

Bachelor of Science (BSc) in Informatik
Modul Software-Entwicklung 1 (SWEN1)

LE 03 – Anforderungsanalyse II

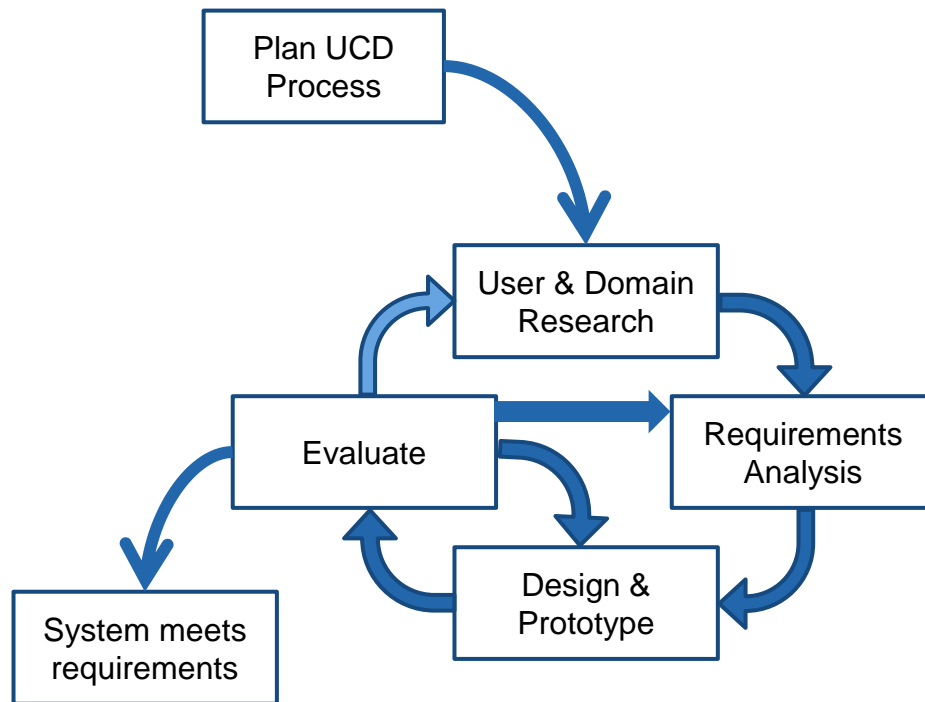
SWEN1/PM3 Team:
R. Ferri (feit), D. Liebhart (lieh), K. Bleisch (bles), G. Wyder (wydg)

Ausgabe: HS24

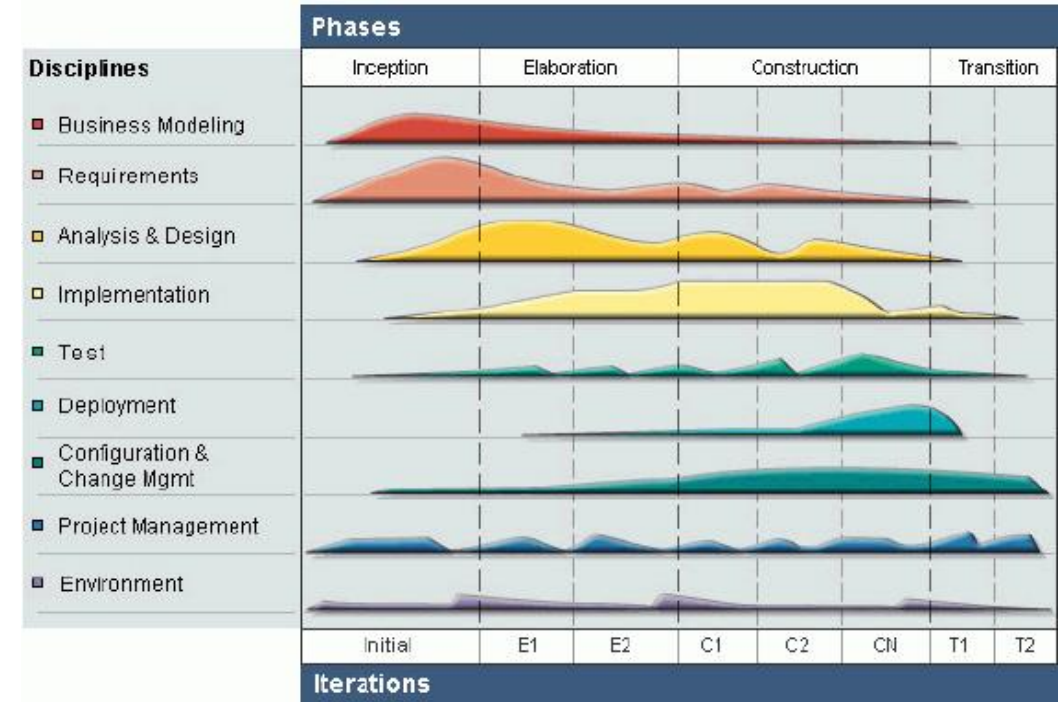
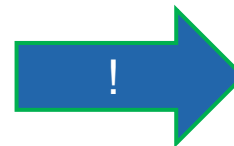
- Wo stehen wir im Software-Entwicklungsprozess nach der Anforderungsanalyse I?
- Was sind die 3 wichtigsten Usability-Anforderungen?
- Welche Artefakte werden typischerweise im UCD erstellt und was beschreiben sie?

Um was geht es?

- Wie bringt man UCD in SWE-Prozesse ein?

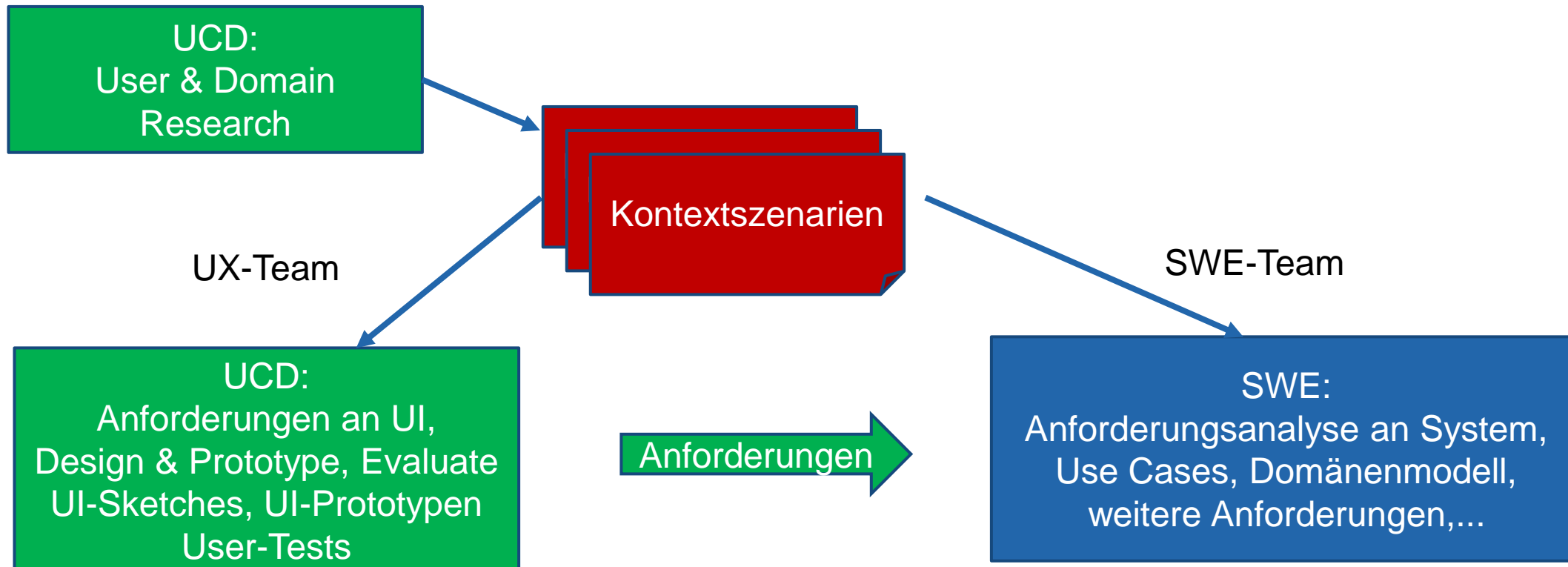


UX-Team



SWE-Team

Anforderungsanalyse: Benutzer-Anforderungen



Lernziele LE 03 – Anforderungsanalyse II

Sie sind in der Lage,

- aus den Artefakten des UCD detaillierte Anforderungen an das Softwareprodukt und dessen UI abzuleiten.
- funktionale Anforderungen in Form von Use Cases in verschiedenen Granularitätsstufen aus Kontextszenarien abzuleiten.
- weitere funktionale und nicht-funktionale Anforderungen aus den Artefakten des UCD abzuleiten.

Agenda

1. Wie erfasst man (dynamische) funktionale Anforderungen mit Use Cases
2. Wie schreibt man gute Use Cases
3. System-Sequenzdiagramme, Systemoperationen und Verträge
4. Wie erfasst man zusätzliche funktionale und nicht-funktionale Anforderungen
5. Wrap-up und Ausblick

SWE: Anforderungsanalyse

- Anforderungen (Requirements)
 - Forderungen bezüglich (Leistungs-) Fähigkeiten oder Eigenschaften, die das System/Projekt unter gegebenen Bedingungen erfüllen muss
 - Können explizit oder implizit sein
- Kurze Diskussion
 - Woher kommen die Anforderungen an ein System und wieso?
 1. Benutzer (möchte einen Job erledigen in einem bestimmten Kontext, hat bestimmte Bedürfnisse, Skills, Ziele)
 2. Weitere Stakeholder (Einkauf, Projektleiter, IT-Abteilung, ...)
 3. Regulatorien, Gesetze, Normen

SWE: Anforderungsanalyse

- Anforderungen (Requirements)
 - Sind (fast) **nie im Vorneherein vollständig bekannt**
 - **Müssen** zusammen mit den Benutzern und anderen Stakeholdern **erarbeitet werden**
 - Sie haben häufig **implizite** Anforderungen nicht explizite
 - Explizite Anforderungen sollten hinterfragt werden (wieso bestehen sie genau so)
 - Können kaum je zu Beginn vollständig erhoben werden, sondern **entwickeln sich im Verlaufe des Projekts**
 - «I don't know what I want but I'll tell you when I see it!»
 - Problematisch bei nicht iterativen Prozessen

Anforderungsanalyse: Anwendungsfälle (Use Cases)

- Textuelle Beschreibung einer konkreten Interaktion eines bestimmten Benutzers mit dem zukünftigen System
 - Aus Sicht des Akteurs
 - Enthalten implizite und explizite Anforderungen
 - Beschreiben das Ziel des Benutzers (= Grund für die Anforderungen)
 - Beschreiben den Kontext

