | |
|  | 1. Die Rolle wählt die Länge und Anzahl der Einsen der Bit-Maske aus 2. Das System generiert die Bit-Maske   **Rückkehr nach**: 3 | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Vergleich von Informationen** | | |
| **Kennung** | | UC-12 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Die Rolle soll Informationen vergleichen können. | | |
| **Vorbedingung(en):** | | |
| * Die Rolle, welche Informationen vergleichen möchte, ist am Zug * Die Rolle verfügt über die zu vergleichenden Informationen | | |
| **Nachbedingung(en):** | | |
| * Die Informationen wurden verglichen * Das System zeigt der Rolle das Ergebnis des Vergleichs an * Das Ergebnis des Vergleichs ist abgespeichert | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Rolle die Aktion zum Vergleich der gewünschten Information ausführt 2. Das System vergleicht die gewünschten Informationen 3. Die Rolle benennt das Ergebnis des Vergleichs | |
| **Ablauf-Varianten:** | | |
| 2a | Es sollen zwei Polarisationsschemata verglichen werden | |
|  | 1. Die Rolle gibt die zu vergleichenden Polarisationsschemata an 2. Das System vergleicht die beiden angegebenen Polarisationsschemata   **Rückkehr nach**: 3 | |
| 2b | Es sollen zwei Bitfolgen verglichen werden | |
|  | 1. Die Rolle gibt die zu vergleichenden Bitfolgen an 2. Das System vergleicht die beiden angegebenen Bitfolgen   **Rückkehr nach**: 3 | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Umwandlung von Informationen** | | |
| **Kennung** | | UC-13 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Die Rolle soll Informationen umwandeln können. | | |
| **Vorbedingung(en):** | | |
| * Die Rolle, welche Informationen umwandeln möchte, ist am Zug * Die Rolle verfügt über die umzuwandelnden Informationen | | |
| **Nachbedingung(en):** | | |
| * Die Informationen wurden umgewandelt * Das System zeigt der Rolle das Ergebnis der Umwandlung an * Das Ergebnis der Umwandlung ist abgespeichert | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Rolle die Aktion zum Umwandeln der gewünschten Informationen ausführt 2. Das System wandelt die gewünschten Informationen um 3. Die Rolle benennt die umgewandelte Information | |
| **Ablauf-Varianten:** | | |
| 2a | Umwandlung von Photonen zu einer Bitfolge | |
|  | 1. Die Rolle gibt die Photonenfolge an 2. Das System berechnet anhand dieser Eingabe die Bitfolge   **Rückkehr nach**: 3 | |
| 2b | Umwandlung eines Textes in eine verschlüsselte Nachricht | |
|  | 1. Die Rolle wählt einen vorhandenen Text, der verschlüsselt werden soll, aus 2. Die Rolle wählt die Bitfolge, mit der der Text verschlüsselt werden soll, aus 3. Das System berechnet anhand dieser beiden Eingaben den verschlüsselten Text   **Rückkehr nach**: 3 | |
| 2c | Umwandlung einer verschlüsselten Nachricht in einen Text | |
|  | 1. Die Rolle wählt einen vorhandenen Text, der entschlüsselt werden soll, aus 2. Die Rolle wählt die Bitfolge, mit der der Text entschlüsselt werden soll, aus 3. Das System berechnet anhand dieser beiden Eingaben den entschlüsselten Text   **Rückkehr nach**: 3 | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Veränderung von Informationen** | | |
| **Kennung** | | UC-14 |
| **Priorität** | | Hoch |
| **Kurzbeschreibung:** | | |
| Die Rolle soll Informationen verändern können. | | |
| **Vorbedingung(en):** | | |
| * Die Rolle, welche Informationen verändern möchte, ist am Zug * Die Rolle verfügt über die zu verändernden Informationen | | |
| **Nachbedingung(en):** | | |
| * Die Informationen wurden verändert * Das System zeigt der Rolle das Ergebnis der Veränderung an * Das Ergebnis der Veränderung ist abgespeichert | | |
| **Normaler Ablauf:** | | |
|  | 1. Der Anwendungsfall beginnt, wenn die Rolle die Aktion zum Verändern der gewünschten Informationen ausführt 2. Das System verändert die gewünschten Informationen 3. Die Rolle benennt die veränderte Information | |
| **Ablauf-Varianten:** | | |
| 2a | Die Veränderung ist ein Streichen von Bits aus einer Bitfolge | |
|  | 1. Die Rolle gibt die Bit-Maske, nach der die Bitstellen aus der Bitfolge gestrichen werden sollen, an 2. Die Rolle gibt die Bitfolge an 3. Die Rolle gibt den zu streichenden Bit-Wert an 4. Das System streicht aus der Bitfolge die Bitstellen gemäß der Bit-Maske   **Rückkehr nach**: 3 | |
| **Spezielle Anforderungen:** | | |
|  | | |
| **Zu klärende Punkte:** | | |
| Zu 2a 3.: Die Rolle gibt an, welche Bits (0en oder 1en) durch Anwendung der Bitmaske gestrichen werden sollen. | | |

