

Use Case

Das Entwicklerteam von Teacher-Data hat einen neuen Auftrag bekommen. Sie sollen für die Automaten-Videothek DVD Rental die Software entwickeln. Nach den ersten Gesprächen konnten folgende Informationen zusammengetragen werden:

- DVD-Verleih-Automat:
 - Der Kunde soll einfach, schnell und günstig am Automaten 24 Stunden, sieben Tage die Woche DVDs ausleihen können. Der Kunde hat des Weiteren die Möglichkeit, Filme vorab über das Internet zu reservieren. In der DVD-Verleihstation gibt es drei verschiedene Automaten (siehe Zeichnung).
 - Um DVDs ausleihen zu können, müssen sich Neukunden während der Service- und Anmeldezeiten der DVD-Verleihstation bei einem Mitarbeiter anmelden. Dabei werden die benötigten Daten (*Name, Vorname, Geburtsdatum, Anschrift, Personalausweisnummer, Guthaben, Kennwort*) des Kunden aufgenommen und im System gespeichert. Zur persönlichen Identifikation des Kunden wird anschließend der Fingerabdruck am Automaten eingelesen. Der Kunde bekommt eine Magnetkarte ausgehändigt, um sich an den Automaten anmelden zu können.
 - Beim DVD-Ausleih muss sich der Kunde mit der Magnetkarte und seinem Fingerabdruck am Automaten für die Filmauswahl identifizieren. Am Eingabe-Terminal hat der Kunde die Möglichkeit Filme nach *Genre, Schauspieler, Titel, TOP-10, Neuerscheinungen* und *Stichworten* zu suchen. Suchergebnisse werden mit folgenden Informationen auf dem Display präsentiert: *Titel des Films, Regisseur, Hauptdarsteller, Kurzbeschreibung, Erscheinungsjahr, Altersfreigabe* und *Verfügbarkeit*. Nach der Wahl eines Films, erhält der Kunde seine Magnetkarte zurück und kann den Film am Automaten für die Ausund Rückgabe nach erneuter Identifikation durch die Karte und den Fingerabdruck abholen. Die DVD wird über den Ausgabeschacht ausgegeben.
 - Bei der Rückgabe einer DVD muss sich der Kunde am Automaten für die Aus- und Rückgabe von Filmen mit der Magnetkarte und seinem Fingerabdruck am Automaten identifizieren. Nach erfolgreicher Identifikation führt der Kunde die DVD in den Rückgabeschacht ein. Der Automat liest daraufhin den Barcode auf der DVD und sortiert diese eigenständig wieder ein und bucht die Leihgebühr vom Guthaben auf der Karte ab.
 - Das Guthaben kann am Automaten zum Aufladen des Guthabens aufgeladen werden. Nach Eingabe der Kundenkarte können Münzen (5, 10, 20, 50 Cent, 1 und 2€) und Scheine (5, 10, 20, 50€) bar eingezahlt werden. Bei einem Betrag von 20€ gibt es einen Bonus von 3€, bei 30€ einen von 5€ und bei einem von 50€ von 10€. Die Karte wird bei beendeter Geldeingabe nach Drücken des roten Knopfes wieder ausgeworfen.