Anforderungsanalyse: Anwendungsfälle (Use Cases)

- **Use Cases (UCs)** bilden in iterativen SWE-Prozessen eine **zentrale Rolle**
 - **Funktionale Anforderungen** werden hauptsächlich mit UCs dokumentiert
 - Mittels UCs können Anforderungen einfach mit dem Kunden diskutiert werden
 - UCs sind ein wichtiger Teil der **iterativen Projektplanung**
 - Projekt wird entlang von UCs geplant
 - UC-Realisierungen **bestimmen die Lösungsarchitektur** und treiben das Lösungsdesign
 - UCs werden für **funktionale Systemtests** eingesetzt
 - UCs bilden die **Basis für Benutzerhandbücher**

Anforderungsanalyse: Anwendungsfälle (Use Cases)

- Akteur (Actor)
Externe Person in einer bestimmten Rolle, die mit dem zu entwickelnden System (SuD) im Laufe eines Anwendungsfalls interagiert
- **Externe Systeme, Organisationen, Maschinen** können auch Akteure sein
- Auch die **Zeit** kann ein Akteur sein
 - Bei zeitabhängigen UCs
- 3 Arten von Akteuren
 - **Primärakteur** (Primary Actor)
 - **Initiiert** einen Anwendungsfall, um sein (Teil-)Ziel zu erreichen
 - Erhält den **Hauptnutzen** des Anwendungsfalls
 - Beispiel Kasse: Kassier
 - **Unterstützender Akteur** (Supporting Actor)
 - **Hilft dem SuD** bei der Bearbeitung eines Anwendungsfalls
 - Beispiel Kasse: externer Dienstleister wie Zahlungsdienst für Kreditkarten
 - **Offstage-Akteur** (Offstage Actor)
 - Weitere Stakeholder, die **nicht direkt** mit dem System interagierten
 - Beispiel Kasse: Steuerbehörde

Anforderungsanalyse: Anwendungsfälle (Use Cases)

- 3 Ausprägungen von Anwendungsfällen
 - Kurz (Brief UC)
 - Titel + 1 Absatz
 - Beschreibt Standardablauf (keine Varianten, Problemfälle)
 - Informell (Casual UC)
 - Titel + informelle Beschreibung in ein bis mehrere Absätze
 - Beschreibt auch wichtige Varianten
 - Vollständig (Fully dressed UC)
 - Titel + alle Schritte und Varianten werden im Detail beschrieben
 - Enthalten weitere Informationen zu Vorbedingungen, Erfolgsgarantien, etc.

Anforderungsanalyse: Anwendungsfälle (Use Cases)

- Umfang eines guten Use Case
 - Muss einen **konkreten Nutzen** für den Akteur erzeugen
 - Eine Handlung, die **eine** Person, an **einem** Ort zu **einer** Zeit mit dem System ausführt
 - Sollte **mehr als eine einzelne Interaktion** umfassen
- Titel eines UC
 - **Aktiv formulieren**
 - Verb! + evtl. Objekt vorangestellt (z.B. „Kasse eröffnen“)
 - Sollte **Ziel des Akteurs** beschreiben
- **Boss-Test**
 - Falls dein Boss dich fragt, was du den ganzen Tag gemacht hast und du sagst, du hast die ganze Zeit den einen Use Case ausgeführt, sollte er zufrieden sein!
- **EBP-Test** (Elementary Business Proc.)
 - Eine Aufgabe, die von einer Person an einem Ort zu einer bestimmten Zeit ausgeführt wird, als Reaktion auf einen Business Event.
- **Size-Test**
 - Mehr als eine einzelne Interaktion
 - Fully dressed meist mehrere Seiten lang

Anforderungsanalyse: Anwendungsfälle (Use Cases)

- Schlechte UC-Namen
 - Initialisierung
 - Einloggen
 - Preis eintippen
 - Einkäufe machen
 - Kasse bedienen
- Gute UC-Namen
 - System initialisieren
 - System aufstarten
 - Artikel erfassen
 - (Einen) Einkauf erfassen

- **Aufgabe 3.1**
 - Studieren Sie das Kontextszenario von Folie 16 und leiten Sie daraus mögliche UCs her
 - Halten Sie fest, wer der Akteur des UCs ist
 - Beschreiben Sie den UC mit einem treffenden Titel
 - Überlegen Sie sich weitere mögliche UCs für den Kassier und andere Akteure
- Wie: in 2er-Gruppen
- Zeit: 10‘

User & Domain Research: Artefakte

Beispiel Kontextszenario

Karin ist eine Kioskverkäuferin. Sie hat heute viele Kunden, aber zum Glück hat sie die neue Kasse «Swift» bekommen, mit der sie Produkte schnell einscannen kann. Kurz nachdem sie die Kasse eröffnet hat, trifft der erste Kunde ein. Er hat sich eine Tasse Kaffee am Automaten herausgelassen und zwei Gipfeli von der Selbstbedienungsauslage genommen: «Ein Kaffee crème und ein, nein zwei Gipfeli, und dann möchte ich noch die neuste NZZ.» Karin nimmt die neuste Ausgabe der NZZ vom Stapel und scannt den Barcode ein. Sofort registriert die Kasse 1x NZZ und zeigt der Preis von 5.60 Fr. an. Dann wählt Karin noch einen Kaffee Crème aus sowie Buttergipfeli und die Anzahl 2. Die Kasse registriert einen Kaffee Crème und zwei Buttergipfeli und listet alle Artikel mit der gewünschten Anzahl, dem Einzelpreis, sowie das Total pro Artikel auf und zeigt das Gesamttotal an. «Macht total 13.20 Fr, bitte» sagt Karin und gibt dem Kunden die NZZ und nimmt die 20 Fr.-Note des Kunden entgegen und tippt den bezahlten Betrag ein. Sofort zeigt die Kasse das auszuzahlende Rückgeld an. «20 Fr., dann erhalten sie hier 6.80 Fr. zurück. Vielen Dank und auf Wiedersehen». Karin gibt dem Kunden das Rückgeld zurück und schliesst gleichzeitig die Geldschublade der Kasse mit der anderen Hand. Dadurch wird der Einkauf abgeschlossen und Karin ist bereit für die nächste Kundin.

Nach Ladenschluss macht Karin die Abrechnung. Die Kasse zeigt ihr die Gesamteinnahmen und wieviel Bargeld in der Kasse vorhanden sein müssten. Schnell zählt Karin das Bargeld und bestätigt, dass sich genau dieser Betrag in der Kasse befindet. Sie entnimmt der Kasse soviel Geld, dass nur wieder der Anfangsbetrag in der Kasse verbleibt. Das entnommene Geld bringt sie gleich zur Bank.

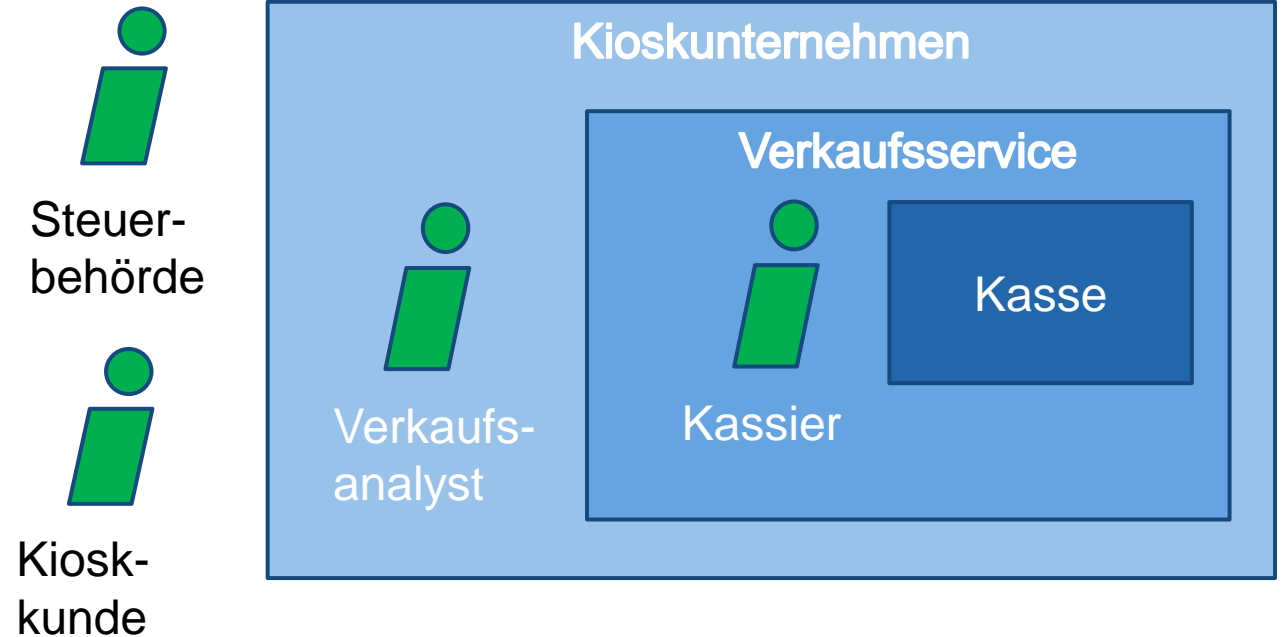
Finden von Anwendungsfällen

- Primärakteur:
 - Aus **Kontextszenarien**
 - **Weitere Ziele/Aufgaben** des Primärakteurs
- Akteur: Kassier
 - Verkauf abwickeln
 - Kasse eröffnen
 - Kasse abrechnen

Finden von Anwendungsfällen

- Schritt 1: Systemgrenzen definieren
 - Kasse

- Schritt 2: Primärakteure identifizieren
 - Kassier
 - Frage: Wieso nicht Kunde?

