

Softwaretechnik

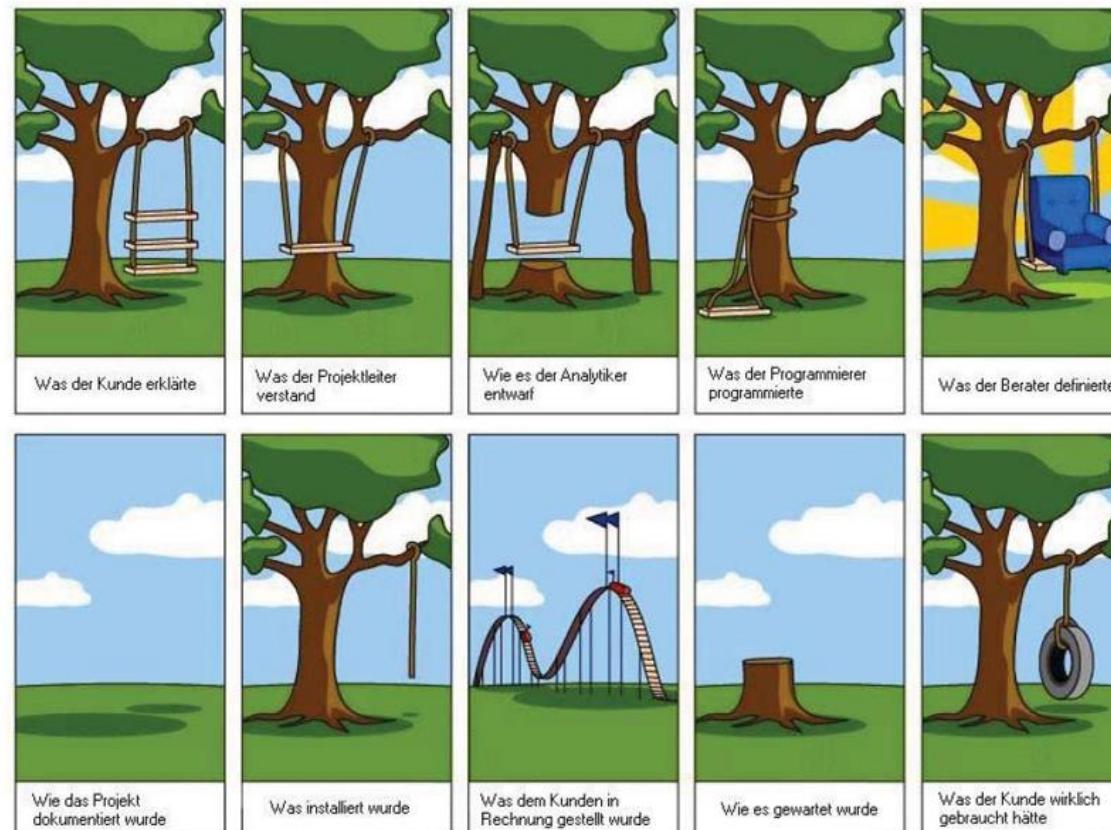
Prof. Dr. Raphael Herding

Ausgangssituation und Motivation

- **Fiktive Aufgabenstellung:** Der Auftraggeber wünscht sich eine originelle Schaukel im eigenen Garten, die etwas anders ist als die des Nachbarn, sicheres Material verwendet, den Kindern Spaß macht und wirklich robust ist.

Ausgangssituation und Motivation

- **Fiktive Aufgabenstellung:** Der Auftraggeber wünscht sich eine originelle Schaukel im eigenen Garten, die etwas anders ist als die des Nachbarn, sicheres Material verwendet, den Kindern Spaß macht und wirklich robust



Requirements Engineering - Anforderungsermittlung

- Kundenfokus ist essentiell.
- Festlegung der Systemkomponenten und der dazugehörigen Funktionsbausteine.
- Die Systemanforderungen bestehen aus der Beschreibung der Systemkomponenten und deren Einschränkungen die innerhalb des Requirements-Engineering-Prozesses ermittelt werden.

Case Study: Mentcare

Ein Patienteninformationssystem für die psychische Gesundheitsversorgung

- Ein medizinisches Informationssystem, das Informationen über Patienten mit psychischen Erkrankungen und deren Behandlungen verwaltet.
- Die meisten Patienten benötigen keine stationäre Behandlung, sondern besuchen regelmäßig spezialisierte Kliniken.
- Kliniken werden nicht nur in Krankenhäusern betrieben, sondern auch in Arztpraxen oder Gemeindezentren.

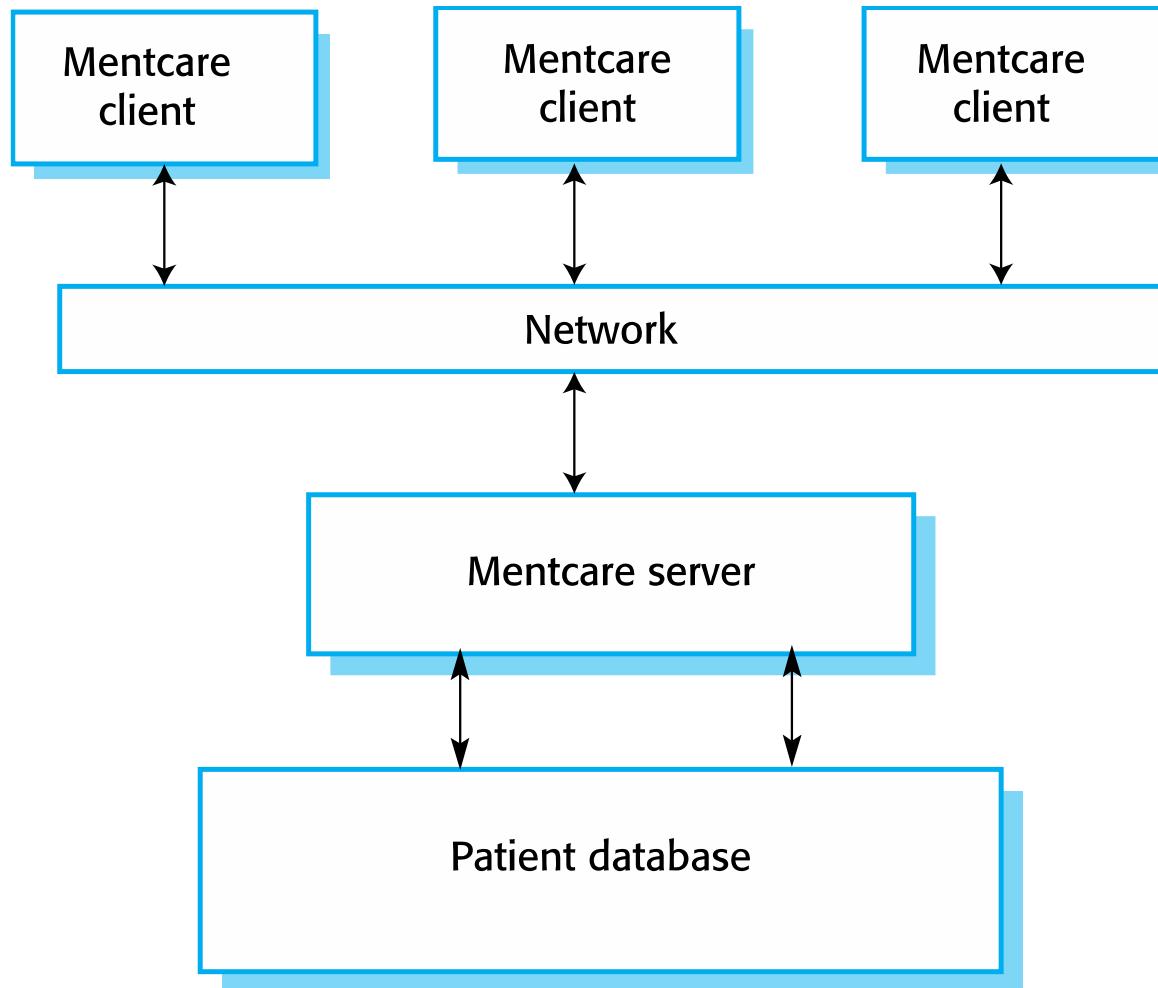
Case Study: Mentcare

Beschreibung und Ziele

- Informationssystem für den Einsatz in Kliniken.
 - Nutzen einer zentralen Patientendatenbank, aber auch offline auf PCs nutzbar.
 - Lokale Kopien von Patientendaten können bei fehlender Netzwerkverbindung genutzt werden.
-
- Ziele:
 - Erstellung von Managementinformationen zum Vergleich mit lokalen und staatlichen Zielvorgaben.
 - Bereitstellung zeitnäher Informationen für medizinisches Personal zur Unterstützung der Patientenbehandlung.