- Reservierung von Filmen über das Internet: Nach erfolgreicher Anmeldung mit Kundennummer und Geheimzahl auf der betreiber-eigenen Homepage kann der Kunde nach *Genre*, *Schauspieler*, *Titel*, *TOP-10*, *Neuerscheinungen* und *Stichworten* Filme suchen. Suchergebnisse werden mit folgenden Informationen präsentiert: *Titel des Films*, *Regisseur*, *Hauptdarsteller*, *Kurzbeschreibung*, *Erscheinungsjahr*, *Alters-freigabe* und *Verfügbarkeit*. Der Kunde kann nach der Suche einzelne Filme reservieren, die er innerhalb von zwei Stunden abholen muss. Sollte dies nicht in dem Zeitraum geschehen, so werden die Filme nach zwei Stunden wieder zum Ausleihen freigegeben.
 - Weitere Rahmenbedingungen:
 - Wird die Kundenkarte im Automaten vergessen, so wird sie zur Si-cherheit automatisch eingezogen und kann während der Servicezei-ten wieder abgeholt werden.
 - Die maximale Leihdauer beträgt 10 Tage.
 - Hat der Kunde versehentlich den falschen Film ausgeliehen, so kann er diese bis 10 Minuten nach Ausgabe der DVD ohne Berechnung einer Leihgebühr wieder am Automaten zurückgeben.
 - Die Leihgebühr für jeden angefangenen Tag (24h) beträgt 2€.
 - Zu jedem Film gibt es mehrere Exemplare.
 - Die Ausgabe und Einsortierung von DVDs übernimmt ein DVD-Archiv-Roboter, der über eine Schnittstelle angesteuert werden muss.
- (a) Identifizieren Sie die Akteure des oben beschriebenen Systems.

Kunde, Service-Mitarbeiter, Wartung, DVD-Archiv-Roboter

- (b) Identifizieren Sie die nicht-funktionalen und funktionalen Anforderungen (als Use Cases).

Nicht-funktional: Bedienoberfläche Web / Automat gleich, einfache Handhabbarkeit, schnell, Dauerbetrieb, Ausfallquote, Leihdauer

Funktional: Geldkarte aufladen, Film suchen, Film auswählen, Film entnehmen, Film zurückgeben, Kunden aufnehmen, Film reservieren

- (c) Geben Sie eine formale Beschreibung für drei Use Cases an. Orientieren können Sie sich an folgendem Beispiel:

Geldkarte aufladen

Use Case:	Geldkarte aufladen
Ziel:	Auf der Geldkarte befindet sich ein gewünschter Betrag an Geld.
Kategorie:	primär
Vorbedingung:	Geldkarte befindet sich im Automaten.
Nachbedingung:	Auf der Geldkarte befindet sich der gewünschte Betrag.
Erfolg:	Betrag auf der Karte ist der gleiche wie davor → Fehlermeldung
Nachbedingung:	
Fehlschlag:	
Akteure:	Kunde
Auslösendes Ereignis:	Geldkarte wird in den Automaten zum Aufladen des Guthabens geschoben.
Beschreibung:	<ul style="list-style-type: none"> (i) Kunde schiebt Karte in den Automaten. (ii) Kunde wirft Münzen oder gibt einen Geldschein ein. (iii) Kunde drückt Knopf zum Auswerfen der Karte. (iv) Karte wird ausgegeben.
Erweiterungen:	Schein oder Münze wird nicht akzeptiert. → Fehlermeldung
Alternativen:	Mindest-Guthaben auf der Karte

Film suchen

Use Case:	Film suchen
Ziel:	Anzeigen des gesuchten Films
Kategorie:	primär
Vorbedingung:	Kunde hat sich mit Magnetkarte und Fingerabdruck identifiziert.
Nachbedingung:	Eine Liste von Filmen wird angezeigt.
Erfolg:	
Nachbedingung:	Kein Film gefunden → Fehlermeldung
Fehlschlag:	
Akteure:	Kunde
Auslösendes Ereignis:	Kunde will einen Film ausleihen.
Beschreibung:	Kunde gibt Genre, Schauspieler, Titel oder Stichwort ein oder lässt sich Neuerscheinungen und TOP10 auflisten.
Erweiterungen:	–
Alternativen:	Vorgaben von weiteren Suchkriterien