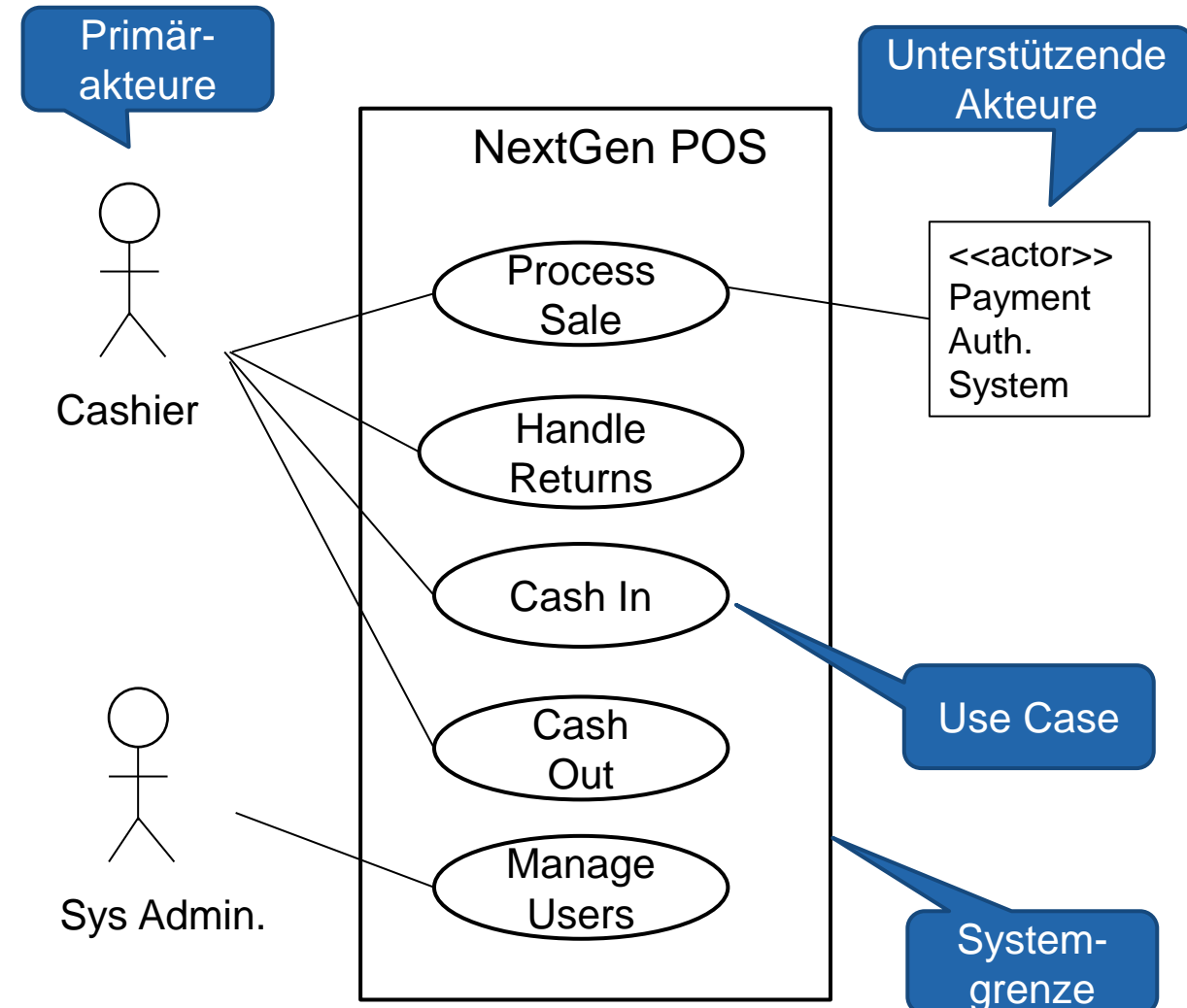


Finden von Anwendungsfällen

- Schritt 1: Systemgrenzen definieren
 - Kasse
- Schritt 2: Primärakteure identifizieren
 - Kassier
 - Frage: Wieso nicht Kunde?
- Schritt 3:
 - Jobs (Ziele, Aufgaben) der Primärakteure identifizieren
- Kassier
 - Kasse eröffnen
 - Verkauf abwickeln
 - Kasse abrechnen
 - Retouren entgegennehmen
- Kassensadministrator
 - Kasse aufsetzen
 - Kasse warten
 - Kasse ausser Betrieb nehmen
 - Kassensbenutzer administrieren

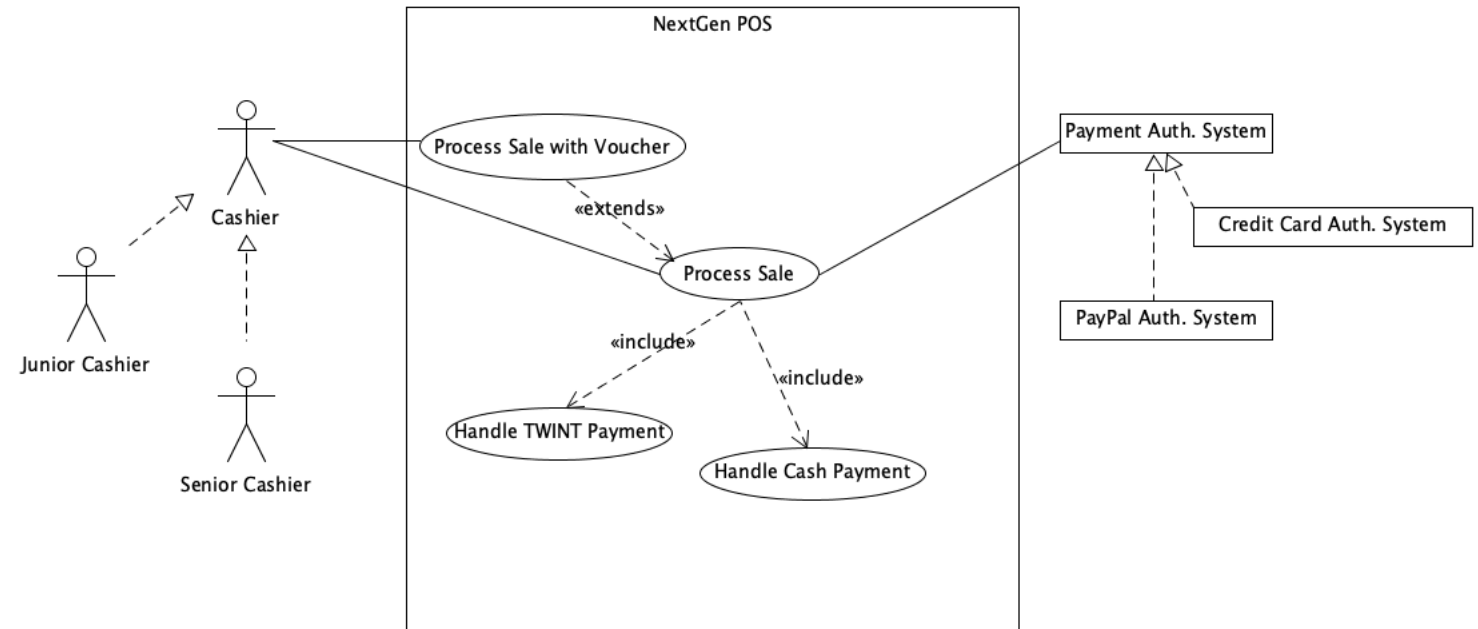
Anwendungsfalldiagramm (Use-Case-Diagramm)

- Zeigt
 - Systemabgrenzung
 - Akteure
 - Primärakteure **initiiieren** einen UC
 - Unterstützende Akteure **sind beteiligt** an einem UC
 - Liste der Anwendungsfälle
 1. Process Sale
 2. Cash In
 3. Cash Out
 4. Handle Returns
 5. Manage Users



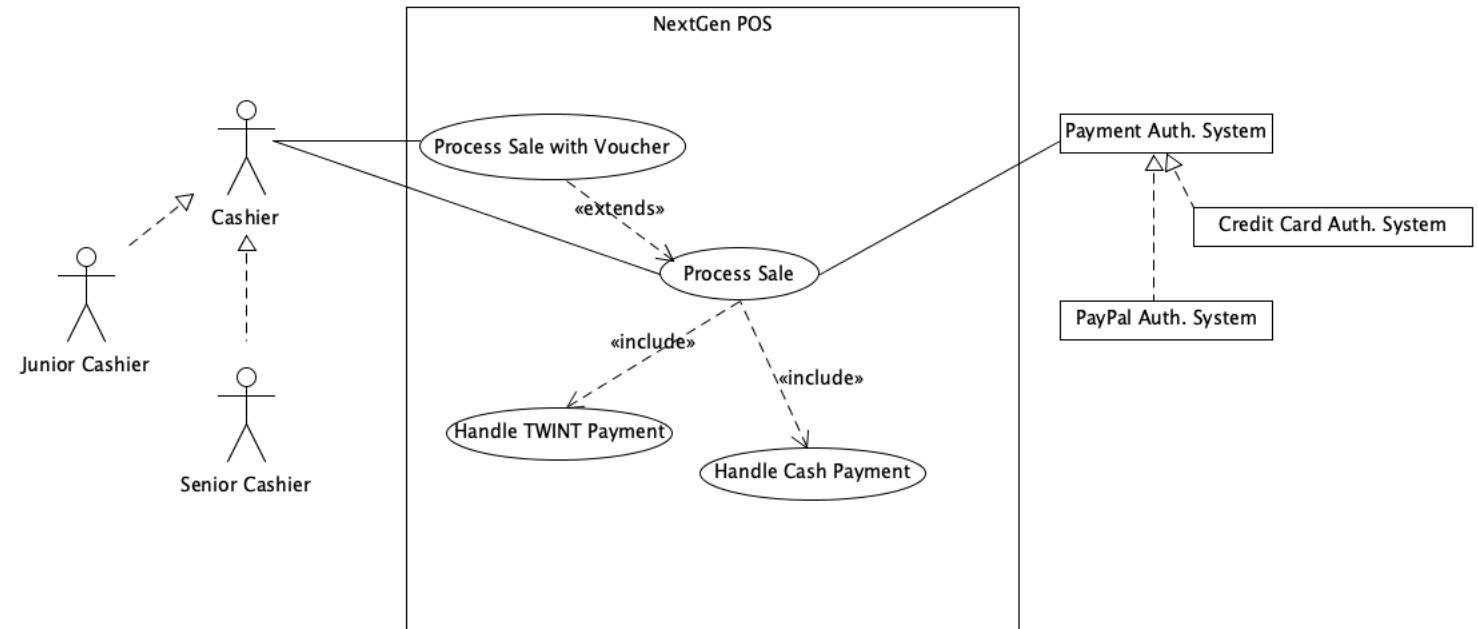
Anwendungsfalldiagramm (Use-Case-Diagramm)

- Zusätzliche Beziehungen im UC-Diagramm
 - <<include>>
 - UC “Handle Cash Payment“ und UC „Handle TWINT Payment“ sind enthalten im UC „Process Sale“
 - Sie sind aber keine eigenständigen UCs
 - Keine Verbindung zu Akteuren



Anwendungsfalldiagramm (Use-Case-Diagramm)

- Zusätzliche Beziehungen im UC-Diagramm
 - <<extends>>
 - Eigenständiger UC, der eine Erweiterung eines anderen darstellt, und
 - ursprünglicher UC nicht verändert werden soll
 - Sonst besser als Erweiterung im UC-Text einfügen
 - Akteur-Generalisierung
 - Um Akteure zusammenzufassen
 - Kann als „ist-ein“-Beziehung modelliert werden



Agenda

1. Wie erfasst man (dynamische) funktionale Anforderungen mit Use Cases
- 2. Wie schreibt man gute Use Cases**
3. System-Sequenzdiagramme, Systemoperationen und Verträge
4. Wie erfasst man zusätzliche funktionale und nicht-funktionale Anforderungen
5. Wrap-up und Ausblick

Anwendungsfälle: Brief UC

- Kurze Beschreibung des Anwendungsfalls in einem Paragraphen
 - Nur Erfolgsszenario
 - Sollte enthalten
 - Trigger des UCs
 - Akteure
 - Summarischen Ablauf des UCs
 - Wann?
 - Zu Beginn der Analyse
- Process Sale (brief)

Kunde kommt mit seinen Waren zur Kasse. Kassier erfasst alle gekauften Produkte. Am Ende berechnet Kasse den Totalbetrag. Kassier zieht das Geld von Kunden ein und gibt den Betrag in die Kasse ein. Diese berechnet das Rückgeld. Kassier gibt Kunde das berechnete Rückgeld zurück.