Case Study: Mentcare

Grobarchitektur



Case Study: Mentcare

Zentrale Funktionen

- **Individuelles Fallmanagement:**
 - Ärzte können Patientendaten anlegen, bearbeiten und Patientenhistorie einsehen.
 - Unterstützung von Datenauswertungen, um schnell Überblick über Probleme und Behandlungen zu erhalten.
- **Patientenüberwachung:**
 - Warnmeldungen bei möglichen Problemen während der Behandlung.
- **Administrative Berichterstellung:**
 - Monatliche Managementberichte zu Patientenanzahl, Neuaufnahmen, Entlassungen, Einweisungen und verschriebenen Medikamenten mit Kosten.

Case Study: Mentcare

Herausforderungen

- **Datenschutz:**
 - Patientendaten müssen streng vertraulich bleiben und nur autorisiertem Personal sowie dem Patienten zugänglich sein.
- **Sicherheit:**
 - Warnung vor potenziell suizidalen oder gefährlichen Patienten.
- **Systemverfügbarkeit**
 - Ist entscheidend für sichere Medikamentenverschreibung.

Klassifizierung von Anforderungen

- **funktionale + nicht-funktionale Anforderungen** - beschreiben die geforderte Funktionalität, das Außenverhalten des zu erstellenden Systems.
- **Qualitätsanforderungen** - präzisieren wichtige Qualitätskriterien (Zuverlässigkeit, Effizienz, Benutzungsfreundlichkeit, etc.).
- **technische Anforderungen** - beschreiben technische Rahmenbedingungen, denen das System genügen muss.
- **Validitäts- und Wartungsanforderungen** - beschreiben Anforderungen des Abnahmetests und Wartungskonditionen.
- **Realisierungsanforderungen** - Festlegung des Vorgehensmodells, gewünschter Dokumentation, einzuhaltender Vorschriften sowie Richtlinien und Normen.

Was ist eine Anforderung?

- Sowohl abstrakte Beschreibung eines Dienstes oder einer Systembeschränkung bis hin zur detaillierten mathematischen Funktionsspezifikation.
- Formulierung korrekter Anforderungen ist unerlässlich
 - Sie können die Grundlage für ein Angebot (für einen Vertrag) sein - daher müssen sie offen für Interpretationen sein.
 - Sie können die Grundlage für den Vertrag selbst sein - daher müssen sie detaillierter definiert werden können.
- Beiden vorherigen Aussagen können als Anforderungen bezeichnet werden.

Anforderungstypen

- Benutzeranforderungen
 - Aussagen in natürlicher Sprache sowie Diagramme zu den Systemdiensten und zu der betrieblichen Einsatzumgebung
 - Für Kunden geschrieben
- Systemanforderungen
 - Ein strukturiertes Dokument mit detaillierten Beschreibungen der Funktionen, Dienste und der betrieblichen Einsatzumgebung
 - Legt fest, **was** implementiert werden soll und kann daher Teil eines Vertrags zwischen Kunde und Auftragnehmer sein

Anforderungstypen

Nutzeranforderungen (Auszug)

Nr.	Anforderung
N5	Ein Benutzer muss in der Lage sein, die Terminlisten für alle Kliniken zu durchsuchen.
N3	Als Klinikmitarbeiter möchte ich Patientenakten auch ohne bestehende Netzwerkverbindung einsehen können, damit die Behandlung auch bei Verbindungsproblemen fortgesetzt werden kann.

Systemanforderungen

Nr.	Anforderung
1	Das Mentcare-System soll monatliche Managementberichte erstellen, die die Kosten der von jeder Klinik im jeweiligen Monat verschriebenen Medikamente anzeigen.
1.2	Das System soll den Bericht am letzten Arbeitstag des Monats nach 17:30 Uhr zur Ausgabe (Druck) generieren.
1.4	Wenn Medikamente in unterschiedlichen Dosierungseinheiten (z. B. 10 mg, 20 mg usw.) verfügbar sind, wird für jede Dosierungseinheit ein separater Bericht erstellt.
1.5	Der Zugriff auf die Kostenberichte für Medikamente ist auf autorisierte Benutzer beschränkt, die in einer Verwaltungsliste für Zugriffsrechte aufgeführt sind.

Funktionale und nicht-funktionale Anforderungen

- **Funktionale Anforderungen**

- Aussagen über die Leistungen, die das System erbringen soll (Funktionen, Reaktion auf Eingaben, Reaktion in bestimmten Situationen)
- Sie können auch angeben, was das System nicht tun soll.

- **Nicht-funktionale Anforderungen**

- Beschränkungen der vom System angebotenen Dienste oder Funktionen (zeitliche Beschränkungen, Verfügbarkeit, Beschränkungen des Entwicklungsprozesses, Normen ...)

- **Domain Requirements** (Einsatzbereichsanforderungen)

- Einschränkungen für das System, die sich aus dem Einsatzbereich ergeben.

Funktionale Anforderungen

- Nutzer- und Systemanforderungen sind typischerweise funktionale Anforderungen.

Anforderungstypen

Nutzeranforderungen (Auszug)

Nr.	Anforderung
N5	Ein Benutzer muss in der Lage sein, die Terminlisten für alle Kliniken zu durchsuchen.
N3	Als Klinikmitarbeiter möchte ich Patientenakten auch ohne bestehende Netzwerkverbindung einsehen können, damit die Behandlung auch bei Verbindungsproblemen fortgesetzt werden kann.

Systemanforderungen

Nr.	Anforderung
1	Das Mentcare-System soll monatliche Managementberichte erstellen, die die Kosten der von jeder Klinik im jeweiligen Monat verschriebenen Medikamente anzeigen.
1.2	Das System soll den Bericht am letzten Arbeitstag des Monats nach 17:30 Uhr zur Ausgabe (Druck) generieren.
1.4	Wenn Medikamente in unterschiedlichen Dosierungseinheiten (z. B. 10 mg, 20 mg usw.) verfügbar sind, wird für jede Dosierungseinheit ein separater Bericht erstellt.
1.5	Der Zugriff auf die Kostenberichte für Medikamente ist auf autorisierte Benutzer beschränkt, die in einer Verwaltungsliste für Zugriffsrechte aufgeführt sind.

Ungenauigkeiten in funktionalen Anforderungen

Anforderungstypen

Nutzeranforderungen (Auszug)

Nr.	Anforderung
N5	Ein Benutzer muss in der Lage sein, die Terminlisten für alle Kliniken zu durchsuchen.
N3	Als Klinikmitarbeiter möchte ich Patientenakten auch ohne bestehende Netzwerkverbindung einsehen können, damit die Behandlung auch bei Verbindungsproblemen fortgesetzt werden kann.

- Bei ungenauer Formulierung können Missverständnisse entstehen.
- Mehrdeutige Anforderungen werden verschieden interpretiert.
- Beispiel: „durchsuchen“
 - **Benutzersicht** - Suche nach einem Patientennamen über alle Termine in allen Kliniken.
 - **Entwicklersicht** - Suche nach einem Patientennamen in einer einzelnen Klinik. Der Benutzer wählt eine Klinik und sucht dann.

Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen (Auszug)

Nr.	Anforderung
NF1	Datenschutz & Vertraulichkeit: Alle Patientendaten müssen gemäß geltenden Datenschutzgesetzen (z. B. DSGVO) verschlüsselt gespeichert und übertragen werden.
NF2	Verfügbarkeit: Das System muss eine Betriebszeit von mindestens 99,5 % pro Monat gewährleisten, um eine unterbrechungsfreie Versorgung zu ermöglichen.
NF3	Benutzerfreundlichkeit: Die Benutzeroberfläche soll so gestaltet sein, dass medizinisches Personal nach maximal 2 Stunden Einarbeitung selbstständig alle Kernfunktionen bedienen kann.

- **Anforderungen an das Produkt** – Produktverhalten (Schnelligkeit, Reaktion, Zuverlässigkeit, ...)
- **Anforderungen aus der Organisation** – Organisationspolitik (Prozessstandards, Code Convention, ...)
- **Externe Anforderungen** – Entwicklungsprozess (Regulierung, Interoperabilität, Gesetze, ...)

Lab

- Definieren Sie jeweils 3 funktionale und 3 nicht-funktionale Anforderungen an das Mentcare System.
 - Erfassen Sie diese zunächst grob in Analogie zu den Folien zuvor.

Requirements-Engineering-Prozess

- Der Prozess beschreibt das Systematische Erfassen von Anforderungen, so dass möglichst keien Missverständnisse entstehen.
- Genaue Prozessschritte sind nicht statisch.
- Es gibt jedoch allgemeingültige Aktivitäten innerhalb aller Prozesse.
 - **Erheben** von Anforderungen
 - **Analyse** der Anforderungen
 - **Validierung** von Anforderungen
 - **Anforderungsmanagement**
- In der Praxis ist Requirements Engineering eine **iterative Aktivität**, bei der diese Prozesse ineinander übergehen und iterativ wiederholt werden.