### Hinweise zu den Use-Cases

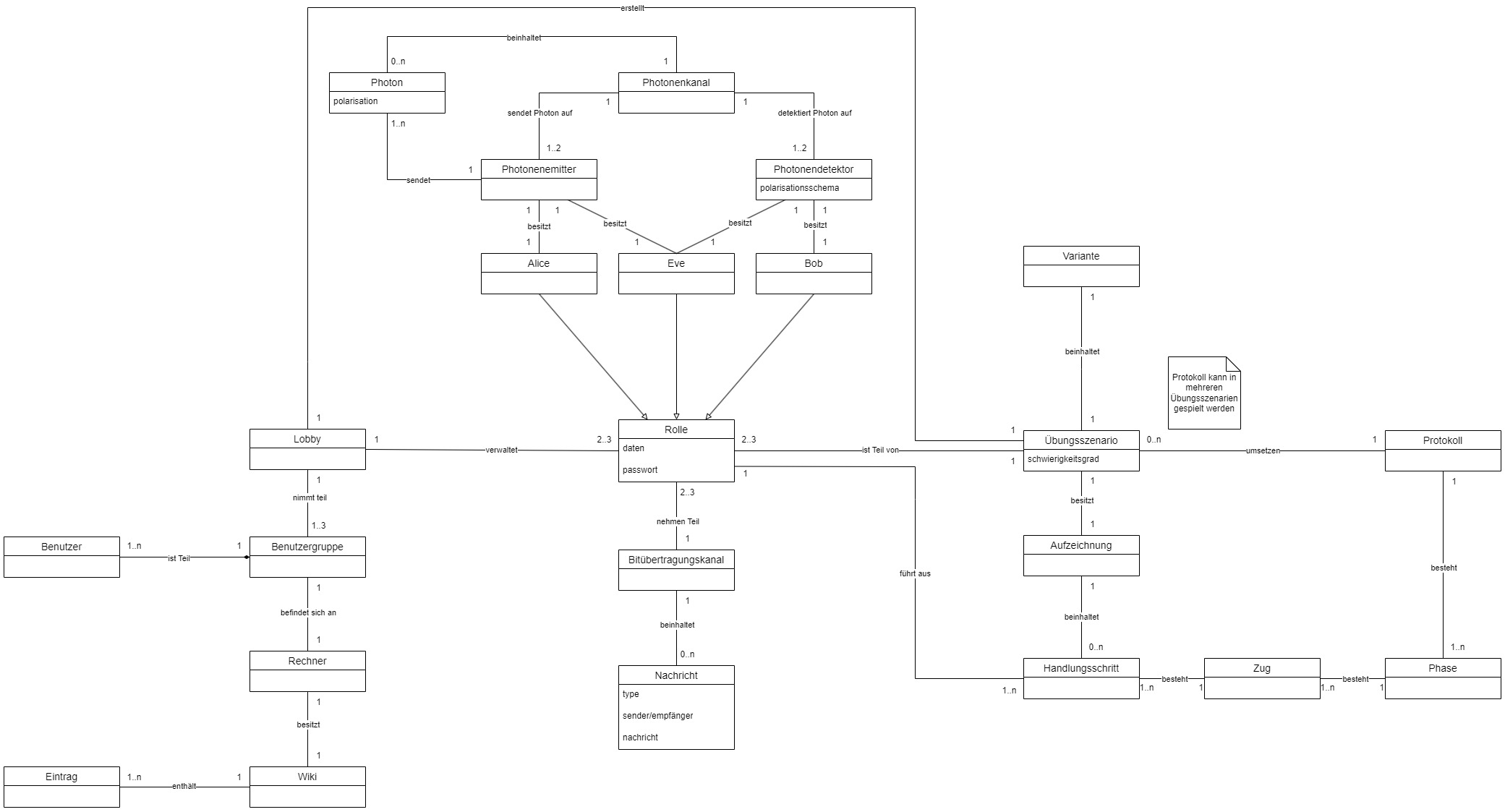
* Auf den Use-Case „One-Time-Pad vereinbaren“ wurde verzichtet!

## (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)



# 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 „Systemanalyse(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) |
| **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) |
| **Implementierung** | | | |
| 3 | Lauffähiger und getesteter Quellcode |  | Abgabe am Semesterende |
| **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) |

# 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.

# Glossar

|  |  |
| --- | --- |
| Begriff | Erklärung |
|  |  |

# 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-1)