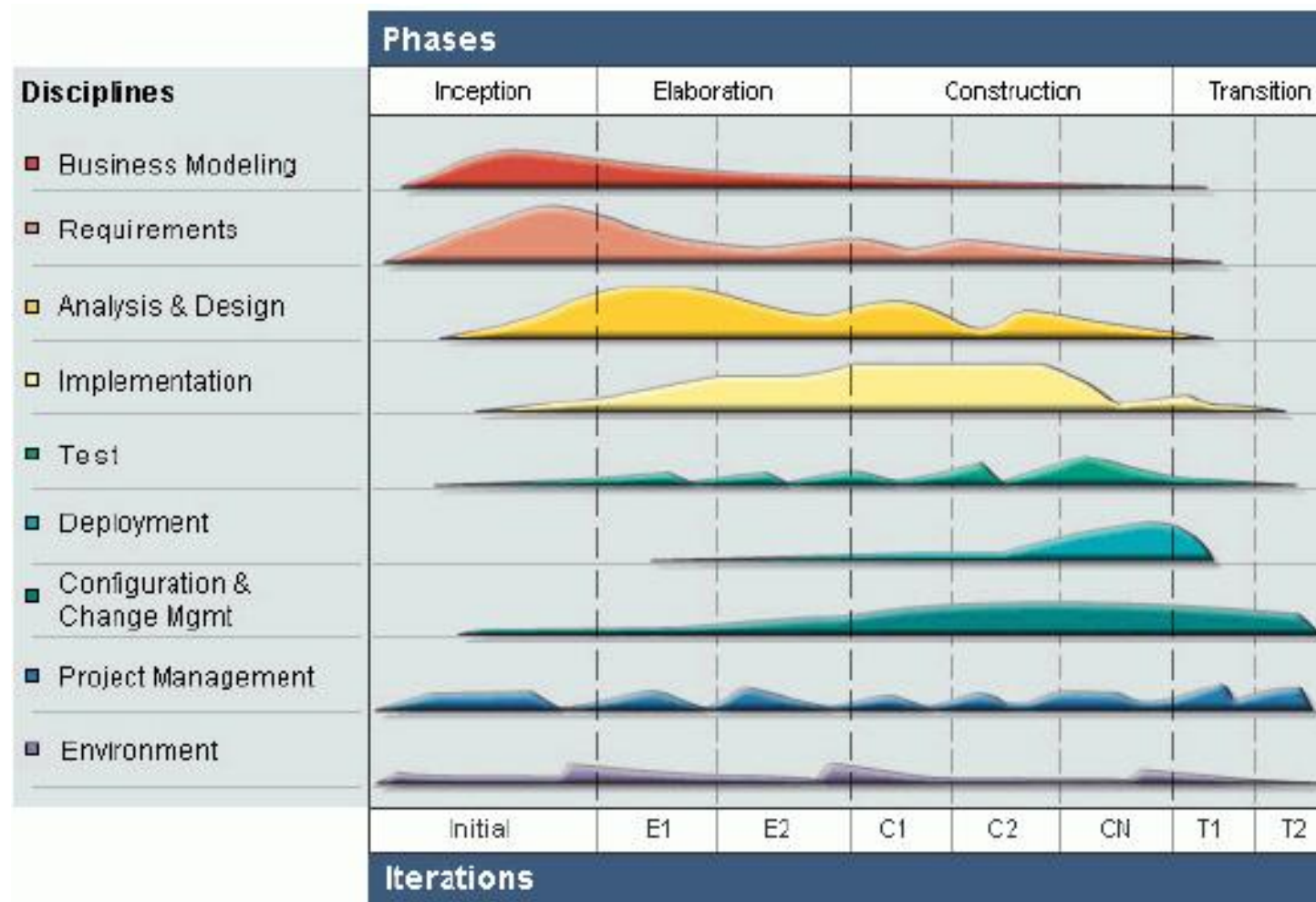
Anwendungsfälle: Casual UC

- Informelle Beschreibung des Anwendungsfalls in mehreren Paragraphen
 - Erfolgsszenario plus wichtigste Alternativszenarien
 - Sollte enthalten
 - Trigger des UCs
 - Akteure
 - Interaktion des Akteurs mit System
 - Wann?
 - Zu Beginn der Analyse
- Process Sale (casual)
 - Kunde kommt mit seinen Waren zur Kasse. Kassier scannt den Produktcode jedes Produkts ein. Kasse zeigt Artikel und Preis. Kassier korrigiert Menge, falls nötig. Bei Produkten ohne Code gibt der Kassier das Produkt und den Preis, sowie die Menge manuell ein.
 - Am Ende schliesst Kassier den Einkauf ab. Kasse zeigt den Totalbetrag. Kassier nimmt das Geld vom Kunden entgegen und gibt den bezahlten Betrag in Kasse ein. Kasse berechnet den Retourbetrag und öffnet die Geldschublade. Kassier entnimmt den Retourbetrag und übergibt das Retourgeld dem Kunden. Kassier schliesst Geldschublade und beendet damit den Verkauf.
 - Kasse druckt die Quittung aus. Kassier übergibt Quittung dem Kunden.

Anwendungsfälle: Fully-dressed UC

- Detaillierte Beschreibung des Ablaufs mit allen Alternativszenarien
 - Wann?
 - Ende der Inception- und v.a. in Elaboration-Phase (Anforderungsdisziplin)
 - Nachdem die meisten UCs identifiziert und kurz beschrieben worden sind
 - Die **wichtigsten UCs (10%)**, die die Architektur bestimmen, werden im Detail ausformuliert
- Formaler Aufbau
 - UC-Name
 - Umfang (Scope)
 - Ebene (Level)
 - Primärakteur (Primary Actor)
 - Stakeholders und Interessen
 - Vorbedingungen (Preconditions)
 - Erfolgsgarantie/Nachbedingungen (Success Guarantee)
 - Standardablauf (Main Success Scenario)
 - Erweiterungen (Extensions)
 - Spezielle Anforderungen (Special Requirements)
 - Liste der Technik und Datavariationen (Technology and Data Variations)
 - Häufigkeit des Auftretens (Frequency of Occurance)
 - Verschiedenes (Miscellaneous)

Projektablauf



Anwendungsfälle: Fully-dressed UC

- UC-Name
 - Umfang
 - Ebene
 - Primärakteur
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - Spezielle Anforderungen
 - Liste der Technik und Datavariationen
 - Häufigkeit des Auftretens
 - Verschiedenes
- Aktiv formulieren (Verb + ev. Objekt)
 - Beschreibt Job (Ziel, Aufgabe), den Akteur ausführen will
 - Beispiele
 - Verkauf abwickeln (Process Sale)
 - Waren zurücknehmen (Handle Return)
 - Spielrunde spielen (Play game round)

Anwendungsfälle: Fully-dressed UC

- UC-Name
- Umfang
- Ebene
- Primärakteur
- Stakeholders und Interessen
- Vorbedingungen
- Erfolgsgarantie/Nachbedingungen
- Standardablauf
- Erweiterungen
- Spezielle Anforderungen
- Liste der Technik und Datavariationen
- Häufigkeit des Auftretens
- Verschiedenes

Umfang:

- Beschreibt das zu entwickelnde System (SuD, System under Development)
- Beispiel: UC Process Sale
 - Kassen-Anwendung (NextGen-POS)

Ebene:

- Anwenderziel oder
- Subfunktion

Anwendungsfälle: Fully-dressed UC

- UC-Name
 - Umfang
 - Ebene
 - **Primärakteur**
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - Spezielle Anforderungen
 - Liste der Technik und Datavariationen
 - Häufigkeit des Auftretens
 - Verschiedenes
- Hauptakteur des UCs
 - Primärer Nutzniesser des UC
 - **Initiiert** den UC
 - Interagiert hauptsächlich mit dem System
 - Beispiel: Process Sale
 - Kassier

Anwendungsfälle: Fully-dressed UC

- UC-Name
 - Umfang
 - Ebene
 - Primärakteur
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - Spezielle Anforderungen
 - Liste der Technik und Datavariationen
 - Häufigkeit des Auftretens
 - Verschiedenes
- Für wen ist der UC sonst noch relevant und welche Interessen hat er daran?
 - Beispiel: Process Sale
 - Kassier (Primärakteur):
 - Will schnelle Eingabe
 - Will keine Fehler machen (Rückgeld)
 - Kunde (Stakeholder)
 - Will schnell und problemlos einkaufen
 - Will Übersicht über gekaufte Produkte, Preise
 - Will einen Kaufbeleg

Anwendungsfälle: Fully-dressed UC

- UC-Name
 - Umfang
 - Ebene
 - Primärakteur
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - Spezielle Anforderungen
 - Liste der Technik und Datavariationen
 - Häufigkeit des Auftretens
 - Verschiedenes
- Was ist die unmittelbare Voraussetzung, damit dieser UC ablaufen kann?
 - Nur wichtige, nicht offensichtliche Voraussetzungen
 - Beispiel: Process Sale
 - Kassier muss identifiziert und für Kasse autorisiert sein

Anwendungsfälle: Fully-dressed UC

- UC-Name
 - Umfang
 - Ebene
 - Primärakteur
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - Spezielle Anforderungen
 - Liste der Technik und Datavariationen
 - Häufigkeit des Auftretens
 - Verschiedenes
- Was muss nach der erfolgreichen Ausführung des UCs gewährleistet sein?
 - Beispiel: Process Sale
 - Verkauf ist gespeichert
 - Steuern sind richtig berechnet
 - Buchhaltung und Lagerbestand sind aufdatiert
 - Kaufbeleg ist erstellt
 - Zahlungsdetails sind gespeichert

Anwendungsfälle: Fully-dressed UC

- UC-Name
- Umfang
- Ebene
- Primärakteur
- Stakeholders und Interessen
- Vorbedingungen
- Erfolgsgarantie/Nachbedingungen
- **Standardablauf**
- Erweiterungen
- Spezielle Anforderungen
- Liste der Technik und Datavariationen
- Häufigkeit des Auftretens
- Verschiedenes

- **Wichtigster Teil** des UCs
- Beschreibt **Interaktion des Primärakteurs mit dem System**
 - Plus allenfalls Interaktion mit unterstützenden Akteuren
- Startpunkt ist nach den Vorbedingungen

Anwendungsfälle: Fully-dressed UC

- Standardablauf
 - Beschreibt erfolgreichen Ablauf des UCs
 - Detaillierte Interaktion des Akteurs mit dem System
 - Eingabe des Akteurs
 - Antwort des Systems
 - Keine Lösungsdetails

Beispiel: Process Sale (Larman S. 50ff/106ff)

- Standardablauf
 1. Kunde kommt mit seinen Waren an die Kasse.
 2. Kassier beginnt neuen Verkauf.
 3. Kassier erfasst Artikel.
 4. System zeigt Artikelbeschreibung, Preis und laufende Summe an.