Erheben von Anforderungen

Probleme

- Zusammenarbeit mit Kunden (Anwendungsdomäne, Kundenumgebung, Wartung, ...)
- Phasen:
 - Ermittlung der Anforderungen,
 - Klassifizierung und Organisation der Anforderungen,
 - Priorisierung und Verhandlung von Anforderungen,
 - Spezifikation der Anforderungen.
- Probleme:
 - Kunde weiß nicht, was er wirklich will
 - Kunden formulieren Anforderungen in „ihrer“ Sprache
 - verschiedene Akteure des Kunden haben teils konkurrierende Anforderungen
 - politische und organisatorische Faktoren
 - Anforderungsänderungen durch neue oder ausgetauschte Akteure
 -

Erheben von Anforderungen

Methoden

- **Interview** führen
 - offen oder geschlossen,
 - formal oder informal
 - effektive Befragung - (aufgeschlossen sein, keine vorgefasste Meinung, zuhören, objektive Fragen, offene Diskussion)
- **Ethnografie** (Beobachtung)
 - Beobachtung der Leute bei der Arbeit (Hospitation, Shadowing)
 - keine Arbeitsbeschreibung durch Kunden nötig
 - zwischenmenschliche Interaktion sorgt für besseres Gesamtverständnis
 - Ableitung von „wie wird gearbeitet“ und nicht wie ist der formale Prozess
 - kann zur IST-Analyse genutzt werden. Neue Features können zumeist nicht abgeleitet werden

Erheben von Anforderungen

Methoden

- Entwicklung von **konkreten Stories und Szenarien**
 - Beispiele der Systemnutzung aus dem „wirklichen Leben“
 - Stories und Szenarien beschreiben, wie ein System für eine bestimmte Aufgabe genutzt werden kann
 - durch praxisnahe Situation können sich Stakeholder identifizieren und ihre Situation in Bezug auf die Story kommentieren
- **Workshop**
 - kleine Gruppe arbeitet intensiv an der Erfassung von Anforderungen
 - klarer Fokus und limitierte Zeit
 - es müssen die „richtigen“ Leute ausgewählt werden
 - Moderator muss Einhaltung des Fokus sicherstellen und bei Konflikten schlichten
 - intensive Vorbereitung nötig
 - Workshop-Teilnehmern muss Entscheidungskompetenz vorab erteilt werden
- **Häufig werden unterschiedliche Szenarien kombiniert**
 - Workshop für Grobermittlung; User Story und Interview für Feinspezifikation

Anforderungsspezifikation

- Prozess des Aufschreibens der Benutzer- und Systemanforderungen in einem Anforderungsdokument.
- Anforderungen müssen für Endbenutzer und Kunden, die keinen technischen Hintergrund haben, verständlich sein.
- Die Anforderungen können Teil eines Vertrages für die Systementwicklung sein. Es ist daher wichtig, dass sie so vollständig wie möglich sind.

Anforderungsspezifikation

Ansätze

- **Natürliche Sprache** - nummerierte Sätze in natürlicher Sprache. Jeder Satz sollte eine Anforderung ausdrücken.
- **Strukturierte Natürliche Sprache** - natürliche Sprache in einem Standardformular oder einer Vorlage. Jedes Feld enthält Informationen über einen Aspekt der Anforderung.
- **Design Description Language** – eigene Sprache zur Anforderungsdefinition (selten genutzt).
- **Grafische Notationen** – grafische Modelle mit erklärendem Text zur Anforderungsspezifikation.
- **Mathematische Spezifikation** – formelle mathematische Spezifikation. Sehr präzise Formulierungen möglich. Nicht immer möglich (Verständnis, Problemsituation, ...).

Anforderungsspezifikation – Natürliche Sprache

Detaillierte Anforderungsspezifikation zu N3

Nr.	Inhalt	Nutzeranforderung N3
1. Ziel der Anforderung	<p>Das System soll sicherstellen, dass Klinikmitarbeiter auch bei einem Ausfall der Netzwerkverbindung Zugriff auf alle für die Patientenbehandlung notwendigen Informationen haben. Dadurch wird gewährleistet, dass Behandlungen ohne Verzögerung fortgeführt werden können.</p>	Als Klinikmitarbeiter möchte ich Patientenakten auch ohne bestehende Netzwerkverbindung einsehen können, damit die Behandlung auch bei Verbindungsproblemen fortgesetzt werden kann.
2. Beschreibung der Funktionalität	<p>2.1 Das System speichert lokal eine verschlüsselte Kopie aller für die Behandlung relevanten Patientendaten.</p> <p>2.2 Bei Verlust der Netzwerkverbindung schaltet das System automatisch in den Offline-Modus, ohne dass der Nutzer zusätzliche Schritte ausführen muss.</p> <p>2.3 Im Offline-Modus sind verfügbar:</p> <ul style="list-style-type: none">– Anzeige aller gespeicherten Patientendaten (medizinische Historie, Behandlungspläne, Medikationslisten).– Suche und Filterung innerhalb der lokal verfügbaren Daten. <p>2.4 Nach Wiederherstellung der Netzwerkverbindung synchronisiert das System automatisch alle zwischenzeitlich lokal erfassten Änderungen zurück an die zentrale Datenbank.</p>	
3. Nicht-funktionale Anforderungen im Kontext	<ul style="list-style-type: none">• Datenschutz: Lokale Datenspeicherung muss verschlüsselt erfolgen (z. B. AES-256).• Performance: Umschaltung zwischen Online- und Offline-Modus darf maximal 5 Sekunden dauern.• Verfügbarkeit: Lokal gespeicherte Daten müssen jederzeit abrufbar sein, auch bei unerwarteten Unterbrechungen.	

Anforderungsspezifikation

Strukturierte Natürliche Sprache

Insulin Pump/Control Software/SRS/3.3.2

Function Compute insulin dose: safe sugar level.

Description

Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.

Inputs Current sugar reading (r2); the previous two readings (r0 and r1).

Source Current sugar reading from sensor. Other readings from memory.

Outputs CompDose—the dose in insulin to be delivered.

Destination Main control loop.

Action

CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.

Requirements

Two previous readings so that the rate of change of sugar level can be computed.

Pre-condition

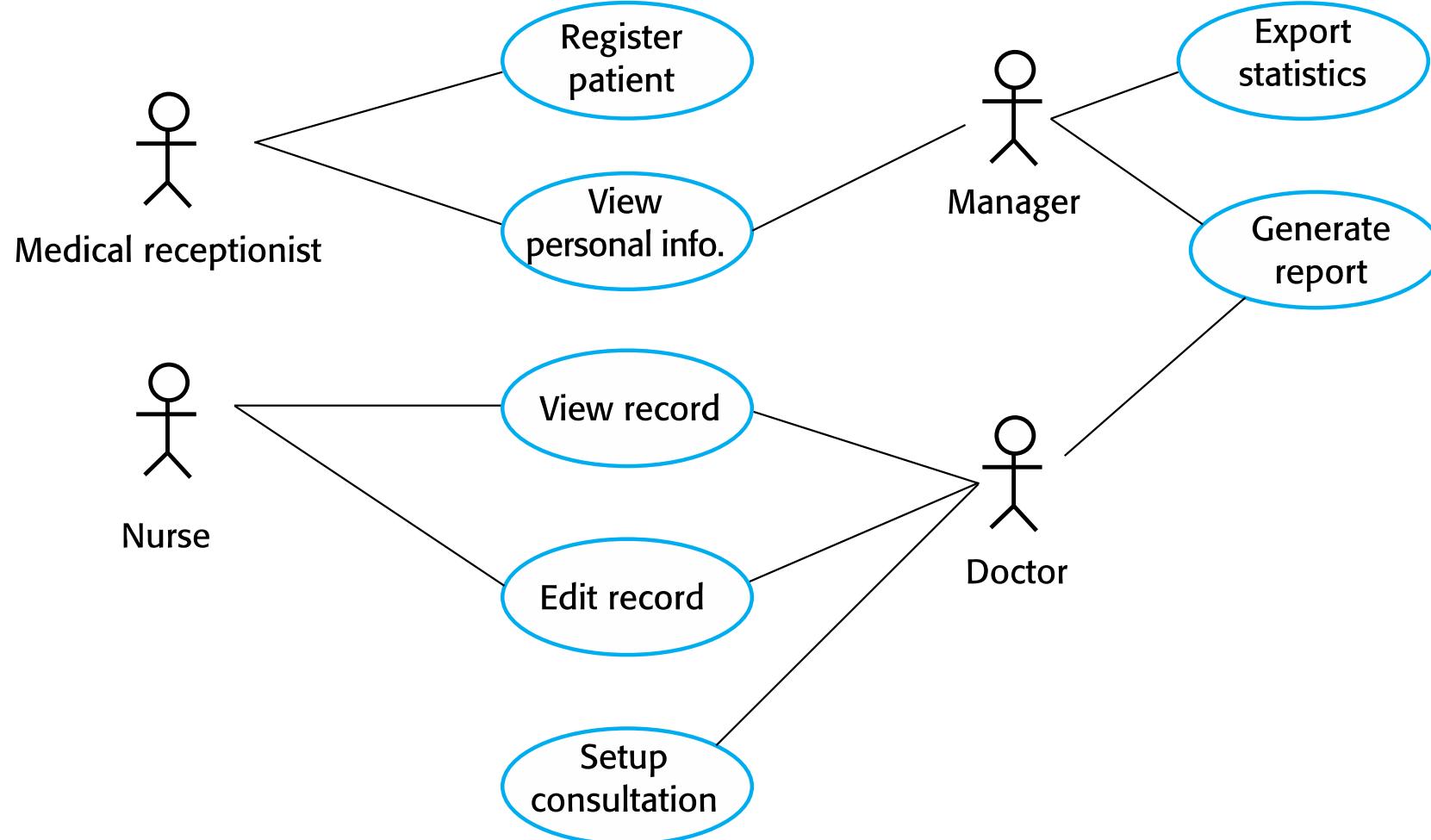
The insulin reservoir contains at least the maximum allowed single dose of insulin.