Kunden aufnehmen

Use Case:	Kunden aufnehmen
Ziel:	Kunde ist in der Kundendatei.
Kategorie:	primär
Vorbedingung:	Kunde ist noch nicht in der Kundendatei vorhanden.
Nachbedingung	Kunde ist als Mitglied in der Kundendatei aufgenommen.
Erfolg:	
Nachbedingung	Kunde kann nicht in die Datei aufgenommen werden.
Fehlschlag:	
Akteure:	Kunde, Service-Mitarbeiter
Auslösendes Ereignis:	Kunde stellt zu den Öffnungszeiten bei einem Service-Mitarbeiter einen Mitgliedsantrag.
Beschreibung:	<p>(i) Kunde meldet sich zu den Öffnungszeiten bei einem Service-Mitarbeiter an.</p> <p>(ii) Service-Mitarbeiter erfasst nötige Daten (Name, Vorname, Geburtsdatum, Anschrift, Personalausweisnummer, Kennwort).</p> <p>(iii) Service-Mitarbeiter kassiert einen Startbetrag und erfasst ihn bei den Kundendaten.</p> <p>(iv) Der Fingerabdruck des Kunden wird erfasst und gespeichert.</p> <p>(v) Kunde bekommt eine Magnetkarte mit seinen Daten.</p>
Erweiterungen:	-
Alternativen:	Kunde ist bereits im System und lässt nur die Kundendaten ändern.

- (d) Verfeinern Sie nun die Use Cases, indem Sie ähnliche Teile identifizieren und daraus weitere Use Cases bilden.

Identifizierung des Kunden an den Automaten, Filmsuche

- (e) Nennen Sie mindestens drei offene Fragen, die in einem weiteren Gespräch mit dem Kunden noch geklärt werden müssten.¹

- Was passiert nach der maximalen Leihdauer?
- Was passiert, wenn der Finger nicht zur Karte passt?

¹Quelle: Universität Stuttgart, Institut für Automatisierungs- und Softwaretechnik Prof. Dr.-Ing. Dr. h. c. P. Göhner

- Was passiert mit defekten DVDs?
- Was passiert bei Rückgabe der falschen DVD?

Wasserfallmodell

Die Firma Teacher-Data soll für einen Auftraggeber ein Informationssystem entwickeln. Das Entwicklerteam entscheidet sich, nach dem *Wasserfallmodell* vorzugehen.

- (a) Geben Sie an, welche *Phasen* dabei durchlaufen werden und erläutern Sie kurz deren *Zweck / Intention / Aufgabe*.

(i) Problem- und Systemanalyse

Grobe Formulierung der Ziele (z. B. Einsatzplattform Windows, einheitliche Software zur Unterstützung des Verkaufs, ...)

- Erfassung des Problembereichs (z. B. Interaktion zwischen Händler und Hersteller soll komplett über das Netz funktionieren.)
- Machbarkeitsstudie (z. B. Händler sind bereit, sich an das zu entwickelnde System anzupassen.)
- Analyse des Ist-Zustandes (z. B. Händler A bietet Interface X wohingegen Händler B Interface Z bietet. Es wird bereits teilweise bei Händler C über das Netz bestellt. ...)
- Anforderungsspezifikation (Lastenheft) (z. B. Einheitliche Oberfläche für alle Benutzer beim Hersteller, möglichst wenige Veränderungen für die Mitarbeiter (= schnelle Eingewöhnungsphase) ...)
- Systemspezifikation (Pflichtenheft) (z. B. soll der Ablauf eines Kaufvorgangs folgendermaßen aussehen: Händlerauswahl, Bestellung, Abrechnung ... Die Benutzeroberfläche soll folgende Informationen anzeigen: Lieferstatus,)

(ii) Systementwurf

- Systemarchitektur (z. B. Klassendiagramm, Festlegung von Komponenten ...)
- Schnittstellen der Komponenten festlegen (z. B. Datenbank, Middleware, ...)
- Modulkonzept (z. B. Modulstruktur und Art der Module (funktional, prozedural, objektorientiert))
- Definition der Einzelschnittstellen (z. B. Interfaces und abstrakte Klassen in Java)

(iii) Implementierung

- Strukturierung der Komponenten in Module (z. B. Klassenrumpfe)

- Spezifikation einzelner Module (z. B. exakte Beschreibung eines Moduls)
- Codierung, Generierung, Wiederverwendung einzelner Module (z. B. Methoden in Klassen, ...)