Kassier wiederholt Schritte 1-4 für alle Artikel
 5. System zeigt Gesamtsumme und berechnete Steuern an.
 6. Kassier teilt Kunde Summe mit und bittet um Zahlung.

Anwendungsfälle: Fully-dressed UC

- Standardablauf
 - Beschreibt erfolgreichen Ablauf des UCs
 - Detaillierte Interaktion des Akteurs mit dem System
 - Eingabe des Akteurs
 - Antwort des Systems
 - Keine Lösungsdetails

Beispiel: Process Sale (Forts. Von Folie 36)

- Standardablauf
 7. Kunde bezahlt und System bearbeitet Zahlung.
 8. System protokolliert den abgeschlossenen Verkauf und sendet Verkaufs- und Zahlungsinformationen an das externe Abrechnungs- und Lagerverwaltungssystem.
 9. System präsentiert Kaufbeleg.
 10. Kunde geht mit Kaufbeleg und Waren.

Anwendungsfälle: Fully-dressed UC

- UC-Name
 - Umfang
 - Ebene
 - Primärakteur
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - Spezielle Anforderungen
 - Liste der Technik und Datavariationen
 - Häufigkeit des Auftretens
 - Verschiedenes
- Beschreibt alternative Erfolgs- aber auch Misserfolgsszenarien

Anwendungsfälle: Fully-dressed UC

- Erweiterungen/alternative Abläufe
 - 3a: Nummer zeigt an, wo im Hauptszenario alternativer Ablauf beginnt

Beispiel: Process Sale

- 3a: Ungültige Artikelbezeichnung
 1. System signalisiert Fehler
 2. Kassier reagiert auf Fehler:
 - 2a. Es gibt eine lesbare Artikelnummer
 1. Kassier gibt Artikelnummer manuell ein.
 2. System zeigt Artikelbeschreibung und Preis.
 - 2b. Es gibt keine Artikelnummer, aber einen Preis.
- ...

Anwendungsfälle: Fully-dressed UC

- Erweiterungen/alternative Abläufe
 - 7a: Nummer zeigt an, wo im Hauptszenario alternativer Ablauf beginnt

Beispiel: Process Sale

- 7a: Barbezahlung
 1. Kassier gibt gezahlten Betrag ein.
 2. System zeigt fälligen Saldo und gibt Geldschublade frei.
 3. Kassier gibt Saldo in bar an Kunden heraus und legt bezahltes Bargeld in Schublade.
 4. System registriert Bargeldzahlung.
- 7b: Zahlung per Kreditkarte
 1. ...

Anwendungsfälle: Fully-dressed UC

- Erweiterungen/alternative Abläufe
 - 3-6a: Bereich von Schritten im Hauptszenario, wo alternativer Ablauf auftreten kann

Beispiel: Process Sale

- 3-6a: Kunde bittet Kassier, einen Artikel zu stornieren:
 1. Kunde gibt Artikelnummer des zu stornierenden Artikels ein.
 2. System entfernt Artikel vom Einkauf und zeigt aktualisierte laufende Summe an.

Anwendungsfälle: Fully-dressed UC

- Erweiterungen
 - * sagt aus, dass der Alternativablauf zu jeder Zeit auftreten kann
 - Interaktion des alternativen Ablaufs analog zu Hauptszenario beschrieben

Beispiel: Process Sale

- *a: Jederzeit, wenn Verkauf abgebrochen wird.
 1. Kassier bricht Verkaufsvorgang ab.
 2. System storniert alle Buchungen des aktuellen Verkaufs.
 - 2a. Falls Kunde bereits bezahlt hat.
 1. System zeigt auszufehlenden Betrag an.
 2. System sendet Stornierungsinformation an externes Abrechnungs- und Lagersystem.
- *b: Jederzeit, wenn das System ausfällt:
 1. ...

Anwendungsfälle: Fully-dressed UC

- UC-Name
 - Umfang
 - Ebene
 - Primärakteur
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - **Spezielle Anforderungen**
 - Liste der Technik und Datavariationen
 - Häufigkeit des Auftretens
 - Verschiedenes
- Weitere Anforderungen, die aus diesem UC resultieren
 - Beispiel: Process Sale
 - Touchscreen UI. Text muss aus 1m Entfernung lesbar sein
 - Antwortzeit bei Kreditautorisierung innerhalb von 30s in 90% der Fälle
 - Internationalisierung der Textanzeigen muss vorbereitet sein
 - ...

Anwendungsfälle: Fully-dressed UC

- UC-Name
 - Umfang
 - Ebene
 - Primärakteur
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - Spezielle Anforderungen
 - Liste der Technik- und Datavariationen
 - Häufigkeit des Auftretens
 - Verschiedenes
- Alternative I/O-Methoden, Datenformate, etc.
 - Beispiel: Process Sale
 - 3a: Eingabe der Artikelnummer mit Barcodeleser oder per Tastatur
 - 3b: Artikelnummer können wahlweise mit folgenden Systemen codiert sein: UPC, EAN, JAN oder SKU

Anwendungsfälle: Fully-dressed UC

- UC-Name
 - Umfang
 - Ebene
 - Primärakteur
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - Spezielle Anforderungen
 - Liste der Technik und Datavariationen
 - Häufigkeit des Auftretens
 - Verschiedenes
- Wie häufig tritt dieser UC auf?
 - Einmal, regelmässig, häufig, fast immer
 - Bestimmt Wichtigkeit des UCs z.B. bezüglich Performance, Testen
 - Beispiel: Process Sale
 - Häufigkeit des Auftretens: laufend.

Anwendungsfälle: Fully-dressed UC

- UC-Name
 - Umfang
 - Ebene
 - Primärakteur
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - Spezielle Anforderungen
 - Liste der Technik und Datavariationen
 - Häufigkeit des Auftretens
 - **Verschiedenes**
- Offene Fragen/Probleme
 - Beispiel: Process Sale
 - Welche Steuersätze sind zu berücksichtigen?
 - Muss Kassier Geldschublade mitnehmen, wenn er sich abmeldet?
 - Ausführliches Beispiel des UC Process Sale
 - Craig Larman S. 106ff (deutsch), S. 50ff (englisch)

Anforderungsanalyse: Anwendungsfälle (Use Cases)

- Eigenschaften guter Anwendungsfälle
 - Aussagekräftiger Titel
 - Beschreibt Anwenderziel, aktiv formuliert
 - Beispiel (Akteur Kassier): „Process Sale“ (Einen Verkauf abwickeln)
 - Essentieller Stil (nicht konkreter Stil)
 - Beschreibt Logik der Interaktion, nicht konkrete Umsetzung
 - Beispiel (Akteur Kassier):
 - Konkret: „Kassier tippt die Produkt-ID ein. System zeigt Produktnamen.“
 - Essentiell: „Kassier erfasst das Produkt. System bestätigt Produkt.“
(z.B. durch Eintippen, Wählen, Scannen, oder per Sprache)

Anforderungsanalyse: Anwendungsfälle (Use Cases)

- Eigenschaften guter Anwendungsfälle
 - Knappe aber präzise Beschreibung der Interaktion des Standardablaufs
 - Keine kann-Formulierungen
 - Alternative Interaktionen sind unter Erweiterungen aufgeführt
 - Nur Aussensicht (Benutzersicht), keine systeminternen Interaktionen

Anwendungsfälle, und wie weiter?

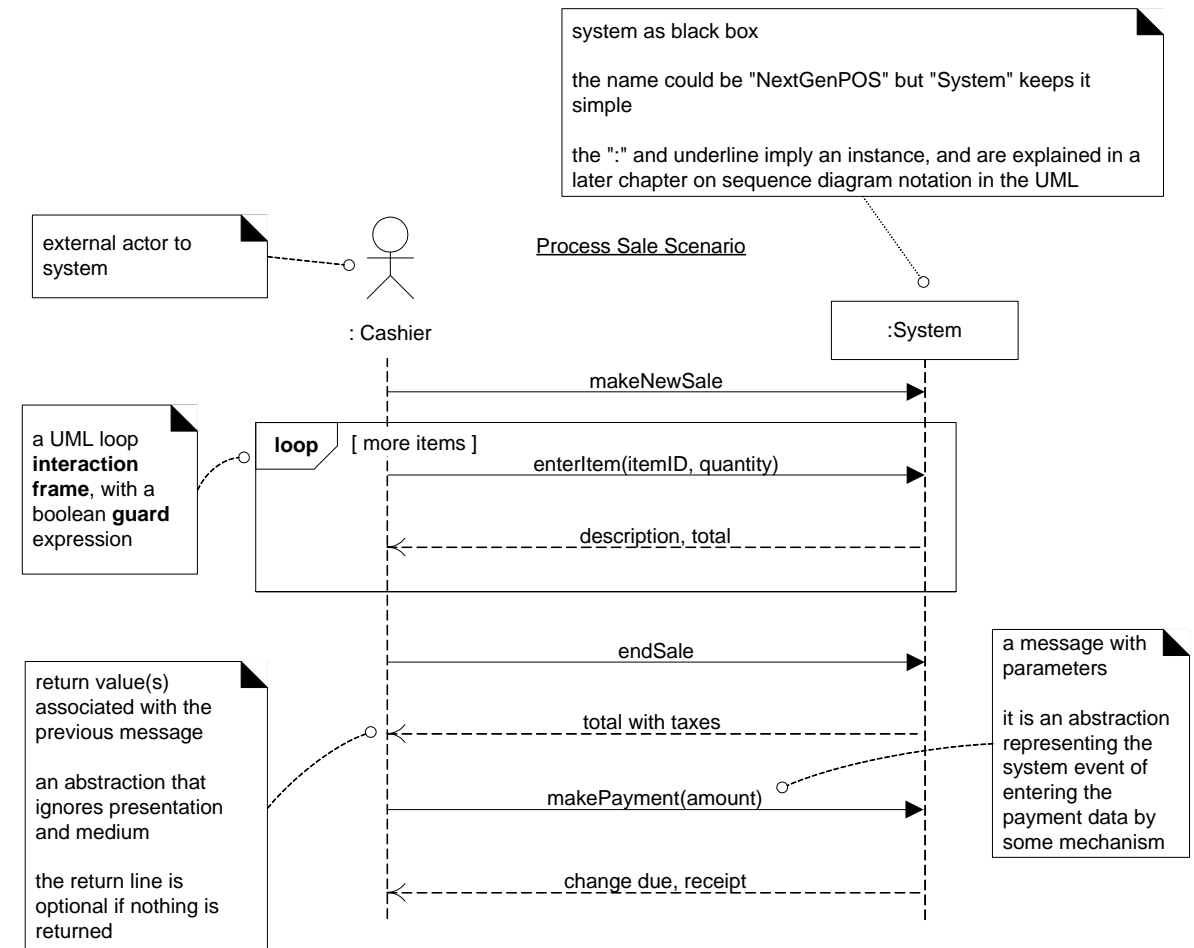
- Frage
 - Wie kommen wir von den Anwendungsfällen, die auf abstrakter Ebene die funktionalen Anforderungen an das System beschreiben, zu den konkreten Funktionalitäten, die das System aufweisen muss?
- Antwort
 - Systemsequenzdiagramme
 - Operation Contracts