Post-condition r0 is replaced by r1 then r1 is replaced by r2.

Side effects None.

Anforderungsspezifikation

Grafische Notationen



Guidelines zur Erfassung von Anforderungen

- Nutzen Sie ein Standardformat und verwenden Sie es für alle Anforderungen.
 - Verwenden Sie eine einheitliche Sprache.
-
- Verwenden Sie „muss/soll“ für obligatorische Anforderungen und „sollte/könnte“ für wünschenswerte Anforderungen.
-
- Verwenden Sie Texthervorhebungen um wichtige Teile der Anforderung zu kennzeichnen.
 - Vermeiden Sie die Verwendung von Computerjargon und Umgangssprache.

Erfassung von Anforderungen

Wie machen wir das nun in der Praxis?

Agile Entwicklung

Die Persona

- Um das Produkt kundenspezifisch „richtig“ zu entwickeln, muss ein gutes Verständnis über den Endnutzer hergestellt werden.
- Personas sind "imaginäre Benutzer", bei denen Sie ein Charakterporträt eines Benutzertyps ihres Produkts erstellen.
- Es ist darauf zu achten konkrete Personen und keine Rollen zu verwenden.

Agile Entwicklung

Die Persona

Merkmal	Beschreibung
Name	Dr. Anna Schneider
Rolle / Position	Fachärztin für Psychiatrie und Psychotherapie
Alter	42 Jahre
Ziele	<ul style="list-style-type: none">- Patienteninformationen schnell erfassen und aktualisieren- Medikamente und Behandlungspläne dokumentieren- Termine für ambulante Konsultationen effizient organisieren
Herausforderungen	<ul style="list-style-type: none">- Hoher Zeitdruck durch viele Patienten- Jederzeit aktuelle Informationen benötigen- Datenschutz bei sensiblen Patientendaten sicherstellen
Erwartungen	<ul style="list-style-type: none">- Intuitive und leicht bedienbare Oberfläche- Schneller Zugriff auf Patientenakten- Sichere und DSGVO-konforme Datenübertragung

Agile Entwicklung

Die Persona

Merkmal	Beschreibung
Name	Thomas Berger
Rolle / Position	Pflegefachkraft in der psychiatrischen Station
Alter	35 Jahre
Ziele	<ul style="list-style-type: none">- Rasch auf aktuelle Patientendaten (Medikamente, Anordnungen) zugreifen- Änderungen bei Vitalwerten oder Medikation direkt dokumentieren- Übergaben zwischen Schichten effizient gestalten
Herausforderungen	<ul style="list-style-type: none">- Häufig wechselnde Patientenzustände erfordern schnelle Dokumentation- Schichtarbeit mit begrenzter Zeit für ausführliche Einträge- Technische Systeme müssen auch unter Zeitdruck stabil laufen
Erwartungen	<ul style="list-style-type: none">- Einfache, übersichtliche Eingabemasken für schnelle Dokumentation- Offline-Zugriff auf Patientendaten bei Netzproblemen- Automatische Synchronisation nach Wiederherstellung der Verbindung

Lab

- Erstellung der Persona
 - Definiere drei fiktive Benutzerpersona für das Mentcare System. Gib ihr einen Namen, Alter, Beruf, Hobbies, Fitnessziele und mögliche gesundheitliche Herausforderungen.
 - Beschreibe ihre technische Kompetenz und ihr Nutzungsverhalten im Hinblick auf die Anwendung.

Agile Entwicklung

Szenarien

- Personas werden dazu genutzt, um Szenarien abzuleiten.
- Ein Szenario ist eine Erzählung, die beschreibt, wie ein Benutzer oder eine Gruppe von Benutzern das System verwendet.
- Es ist eine **Beschreibung einer Situation**, in der ein **Benutzer die Funktionen des Produkts nutzt**.

Agile Entwicklung

Szenarien

Aspekt	Beschreibung
Ausgangslage	Es ist Montagmorgen. Dr. Anna Schneider hat einen vollen Terminplan mit ambulanten Konsultationen.
Auslöser	Ein neuer Patient wird in der psychiatrischen Ambulanz aufgenommen. Dr. Schneider muss vor der ersten Konsultation alle relevanten Daten im Mentcare-System anlegen.
Ablauf	<ol style="list-style-type: none">1. Dr. Schneider meldet sich im Mentcare-System an.2. Sie öffnet den Bereich „Patient registrieren“ und gibt die Stammdaten (Name, Geburtsdatum, Kontaktdaten) ein.3. Sie ergänzt bekannte Vorerkrankungen und aktuelle Medikationen.4. Mentcare validiert die Daten und legt automatisch eine neue Patienten-ID an.5. Dr. Schneider plant direkt im System einen Ersttermin für die Diagnostik und erstellt den ersten Behandlungsplan.

Lab

- Entwicklung von Szenarien
 - Erstelle zwei Szenarien zu den zuvor angegebenen Personas. Die Szenarien sollen zeigen, wie die Persona die Anwendung in ihrem Alltag verwenden würde. Die Szenarien sollten unterschiedliche Aspekte ihres Alltags abdecken.
 - Beschreibe die Herausforderungen oder Bedürfnisse, die in jedem Szenario adressiert werden müssen.

Agile Entwicklung

User Stories

- User-Stories werden häufig von Szenarien abgeleitet.
 - User-Stories detaillieren Szenarien in kleine logisch zusammenhängende Teile.
-
- Kunde oder Benutzer ist Teil des Teams und ist für die Erstellung verantwortlich.
 - Die **Nutzeranforderungen** werden in Form von User-Stories durch den Kunden formuliert.
 - Diese werden vom **Entwicklungsteam** in Implementierungsaufgaben (**Tasks**) aufgeteilt.
-
- Die erstellten Stories und deren Tasks bilden die Grundlage für Zeit- und Kostenschätzungen.
 - Der Kunde priorisiert basierend auf den Stories und definiert somit den Umfang des nächsten Releases.

Agile Entwicklung

User Stories (2)

- Zu einer User-Story gehören
 1. Ein Einleitungssatz mit Zieldefinition
 2. Aussagekräftige Beschreibung der Story mit Eigenschaften
 3. Definition of Done (wann ist die Story als erfüllt anzusehen)

1. Einleitung mit Zieldefinition

- Eine User-Story wird stets im folgenden Format formuliert:
 - **As a** <role>, I <want | need> **to** <do something>
 - (Bsp. As a teacher, I want to tell all members of my group when new information is available)
- Als Alternative kann auch eine Begründung eingefügt werden:
 - **As a** <role> I <want | need> **to** <do something> **so that** <reason>
 - (Bsp. As a teacher, I need to be able to report who is attending a class trip so that the school maintains the required health and safety records.)
- In deutsch jeweils "Als [Rolle] [möchte | benötige] ich <Grund | Irgendetwas>, [damit]."

Agile Entwicklung

User Stories (3)

- Zu einer User-Story gehören
 1. Ein Einleitungssatz mit Zieldefinition
 2. Aussagekräftige Beschreibung der Story mit Eigenschaften
 3. Definition of Done (wann ist die Story als erfüllt anzusehen)
- 2. Häufig wird Freitext verwendet
- 3. Definition of Done
 - Gibt an, wann eine User-Story als erledigt anzusehen ist.
 - Die Story darf erst dann in den Zustand „abgeschlossen“ versetzt werden, wenn die Definition of Done nachweislich erfüllt wurde.
 - Dazu werden typischerweise Tests erstellt, die die Definition of Done automatisiert prüfen.

Agile Entwicklung

User Stories

User Story 1: Patientendaten erfassen

Aspekt	Beschreibung
Einleitungssatz (Zieldefinition)	Als Fachärztin für Psychiatrie möchte ich neue Patienten im Mentcare-System anlegen, damit ich alle relevanten Daten für Diagnose und Behandlung sofort verfügbar habe.
Beschreibung	<p>Ausschnitt!</p> <ul style="list-style-type: none">- Dr. Anna Schneider kann im Modul „Patient registrieren“ alle notwendigen Stammdaten eingeben (Name, Geburtsdatum, Kontaktdaten).- Vorerkrankungen und aktuelle Medikationen können direkt ergänzt werden.- Das System validiert Eingaben auf Vollständigkeit und Plausibilität.- Bei fehlerhaften Eingaben werden klare Fehlermeldungen angezeigt.
Definition of Done	<ul style="list-style-type: none">- Der Patient ist mit einer eindeutigen Patienten-ID im System erfasst.- Alle relevanten Stammdaten sowie Vorerkrankungen/Medikationen sind gespeichert.- Bei Testläufen sind keine Validierungsfehler vorhanden.