(iv) Integration und Test

- Integration von Modulen (z. B. Zusammensetzen von Datenbank und Applikation zum Server)
- Integration von Komponenten (z. B. Einsetzen von Client und Server ins Gesamtsystem.)
- Testen und Verifikation (z. B. Testen, ob die Implementierung zusammen mit der Umgebung funktioniert. Zusichern, dass gewisse Eigenschaften gültig sind.)

(v) Wartung und Weiterentwicklung

- Wartung (z. B. Korrektur eines Fehlers im Programm, Portierung auf anderes Betriebssystem, ...)
- Erweitern der Funktionalität (z. B. Einbeziehen einer neuen Option im Geschäftsprozess)
- Anpassung der Funktionalität (z. B. Anpassung an Änderung im Geschäftsprozess)

(b) Während des Entwicklungsprozesses werden nach und nach im Folgenden aufgelistete Teilprodukte entstehen. Ordnen Sie diese den jeweiligen Phasen zu.

- Schätzung der Entwicklungskosten
- Dokumentation der Änderungen nach Produktabnahme
- Pflichtenheft
- Beschreibung der Datenstruktur in der Datenbank
- Integration von Modulen/Komponenten
- Betriebsbereites Informationssystem
- Beschreibung der Schnittstelle einzelner Softwarekomponenten
- Quellcode für die GUI
- Durchführbarkeitsstudie
- Systemarchitektur

- Schätzung der Entwicklungskosten → *Anforderungsdefinition*
- Dokumentation der Änderungen nach Produktabnahme → *Betrieb und Wartung*
- Pflichtenheft → *Anforderungsdefinition*
- Beschreibung der Datenstruktur in der Datenbank → *System- und Softwareentwurf*
- Integration von Modulen/Komponenten → *Integration und Systemtest*

- Betriebsbereites Informationssystem → *Betrieb und Wartung*
- Beschreibung der Schnittstelle einzelner Softwarekomponenten
→ *System- und Softwareentwurf*
- Quellcode für die GUI → *Implementierung und Komponententest*
- Durchführbarkeitsstudie → *Anforderungsdefinition*
- Systemarchitektur → *System- und Softwareentwurf*

Vorgehensmodelle

Dem Team von Teacher-Data kommen Zweifel, ob die Entscheidung für das Wasserfallmodell für die Umsetzung des Informationssystems richtig war, oder ob ein anderes Vorgehen eventuell besser wäre. Beurteilen und vergleichen Sie die Vorgehensmodelle

- Wasserfallmodell
- Spiralmodell
- V-Modell
- Evolutionäres Modell oder Inkrementelles Modell
- agile Entwicklung, wie z. B. Scrum

nach den folgenden Gesichtspunkten:

- (a) Größe des Entwicklerteams
- (b) Komplexität des Projekts
- (c) Bekanntheit der Anforderungen
- (d) Änderung der Anforderungen
- (e) Zeitspielraum (Time-to-Market; was muss bis wann fertig sein?)
- (f) Dokumentation
- (g) IT-Kenntnisse des Kunden
- (h) Durchschnittliche Anzahl an Iterationen