Agenda

1. Wie erfasst man (dynamische) funktionale Anforderungen mit Use Cases
2. Wie schreibt man gute Use Cases
3. **System-Sequenzdiagramme, Systemoperationen und Verträge**
4. Wie erfasst man zusätzliche funktionale und nicht-funktionale Anforderungen
5. Wrap-up und Ausblick

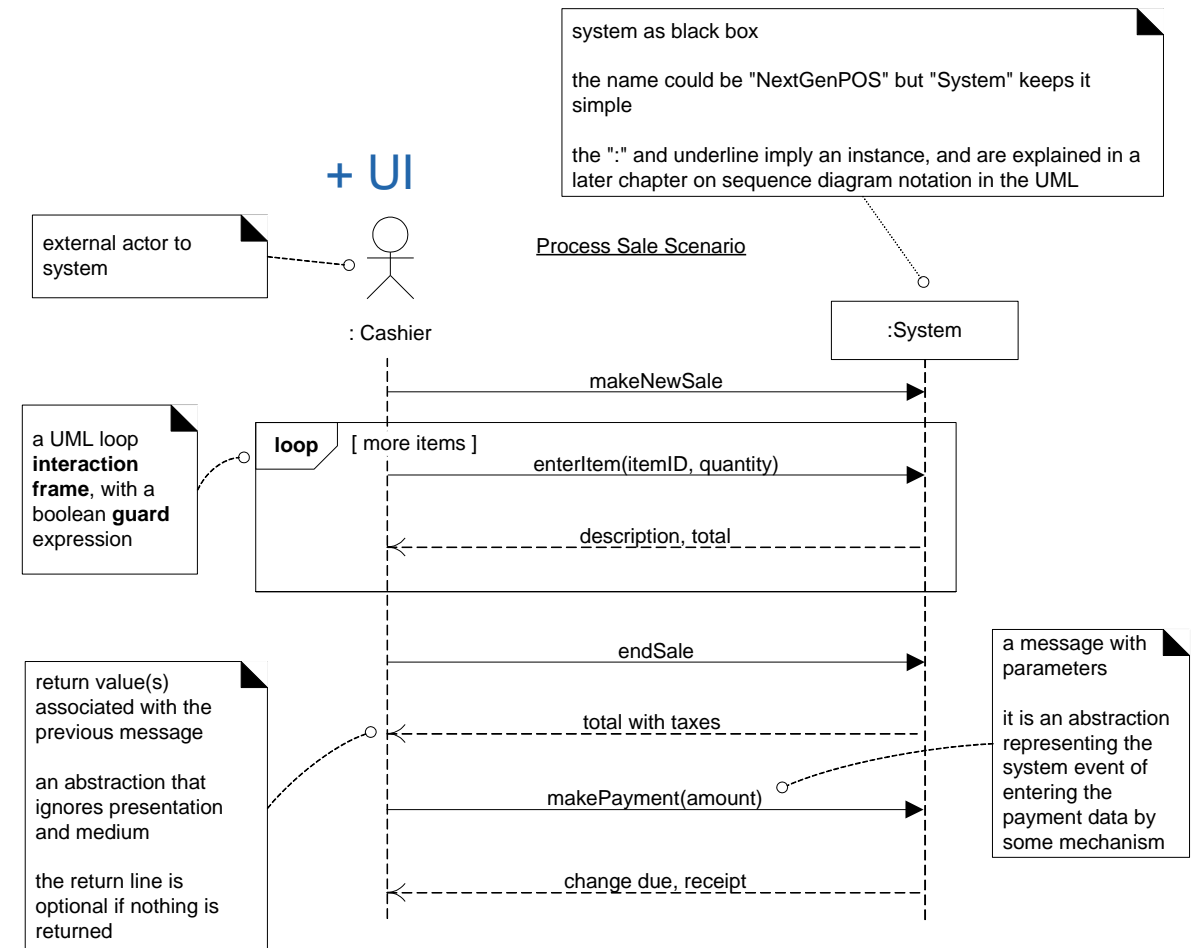
Systemsequenzdiagramm (SSD)

- Ist formal ein UML Sequenzdiagramm
- Zeigt Interaktionen der Akteure mit dem System
 - Welche Input-Events auf das System einwirken
 - Welche Output-Events das System erzeugt
- Ziel
 - Wichtigste Systemoperationen identifizieren, die das System zur Verfügung stellen muss (API) für einen gegebenen Anwendungsfall

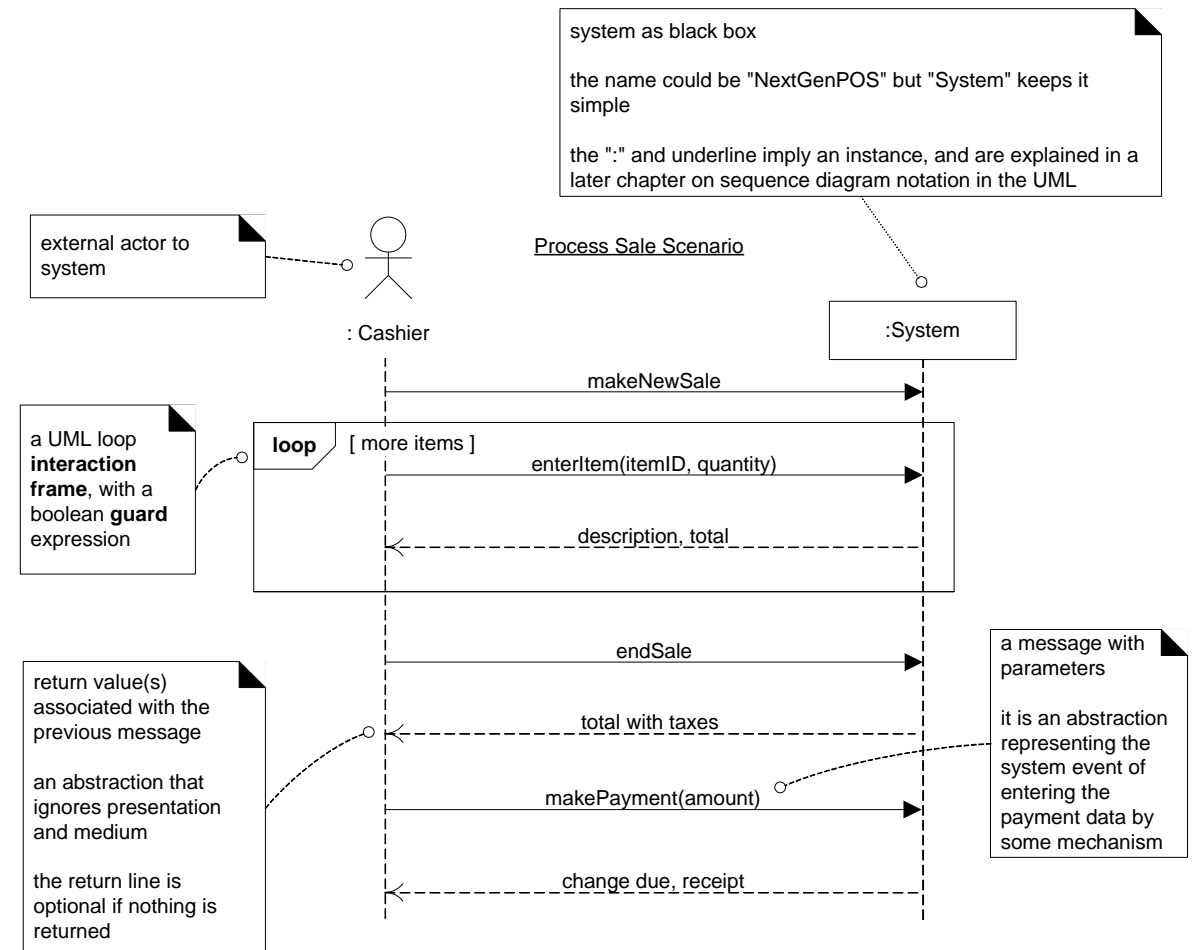


Systemsequenzdiagramm (SSD)