Agile Entwicklung

User Stories

User Story 2: Termin für Erstdiagnose planen

Aspekt	Beschreibung
Einleitungssatz (Zieldefinition)	Als Fachärztin für Psychiatrie möchte ich direkt nach der Registrierung einen Ersttermin anlegen, damit Patient und Team schnell über den geplanten Diagnosetermin informiert sind.
Beschreibung	<p>Ausschnitt!</p> <ul style="list-style-type: none">- Nach erfolgreicher Registrierung öffnet sich automatisch das Modul „Terminplanung“.- Dr. Anna Schneider kann Datum, Uhrzeit und behandelnden Arzt auswählen.- Das System prüft auf Terminkonflikte und zeigt freie Zeitfenster an.- Der Termin wird automatisch im Patientenprofil hinterlegt und eine Bestätigung wird an die Verwaltung gesendet.
Definition of Done	<ul style="list-style-type: none">- Ein Ersttermin ist im System erstellt und im Patientenprofil sichtbar.- Terminbestätigung wurde automatisch generiert und an Verwaltung/Patient weitergegeben.- Bei Testläufen sind keine Terminüberschneidungen aufgetreten.

Agile Entwicklung

Task

- Tasks werden häufig aus User-Stories abgeleitet.
 - Tasks detaillieren User-Stories in kleine implementierbare Einheiten.
 - Manchmal können User-Stories auch direkt als Tasks interpretiert werden.
-
- Ein Task sollte **ein bis drei Tage Realisierungsaufwand** nicht übersteigen.

Agile Entwicklung

Task

Task 1: Frontend – Eingabemaske „Patient registrieren“

Bezug zu User Story 1: Erlaubt Dr. Schneider, alle relevanten Stammdaten, Vorerkrankungen und Medikationen einzugeben.

UI/UX-Anforderungen

- Seite/Modal: `/patients/new` mit primärer Aktion „Speichern“ und Sekundäraktion „Abbrechen“.
- Formular-Gruppen: Stammdaten, Kontakt, Medizinisch, Einverständnis/DSGVO.
- Live-Validierung pro Feld (onBlur) + Gesamtsumme beim Submit (scroll to first error).
- Erfolgs-Toast („Patient angelegt“) und Redirect auf `/patients/{patientId}`.

Felder & Typen (Client)

- Stammdaten:
 - Vorname (String, Pflicht, 1–50), Nachname (String, Pflicht, 1–50)
 - Geburtsdatum (Date, Pflicht, ISO-8601, darf nicht in der Zukunft liegen)
 - Geschlecht (Enum: {„weiblich“, „männlich“, „divers“, „unbekannt“}, optional)
- Kontakt:
 - Adresse (String, optional, max. 200), Telefonnummer (String, optional, E.164 empfohlen), Email (String, optional, RFC 5322)
- Medizinisch:
 - Vorerkrankungen (Mehrfachauswahl/Tags, optional)
 - AktuelleMedikation (Tabelle: Name:String, Dosis:String, Häufigkeit:String; 0..n)

- Rechtliches:
 - EinwilligungDatenerarbeitung (Checkbox, Pflicht)
 - Notfallkontakt (optional: Name, Beziehung, Telefon)

Validierungsregeln (Client)

- Pflichtfelder: Vorname, Nachname, Geburtsdatum, Einwilligung.
- Geburtsdatum \leq heute.
- Email gültig, Telefonnummer nur Ziffern/„+“, max. 20.

Akzeptanzkriterien

- Bei gültiger Eingabe wird POST `/api/patients` abgesetzt; nach 201 Redirect auf Detailseite.
- Bei Validierungsfehlern: Fehler-Tooltip am Feld + Zusammenfassung oben.
- Formular ist per Tastatur bedienbar (A11y: Labels, ARIA, Fokusmanagement).

Lab

- Formulierung von User-Stories
 - Wählen Sie eins der vorherigen Szenarios aus und entwickeln Sie dazu zwei User-Stories/Anwendungsfälle.
 - Jede Story sollte folgendes Format haben: „Als [Rolle] möchte ich [Ziel], um [Nutzen].“
 - Stelle sicher, dass jede Story ein spezifisches Bedürfnis oder Ziel der Persona reflektiert und wie die Anwendung dabei helfen kann.
- Ergänzen Sie zu jeder User Story:
 - Beschreibung:** kurze Erläuterung der Funktion oder des Ziels.
 - Akzeptanzkriterien:** wann gilt die Story als erfüllt?
 - Definition of Done:** wie kann die Erfüllung überprüft werden (z. B. Testfall, Nutzerfeedback)?

Erfassung von Anforderungen

- Wir haben nun “alle” Anforderungen systematisch erfasst.
- Wenn aber die Anforderungen zu Beginn nicht detailliert erfasst werden können, wie können wir sicherstellen, dass der Kunde bekommt was er bestellt?

Erfassung von Anforderungen

- Wir haben nun “alle” Anforderungen systematisch erfasst.
- Wenn aber die Anforderungen zu Beginn nicht detailliert erfasst werden können, wie können wir sicherstellen, dass der Kunde bekommt was er bestellt?
- →Anforderungsvalidierung

Validierung von Anforderungen

- Sicherstellung, dass der Kunde tatsächlich bekommt, was er bestellt (bezahlt)
- „Aufgeschriebene“ Anforderungen können vom Kundenwunsch abweichen
 - lückenhafte Anforderungserfassung
 - falsches Domänenverständnis
 - falsche Interviewpartner
 - falsche Beobachtung der Benutzer
 - falsche technische Rahmenbedingungen
 - Kunde weiß eigentlich gar nicht was er genau will
 -

Validierung von Anforderungen

- Zunächst ist eine erste Prüfung mit aktuellem Wissen sinnvoll (ohne Kunden).
- **Gültigkeit** - bietet das System die Funktionen, die den Bedürfnissen des Kunden bestmöglich entsprechen?
- **Konsistenz** - gibt es Anforderungskonflikte?
- **Vollständig** - sind alle vom Kunden benötigten Funktionen enthalten?
- **Realitätsnähe** - können die Anforderungen mit dem verfügbaren Budget und der verfügbaren Technologie umgesetzt werden?
- **Überprüfbarkeit** - können die Anforderungen von allen überprüft werden?

Validierung von Anforderungen

Schön und gut, aber wie machen wir das in der Praxis und wie kann das ohne zu viel „Papierkram“ erledigt werden?

Überprüfung und Machbarkeit von Anforderungen

- Systematische **manuelle Analyse** der Anforderungen.
 - **Proof-of-Concept** - Aufklärung von technischen Unsicherheiten (z.B. Integrationen).
 - **Prototyping** - Abstimmung und Prüfung wichtiger Details.
 - **Entwicklung von Tests** – kontinuierliche automatisierte Validierung der Anforderungen.
 - **Regelmäßige Reviews** mit Kunden minimieren Unsicherheit.
-
- Gute Kommunikation zwischen Kunden und Projektteam (auch Developer).

Validierung von Anforderungen

Wie können wir Missverständnisse bei der
Anforderungsermittlung und Validierung möglichst vermeiden?

Validierung von Anforderungen

Wie können wir Missverständnisse bei der
Anforderungsermittlung und Validierung möglichst vermeiden?

→ Unterstützung von grafischen Tools zur Modellierung.

Systemmodellierung

- Anders als früher, wird eine vollständige Modellierung der Anforderungen eines Systems durch eine grafische Notation nicht mehr praktiziert
- Da viele Menschen visuell getrieben sind, helfen grafische Darstellungen dennoch beim Verständnis von Anforderungen
- Komplexe Details oder Abhängigkeiten können visuell häufig gut veranschaulicht werden.
- Systemumgebungen lassen sich durch grafisch basierte Standards gut abbilden.
- Gleiches gilt für das Beschreiben eines Ablaufs oder Prozesses.

Systemmodellierung

- Systemmodellierung: Prozess der Entwicklung abstrakter Modelle eines Systems.
- Es existieren verschiedene Modelle, die jeweils **einzelne Perspektiven** des Systems abbilden.
- Um die Modelle darzustellen, wird eine Modellierungssprache benötigt.
 - Es sollte ein Standard genutzt werden, den **möglichst viele kennen!**
 - Bsp: UML, Archimate, Togaf, LeanFx, ...