	Wasserfallmodell	Spiralmodell	V-Modell	evolutionär / inkrementell	agil
Größe des Entwicklerteams	diskutiert die Projektgröße nicht	mittlere Teams	mittlere und große Teams	diskutiert die Projektgröße nicht	ca. 3-9 Entwickler (ohne Product Owner und Scrum Master)
Komplexität des Projekts	einfache Projekte mit stabilen Anforderungen	einfache Projekte mit stabilen Anforderungen; Anforderungen können jedoch in den Iterationen angepasst werden.	komplexe Projekte; bei einfachen Projekten relativ viel bürokratischer Overhead.	große, lange und komplexe Projekte	hohe Komplexität möglich
Bekanntheit der Anforderungen	Alle Anforderungen müssen zu Beginn des Projekts bekannt sein.	Es muss eine initiale Menge von Anforderung bekannt sein. Der Kunde kann die Anforderungen aber aufgrund von Prototypen erweitern.	evolutionär: Alle Anforderungen müssen zu Beginn des Projekts bekannt sein. inkrementell: Anforderungen müssen nicht von Anfang an bekannt sein.	Anforderungen müssen von Anfang an bekannt sein.	Es muss eine initiale Menge von Anforderung bekannt sein. Der Kunde kann die Anforderungen aber laufend erweitern / ggf. ändern.
Änderungen der Anforderungen	Schwer möglich. Durch den strikten Top-Down-Ansatz von der Anforderung hin zum Code zieht eine Änderung einen tiefen Eingriff in den Prozess mit sich.	Gut möglich. In jeder Runde der Spirale können die Anforderungen neu definiert werden.	Schwer möglich → strikter Top-Down-Ansatz	evolutionär: häufige Änderungen möglich. inkrementell: Änderungen teilweise möglich.	rasche Anpassung an neue Anforderungen möglich (deshalb agil!)
Zeitspielraum	Wenig Spielraum. Üblicherweise muss der ganze Prozess durchlaufen werden. Es ist höchstens möglich, den Prozess zu beschleunigen, indem man Anforderungen streicht. Es wird also alles zum Schluss fertig.	Relativ flexibel. Sobald ein Prototyp existiert, der akzeptabel ist, kann dieser auf den Markt gebracht werden.	Wenig Spielraum (vgl. Wasserfallmodell)	Einsatzfähige Produkte in kurzen Zeitabständen, daher relativ flexibel.	Unterteilung in Sprints, daher schnell lauffähige Prototypen vorhanden. Bei der Planung des nächsten Sprints kann auf neue zeitliche Gegebenheiten relativ flexibel eingegangen werden.
Dokumentation	Viel Dokumentation – Für jede Phase wird eine komplette Dokumentation erstellt.	Sehr viel Dokumentation – In jedem Zyklus werden alle Phasen (inkl. Dokumentation) durchlaufen, aber nur im Umfang des Prototypen. Es werden alle Artefakte und Tätigkeiten dokumentiert.	Viel Dokumentation – Gerade in den ersten Phasen wird sehr viel über die Software schriftlich festgehalten. Aber auch für die anderen Phasen wird eine komplette Dokumentation erstellt.		Anforderungsdokumentation ist sehr wichtig. Ansonsten ist funktionierende Software höher zu bewerten als eine umfangreiche Dokumentation.
IT-Kenntnisse des Kunden	Wenig Kenntnisse nötig. Meistens schreibt der Kunde das Lastenheft und der Entwickler versucht dieses dann in einem Pflichtenheft umzusetzen. Wenn das Lastenheft vom Entwickler als machbar befunden wird, bekommt der Kunde das fertige Produkt.	Wenig Kenntnisse nötig. – Kunde sieht immer nur die fertigen Prototypen und äußert dann seine Wünsche.	Wenig Kenntnisse nötig (vgl. Wasserfallmodell)	Wenig Kenntnisse nötig	Wenig Kenntnisse nötig, aber von Vorteil, da enge Zusammenarbeit mit dem Kunden
Durchschnittliche Anzahl an Iterationen	vgl. V-Modell	ca. 3-5 Iterationen	Nur lange Zyklen. – Mit dem V-Modell sind nur sehr lange Zyklen handhabbar, da immer wieder der ganze Entwicklungsprozess durchlaufen werden muss.		variable Anzahl an Sprints, je nach Projektgröße