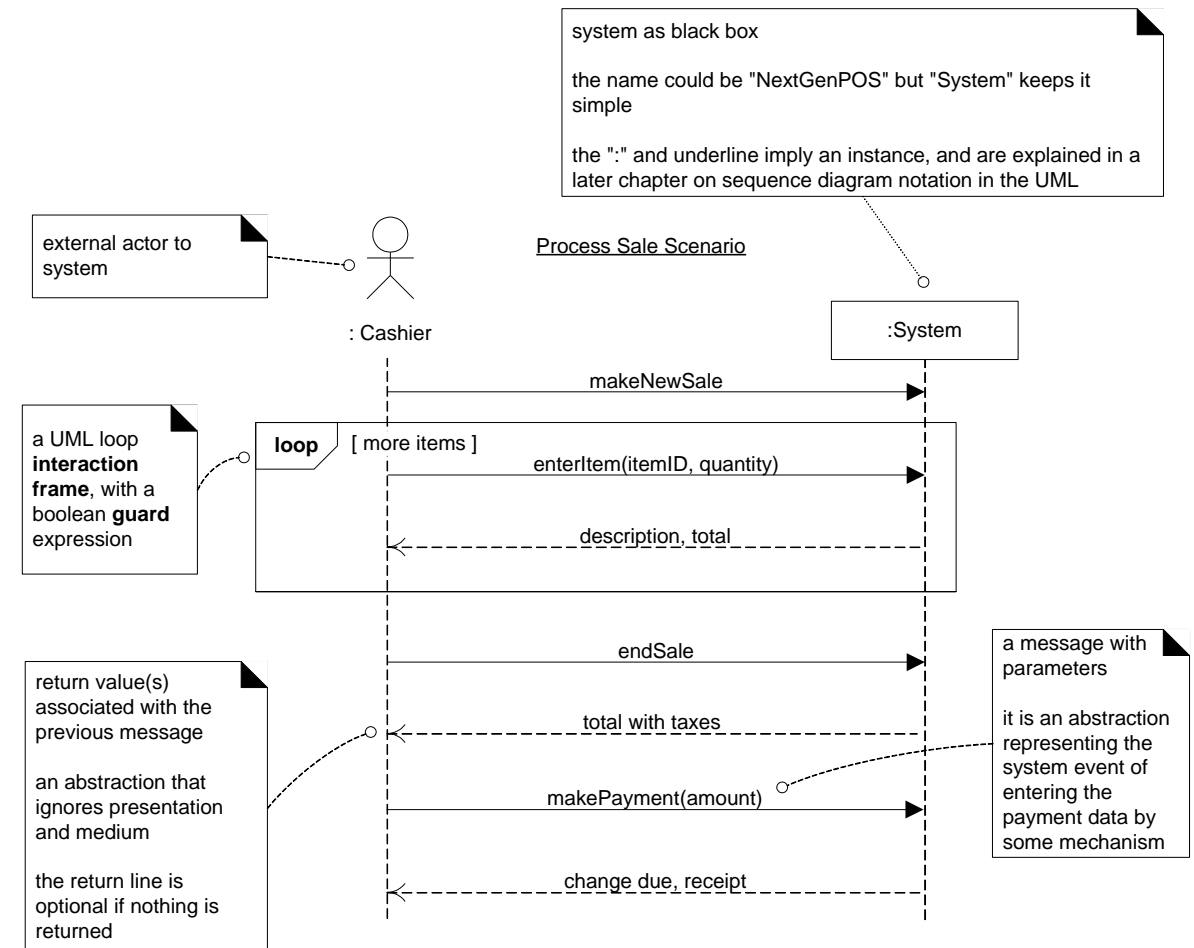
- Links ist Primärakteur aufgeführt
 - Hier Cashier
 - Inkl. seiner Benutzerschnittstelle
 - Initiiert die Systemoperationen (via UI)
 - UI findet zusammen mit Akteur heraus, was dieser tun möchte
 - UI ruft sodann entsprechende Systemoperation auf
- Mitte das System (:System)
 - Muss die Systemoperationen zur Verfügung stellen
- Rechts
 - Sekundärakteure, falls nötig



- Formal wie ein Methodenaufruf
 - Treffender Name, der die Absicht des Akteurs repräsentiert
 - Evtl. mit Parametern
 - Information, die für die Ausführung der Systemoperation nötig sind, aber noch nicht im System vorhanden sind
 - Details zu den Parametern sollten im Glossar erläutert werden
 - Durchgezogener Pfeil für Methodenaufruf
 - Rückgabewert
 - Kann fehlen, falls unwichtig
 - Kein Methodenaufruf, sondern indirektes Update des UI (deshalb gestrichelte Linie)



- Beispiel: enterItem(itemID, quantity)
 - Systemoperation, um einen/mehrere Artikel vom gleichen Typ zu erfassen
 - Braucht als Info (Vom Akteur bzw. UI)
 - Welcher Artikeltyp (Artikel-ID)
 - Menge der gekauften Artikel (int)
 - Rückgabewert (Systemantwort)
 - Im UI erscheint die Beschreibung des Artikels, die Menge und der Gesamtbetrag
 - System ruft nicht direkt das UI auf, sondern das UI wird indirekt aufdatiert (Model-View-Trennung)
 - Deshalb gestrichelter Pfeil



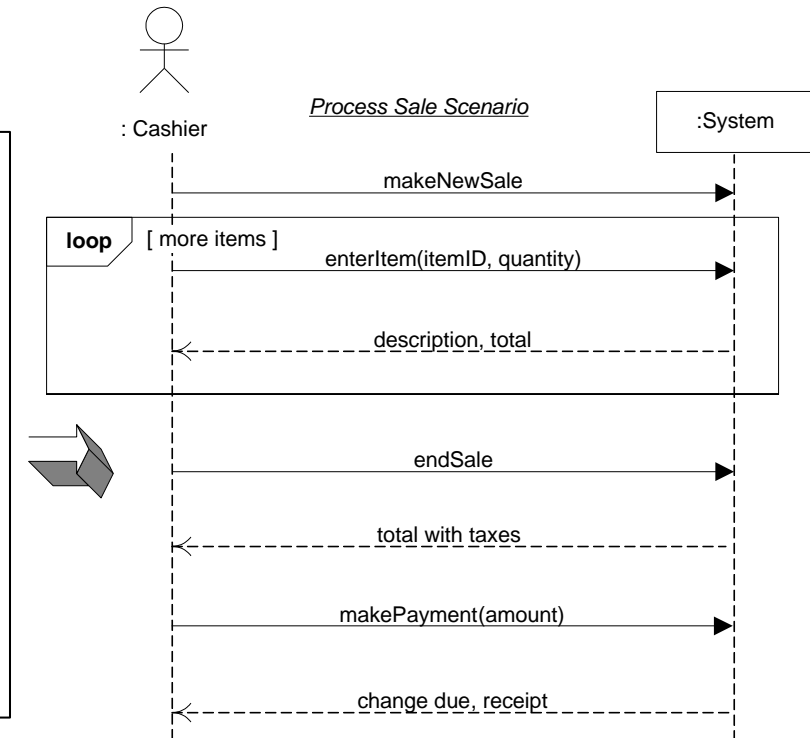
Systemoperation

Wie findet man Systemoperationen?

- **Szenario** des UCs Schritt für Schritt durchgehen
- Für jeden Schritt des Akteurs überlegen, **welche Systemoperation es dafür braucht**
 - Geeigneten präzisen Namen wählen
 - Aus Sicht Akteur
 - Welche Info braucht das System, um diese Systemoperation auszuführen?
 - Falls noch nicht im System vorhanden -> Parameter

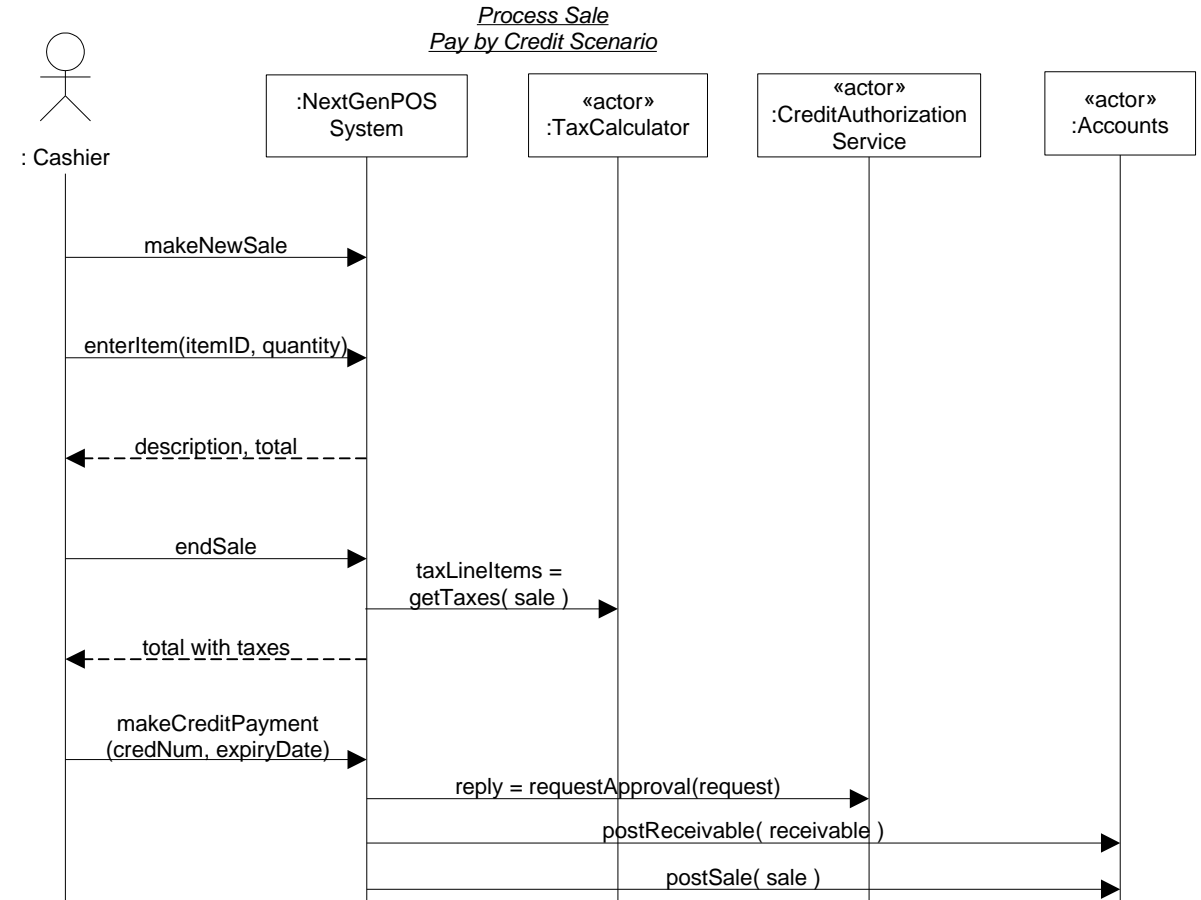
Simple cash-only *Process Sale* scenario:

1. Customer arrives at a POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total.
Cashier repeats steps 3-4 until indicates done.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
- ...