Systemmodellierung

System Perspektiven

- **Externe Perspektive** – Modellierung des Kontext und der Umgebung des Systems.
- **Interaktionsperspektive** – Modellierung der Interaktionen zwischen einem System und seiner Umgebung oder zwischen den Komponenten eines Systems.
- **Strukturelle Perspektive** – Modellierung der Strukturierung eines Systems.
- **Verhaltensperspektive** – Modellierung des dynamischen Verhaltens des Systems und dessen Reaktion auf Ereignisse.

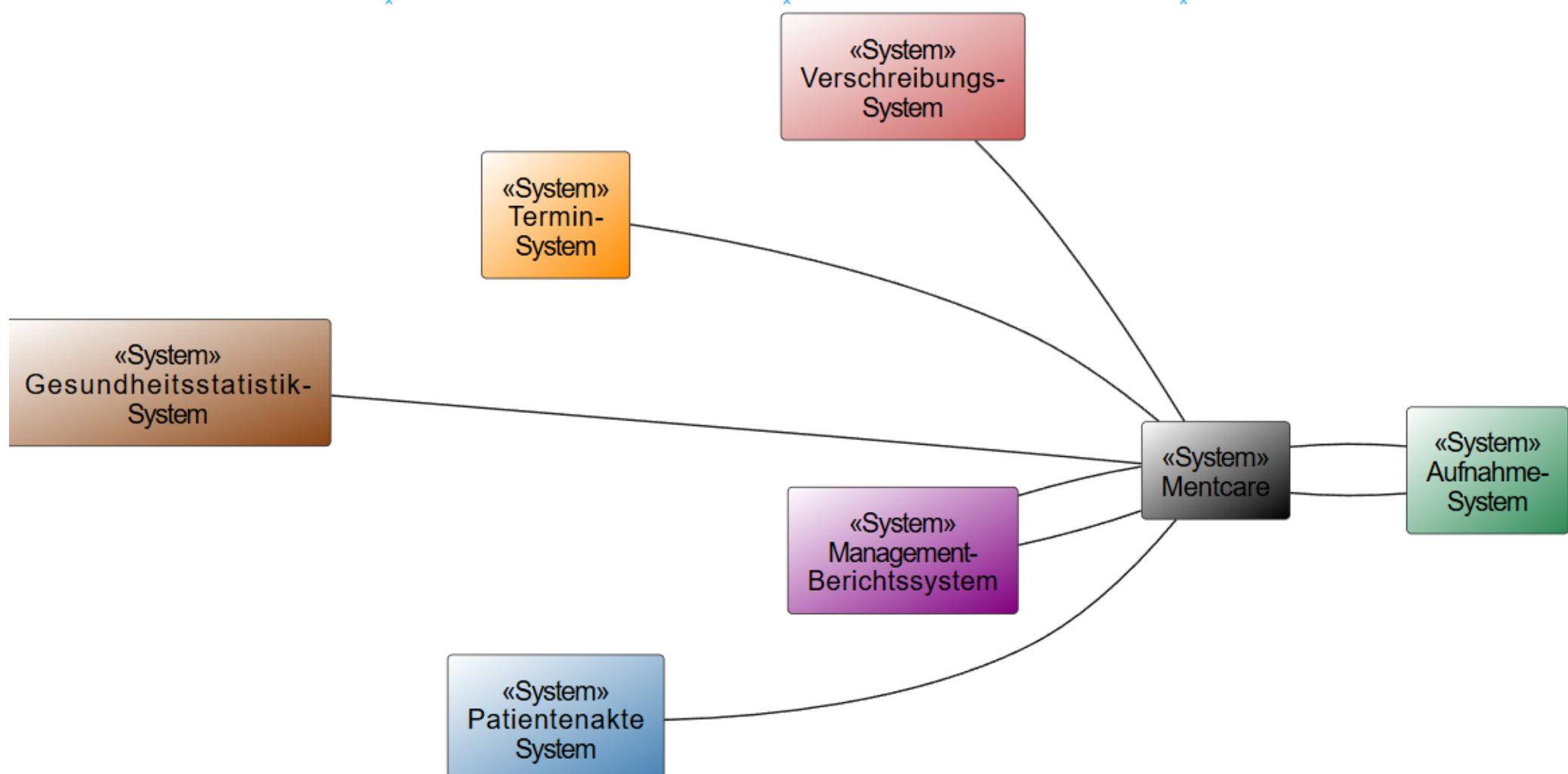
Kontextmodelle

Systemgrenzen

- Werden festgelegt um zu definieren, was sich innerhalb und außerhalb des Systems befindet.
- Die Position der Systemgrenze hat einen tiefgreifenden Einfluss auf die Systemanforderungen.
- Das Kontextmodell wird häufig durch ein **Komponentendiagramm** umgesetzt.

Kontextmodelle

Systemgrenzen im Mentcare System



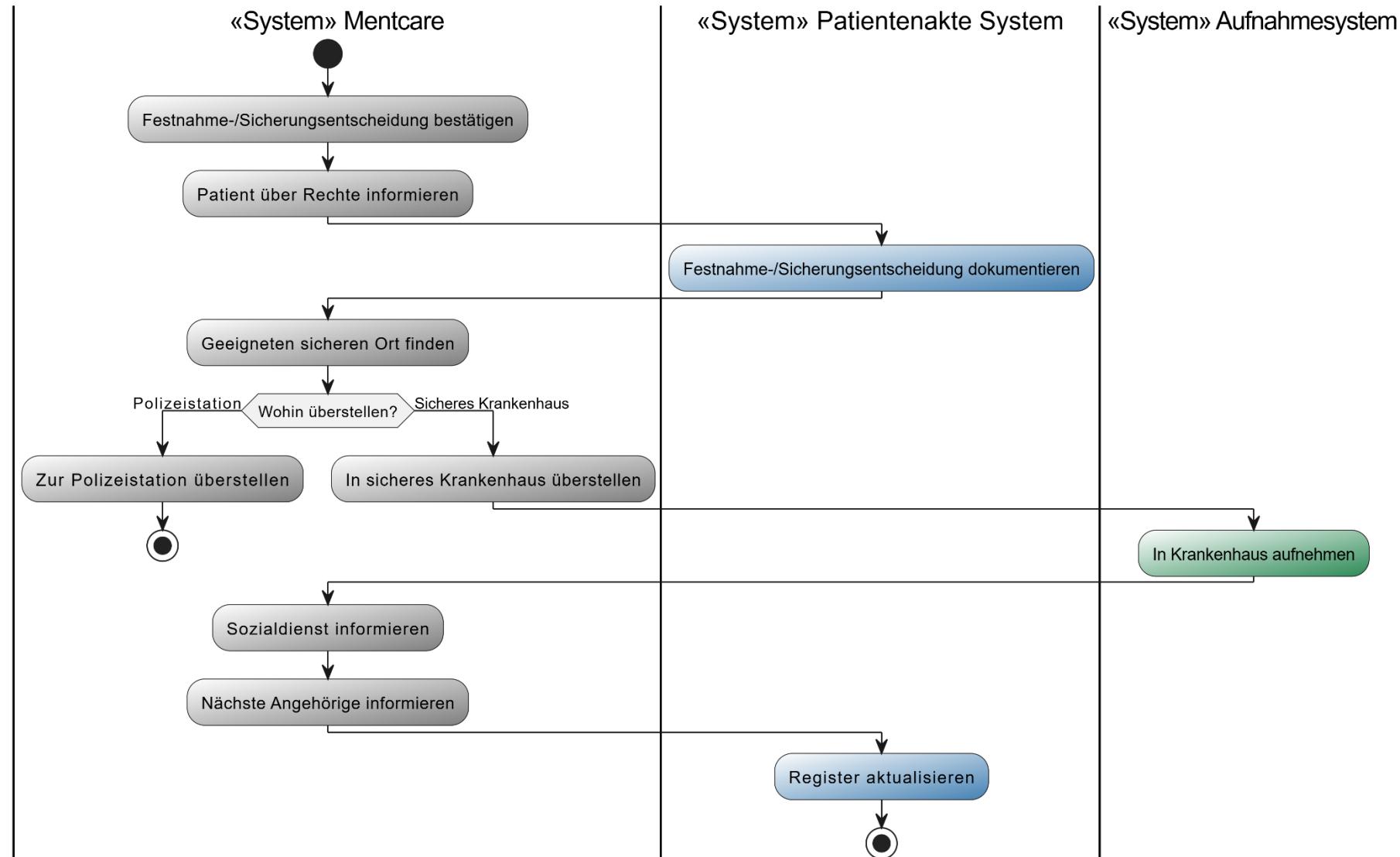
Kontextmodelle

Prozess Perspektive

- Kontextmodelle zeigen die anderen Systeme in der Umgebung.
- Sie zeigen nicht, wie das zu entwickelnde System in dieser Umgebung eingesetzt wird.
- Prozessmodelle definieren, wie das zu entwickelnde System in umfassenden Geschäftsprozessen eingebunden wird.
- **Aktivitätsdiagramme** können zur Definition von Geschäftsprozessmodellen verwendet werden.

Kontextmodelle

Prozess Perspektive – Gefährlichen Patienten in Gewahrsam nehmen



Interaktionsmodelle

- Modellierung der Benutzerinteraktion.
- Modellierung der System-zu-System-Interaktion kann Kommunikationsprobleme aufzeigen.
- Interaktionsmodellierung hilft, die nötige Kommunikationsstruktur und Architektur festzulegen und zu überprüfen (häufige Kommunikation, datenintensiv, ...).
- Für die Modellierung der Interaktion können **Anwendungsfalldiagramme** und **Sequenzdiagramme** verwendet werden.

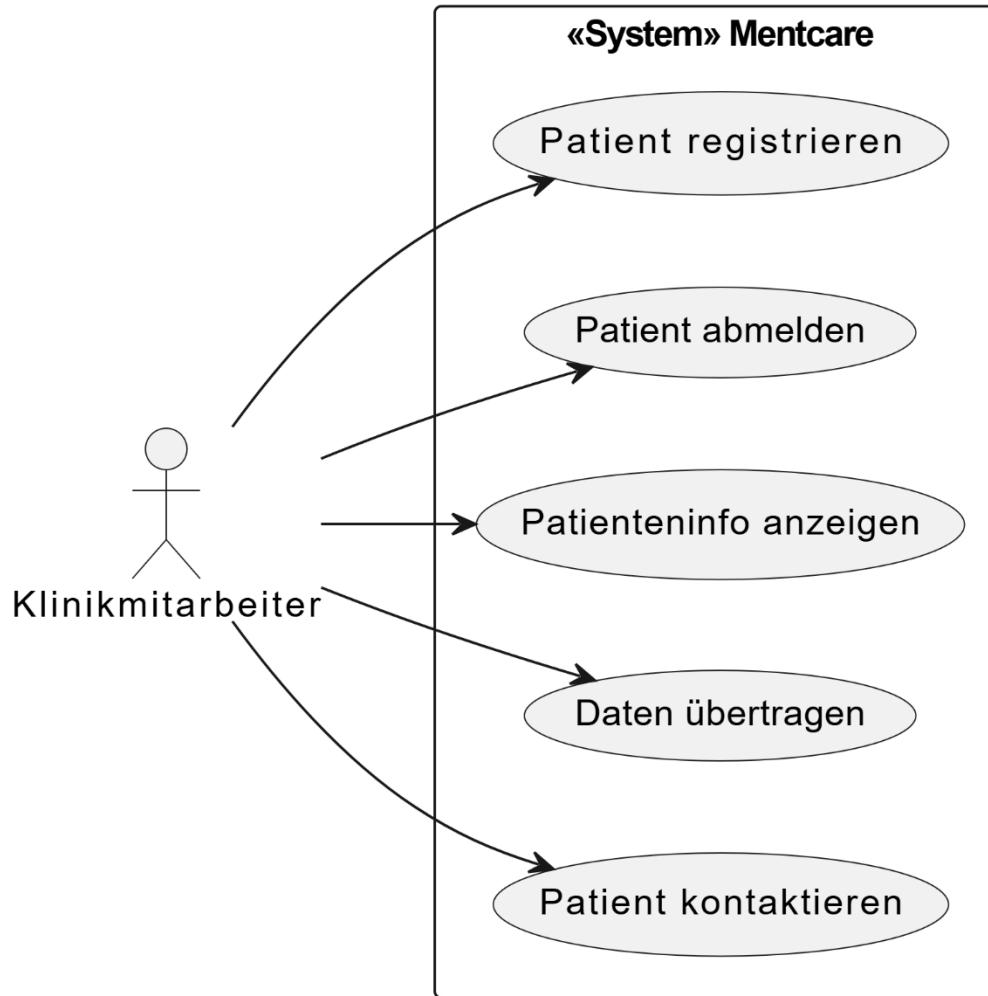
Interaktionsmodelle

Anwendungsfalldiagramme

- Anwendungsfälle dienen zur Unterstützung der Anforderungserhebung.
 - Jeder Anwendungsfall stellt eine diskrete Aufgabe dar, die eine externe Interaktion mit einem System beinhaltet.
 - Die **Akteure** in einem Anwendungsfall können **Menschen** oder **andere Systeme** sein.
-
- Die Darstellung erfolgt in Diagrammen.
 - Eine textuelle Beschreibung ergänzt wichtige Details.

Interaktionsmodelle

Anwendungsfalldiagramme



Lab

- Definieren Sie neben dem Klinikmitarbeiter einen weiteren Akteur und modellieren Sie die dazugehörigen Anwendungsfälle mit Hilfe eines Anwendungsfalldiagramms.
- Denken Sie dabei auch an die Abhängigkeiten zwischen den Anwendungsfällen.

Lab

- Nutzen und prüfen Sie das bestehende Kontextmodell und passen dieses ggf. an.

Interaktionsmodelle

Anwendungsfalldiagramme – Details „Patient registrieren“

- Nach der Modellierung muss jeder Anwendungsfall detailliert beschrieben werden.

Nr.	Inhalt (gekürzt)
1. Ziel	Neuer Patient wird im Mentcare-System erfasst, damit Behandlungs- und Verwaltungsprozesse möglich sind.
2. Akteure	Hauptakteur: Klinikmitarbeiter ... Nebenakteur: Patient ...
3. Auslöser	Patient kommt neu in die Klinik und muss im System angelegt werden.
4. Vorbedingungen	Mitarbeiter ist angemeldet ... alle relevanten Patientendaten liegen vor.
5. Ablauf	1. Mitarbeiter wählt „Patient registrieren“. 2. System zeigt Eingabemaske ... 3. Daten erfassen ... 4. System validiert ... 5. Daten werden gespeichert ... 6. System erstellt Patienten-ID.
6. Ergebnis	Patient ist mit eindeutiger ID registriert ... Daten stehen sofort zur Verfügung.
7. Alternativen	Ungültige Daten → Fehlermeldung ... Keine Verbindung → Eingaben lokal speichern ...
8. Nicht-funktionale Anforderungen	Datenschutz gemäß DSGVO ... Registrierung max. 3 Minuten.
9. Abnahmekriterien	Test zeigt, dass Patient fehlerfrei angelegt wird ... ID sofort verfügbar.

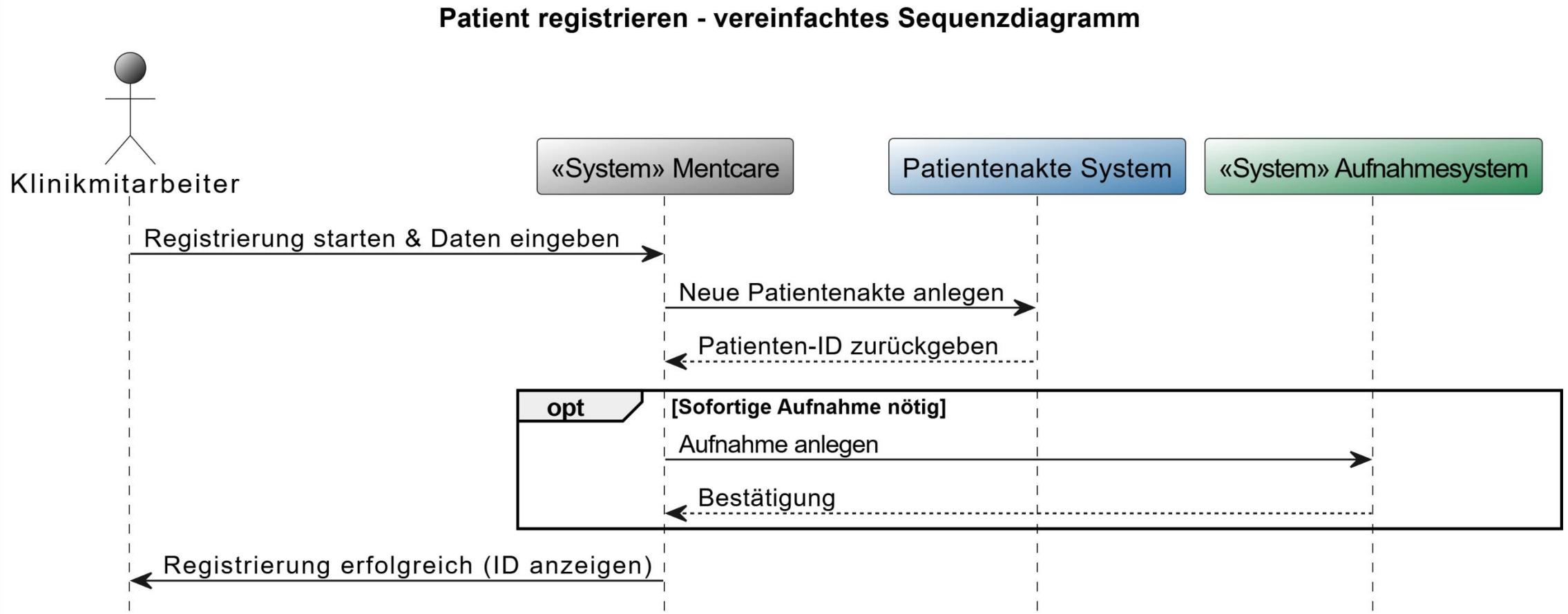
Interaktionsmodelle

Sequenzdiagramme

- Modellierung der Interaktionen zwischen den Akteuren und den Komponenten innerhalb eines Systems.
- **Aufzeigen der Abfolge von Interaktionen, die während eines Anwendungsfalls stattfinden.**
- Die beteiligten Komponenten und Akteure werden oben im Diagramm aufgelistet, wobei eine gestrichelte Linie vertikal von diesen ausgeht.
- Die Interaktionen zwischen den Komponenten werden durch kommentierte Pfeile angezeigt.