SSD für Interaktionen zwischen Systemen

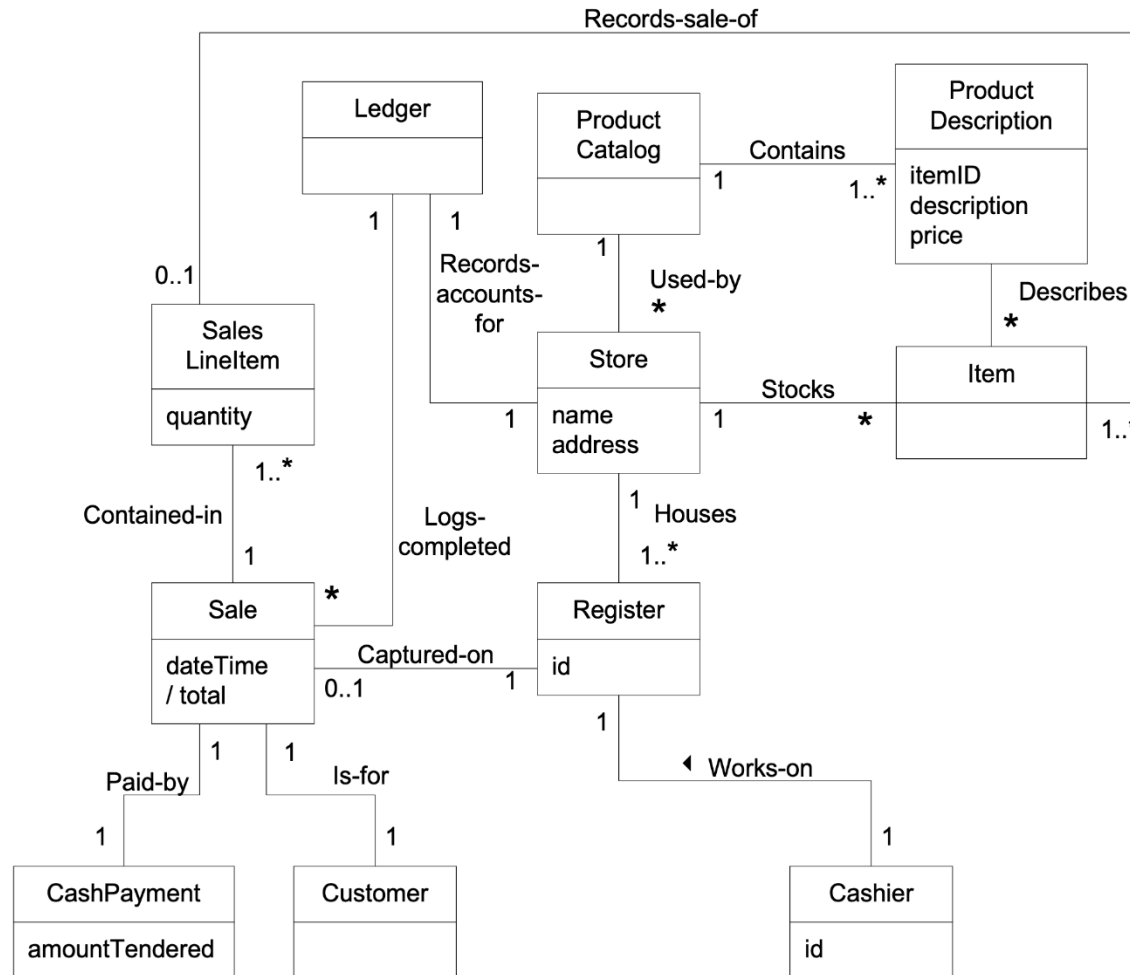
- SSD können auch Interaktionen zwischen SuD und externen unterstützenden System zeigen



- ## Contract CO2: enterItem

- ## Zeitform beachten!

Operation Contract



Contract CO2: enterItem

- **Operation:**
`enterItem(idemID: ItemID,
 quantity: integer)`
- **Querverweis:** UC Process Sale
- **Vorbedingungen:**
 - Verkauf muss gestartet sein
- **Nachbedingungen**
 - SaleLineItem-Instanz sli erstellt
 - sli mit aktueller Sale-Instanz verknüpft
 - sli.quantity auf quantity gesetzt
 - sli mit entsprechender ProductDescription verknüpft (gemäss itemID)

Operation Contract

- Wann Operation Contracts?
 - Nur wenn aus einem Anwendungsfall nicht klar wird, was die Systemoperation genau machen muss
 - Meist nur bei **sehr komplizierten Operationen** und/oder
 - Wenn **Entwicklung** der Systemoperation **ausgelagert wird** (anderes Team, externe Entwickler)
 - Erst gegen **Ende des Meilensteins Lösungsarchitektur** oder kurz vor Start des Designs der Systemoperation

Nutzen von SSD und Systemoperationen

- Systemoperationen definieren die Schnittstelle (API) des Systems
- Während dem Design
 - wird das System ausgehend von den Systemoperationen entwickelt (anhand der Verträge)
 - Das UI-Team kann parallel das UI entwickeln unter Verwendung der vereinbarten Systemoperationen (und ihren Verträgen)
 - SSD können auch zur Darstellung der Kommunikation von Subsystemen verwendet werden (z.B. bei Client-Server-Architektur)
- Achtung Frameworks!
 - UI- und andere Frameworks geben häufig gewisse Systemoperationen vor, die das System implementieren muss
 - Sollte man bei den SSD bereits berücksichtigen

Agenda

1. Wie erfasst man (dynamische) funktionale Anforderungen mit Use Cases
2. Wie schreibt man gute Use Cases
3. System-Sequenzdiagramme, Systemoperationen und Verträge
4. **Wie erfasst man zusätzliche funktionale und nicht-funktionale Anforderungen**
5. Wrap up und Ausblick

Weitere Anforderungen

- Die **UCs** beschreiben einen grossen Teil der funktionalen Anforderungen aus Benutzersicht
- Es gibt aber **weitere funktionale und nicht-funktionale Anforderungen** (Qualitätsanforderungen und Randbedingungen), die schlecht in UCs beschrieben werden können
- Diese werden als **zusätzliche Anforderungsspezifikation** formuliert (Supplementary Specification)
- **Zusätzliche Anforderungen**
 - Können als **Anforderungsstatement** oder als **User-Story** (agile SWE) formuliert werden
 - Bsp:
 - Anforderungsstatement:
„Das Kassensystem muss in weniger als 1 Minute aufgestartet sein“
 - User-Story:
Als Kassier möchte ich, dass bei mehreren gleichen Artikeln der Einzelpreis, die Anzahl und der Gesamtpreis angezeigt werden, damit ich einen schnellen Überblick habe.

Weitere Anforderungen

- Anforderungsstatements
 - Sollten **als Anforderung formuliert** werden
 - Das System muss/soll mindestens/darf nicht...
 - Sollten **messbar/verifizierbar** sein
 - Sie müssen dem Auftraggeber irgendwann belegen, dass ihr System diese Anforderung erfüllt
- So wenig wie nötig
 - Nur Anforderungen stellen, die auch wirklich von jemandem begründet gefordert werden
 - **Keine ersten Lösungsideen** als Forderungen formulieren
 - „Das System muss eine Web-App sein“
- User-Stories
 - Sagen in einem Satz **wer, was, warum fordert**
 - Erfüllen damit einige der Bedingungen automatisch
 - Die anderen sollten aber auch bei User-Stories erfüllt sein
 - Messbarkeit/Verifizierbarkeit (z.B. durch Akzeptanzkriterien)

Weitere Anforderungen: FURPS+

- Checkliste für zusätzliche Anforderungen
 - **Functionality (Funktionalität)**
 - Features, Fähigkeiten, Sicherheit
 - **Usability (Gebrauchstauglichkeit)**
 - Siehe Usability-Anforderungen (LE02)
 - Accessibility (Benutzer mit spez. Bedürfnissen)
 - **Reliability (Zuverlässigkeit)**
 - Fehlerrate, Wiederanlauffähigkeit, Vorhersagbarkeit, Datensicherung
 - **Performance (Performanz)**
 - Reaktionszeiten, Durchsatz, Genauigkeit, Verfügbarkeit, Ressourceneinsatz
 - **Supportability (Unterstützbarkeit)**
 - Anpassungsfähigkeit, Wartbarkeit, Internationalisierung, Konfigurierbarkeit
 - **+**
 - Implementation
 - HW, Betriebssysteme, Sprachen, Tests, Werkzeuge,..
 - Interface
 - Schnittstellen von ext. Systemen, Protokolle
 - Operations
 - Betriebliche Aspekte
 - Packaging (Verpackung)
 - Auslieferung physisch, logisch (Container, Plugin,..)
 - Legal
 - Lizenzen, rechtl. Rahmenbedingungen

Glossar

- Einfaches Glossar
 - Definiert die **Begriffe, die in diesem Projekt und im SW-Produkt verwendet** werden
 - Kann **beliebige Elemente** enthalten
 - Wichtige Begriffe, Konzepte des Domänenmodells, Attribute, Parameter von Operationen
- Data Dictionary
 - Definiert **zusätzlich Datenformate, Wertebereiche, Validierungsregeln**
- Empfehlungen Glossar
 - **So früh wie möglich** mit Glossar **beginnen**, sobald der erste Begriff auftaucht, der nicht sofort allen klar ist
 - **Fortwährend aufdatieren**, sobald weitere Begriffe, Konzepte etc. auftauchen, die nicht allen sofort klar sind
 - In der Kommunikation untereinander, die **Begriffe des Glossars verwenden**
 - Fördert Klarheit in der Kommunikation unter den Projektmitgliedern
 - Für dieselbe Sache **nur einen Begriff** verwenden (wenn möglich)

Agenda

1. Wie erfasst man (dynamische) funktionale Anforderungen mit Use Cases
2. Wie schreibt man gute Use Cases
3. System-Sequenzdiagramme, Systemoperationen und Verträge
4. Wie erfasst man zusätzliche funktionale und nicht-funktionale Anforderungen
5. **Wrap-up und Ausblick**

Wrap Up

- Ausgehend von den Kontextszenarien werden die funktionalen Anforderungen immer mehr konkretisiert.
 - Anwendungsfälle (Use Cases)
 - Beschreiben die konkrete Interaktion der Akteure mit dem System (lösungsneutral)
 - Systemsequenzdiagramme (System Sequence Diagram)
 - Identifizieren die notwendigen Systemoperationen, die das System für die Anwendungsfälle zur Verfügung stellen muss
 - Systemoperationen und Verträge (Operation Contracts)
 - Spezifizieren die Systemoperationen im Detail

Ausblick

- In dieser Lerneinheit haben Sie gelernt, wie Sie «dynamische» funktionale Anforderungen aus den Interaktionsbedürfnissen der Benutzer ableiten können.
- In der nächsten Lerneinheit werden Sie lernen,
 - wie Sie die Problemdomäne an sich im Detail verstehen und kommunizieren können.

Quellenverzeichnis

- [1] M. Richter and M. D. Flückiger, Usability und UX kompakt: Produkte für Menschen, 4th ed. Springer Vieweg, 2016.
- [2] Larman, C.: UML 2 und Patterns angewendet, mitp Professional, 2005
- [3] Seidel, M. et al.: UML @ Classroom: Eine Einführung in die objektorientierte Modellierung, dpunkt.verlag, 2012
- [4] Martin, R. C.: Clean Architecture: A Craftsman's Guide to Software Structure and Design, mitp Professional, 2018