Interaktionsmodelle

Sequenzdiagramme – zunächst ohne Datenübergabe



Lab

- Erstellen Sie mindestens ein Sequenzdiagramm, welches eins ihrer zuvor definierten Szenarien abbildet.

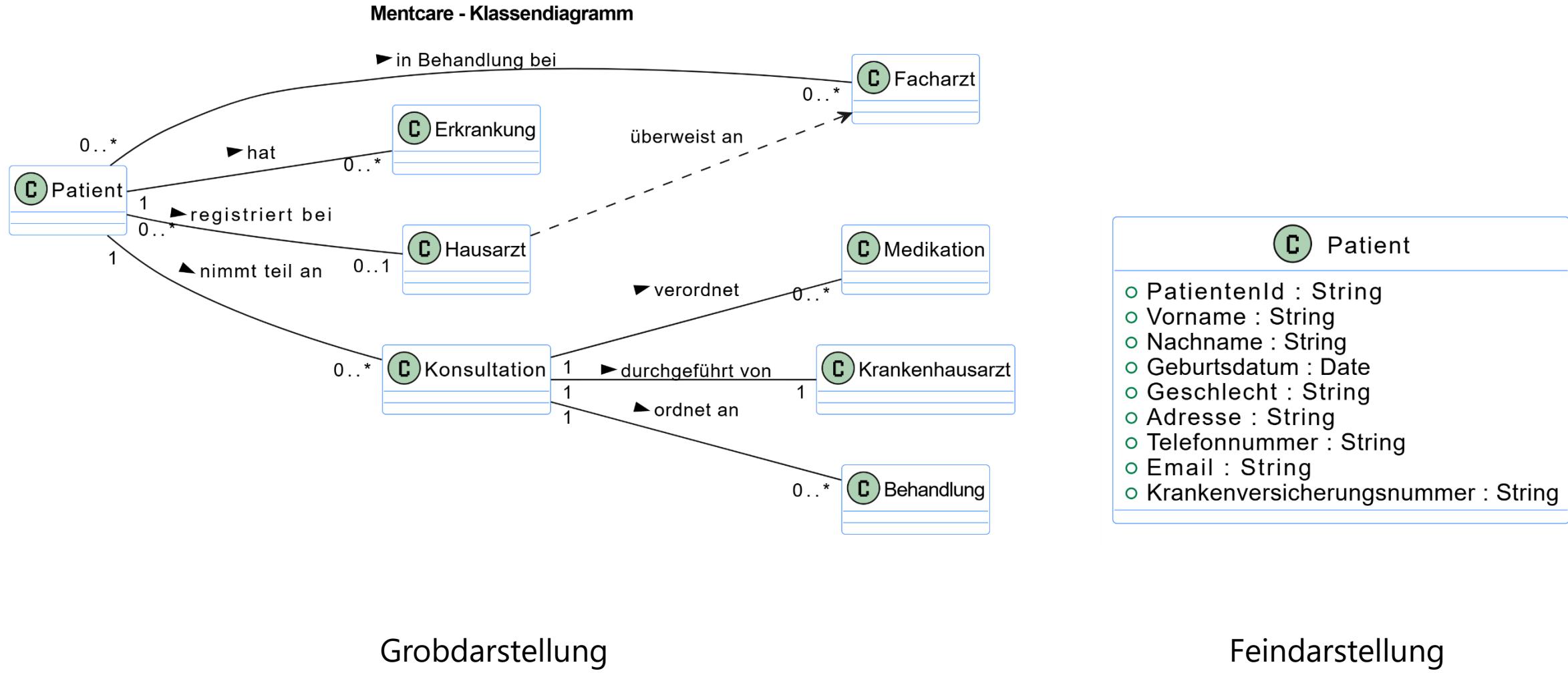
Strukturelle Modelle

- Zeigen die Organisation eines Systems in Form von Komponenten und Services.
- Es werden die Beziehungen der Komponenten/Services angegeben.
- **Statische Modelle** – feste Struktur des Systementwurfs (**Klassendiagramme**)
- **Dynamische Modelle** - Organisation des Systems während der Ausführung
(Verhaltensdiagramm, Zustandsdiagramm)
- Strukturelle Modelle werden erstellt, um die Architektur innerhalb einer Komponente zu entwickeln/diskutieren (Schichtenmodell, Klassenhierarchien, Datenstrukturen, ...).

Strukturelle Modelle - Klassendiagramme

- Helfen bei der Entwicklung eines objektorientierten Systems.
- **Modellierung der Klassen** und der **Abhangigkeiten** in einem System.
- Eine **Assoziation** ist eine Verbindung zwischen Klassen, die deren Beziehung angibt.
- Bei der Entwicklung von Modellen in fruhen Phasen stellen Objekte etwas in der **realen Welt** dar (einen Patienten, ein Rezept, einen Arzt, ...).

Strukturelle Modelle - Klassendiagramme

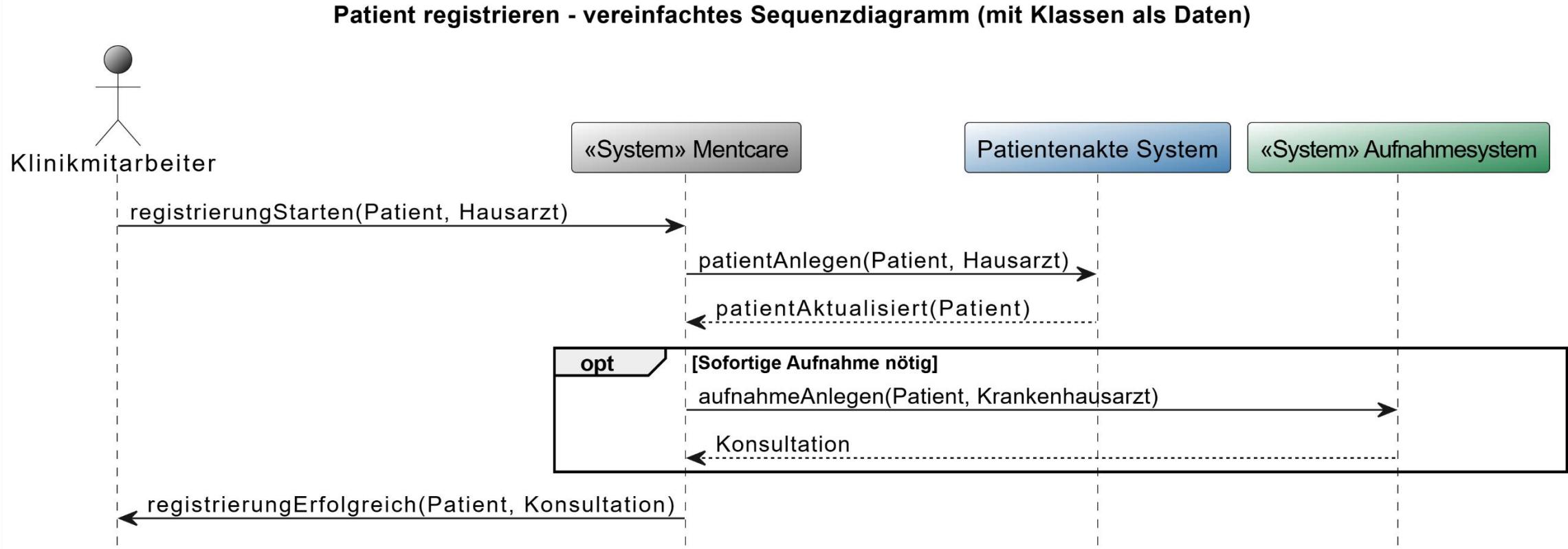


Lab

- Nutzen Sie die Modellierungstechnik eines groben Klassendiagramms, um weitere Systemdaten zu modellieren.
 - Passen Sie dazu das bestehende Diagramm an oder erstellen ein neues.

Interaktionsmodelle

Sequenzdiagramme – finale Darstellung



Lab

- Passen Sie ihr Sequenzdiagramm so an, dass nun die benötigen Datenstrukturen/Klassen an den Funktionen aus dem Sequenzdiagramm hinterlegt sind.
- Falls Datenstrukturen benötigt werden, die aktuell nicht im Klassendiagramm sind, passen sie dieses an.