



# Anwendungsbeispiel: Analyse



# Inhalt

- Requirements Engineering Prozess
- Use-Case Modellierung
- Domänenmodell
- ERM Diagramme



# Anwendungsbeispiel: Computerspieleplattform



















## Computerspieleplattform „DAMPF“

Die Firma VAULT möchte eine Internet-Vertriebsplattform für Computerspiele erstellen bei denen Hersteller ihre Computerspiele anbieten können. Benutzer können sich die Plattform herunterladen und installieren und bequem die angebotenen Spiele durchsuchen, filtern und direkt kaufen. Gekaufte Spiele werden in einer persönlichen Bibliothek gespeichert auf die der Benutzer jederzeit Zugriff hat. Spielehersteller wie auch Benutzer müssen einen Account für die Plattform besitzen. Wenn ein Spiel von einem Benutzer über die Plattform erworben wird bekommt VAULT einen gewissen Prozentsatz des Umsatzes. Für die Bezahlung sollen verschiedene Methoden angeboten werden. Zudem soll es möglich sein das ein Benutzer ein erworbenes Spiel zurückgeben kann, wenn er dieses weniger als 10h gespielt hat. Auch werden Aktualisierungen für die Spiele ("Updates") direkt von den Herstellern der Spiele über das System verteilt.

# Anwendungsbeispiel: Computerspieleplattform

## Computerspieleplattform „DAMPF“

Die Firma VAULT möchte eine Internet-Vertriebsplattform für Computerspiele erstellen bei denen Hersteller ihre Computerspiele anbieten können. Benutzer können sich die Plattform herunterladen und installieren und bequem die **angebotenen Spiele durchsuchen, filtern und direkt kaufen**. Gekaufte Spiele werden in einer persönlichen Bibliothek gespeichert auf die der Benutzer jederzeit Zugriff hat. Spielehersteller wie auch Benutzer müssen einen Account für die Plattform besitzen. Wenn ein Spiel von einem Benutzer über die Plattform erworben wird bekommt **VAULT** einen gewissen **Prozentsatz des Umsatzes**. Für die Bezahlung sollen verschiedene Methoden angeboten werden. Zudem soll es möglich sein das ein Benutzer ein erworbenes **Spiel zurückgeben** kann, wenn er dieses weniger als 10h gespielt hat. Auch werden Aktualisierungen für die Spiele ("Updates") direkt von den Herstellern der Spiele über das System verteilt.

← → STORE LIBRARY						
Q Search		STEAMOS + LINUX		VIEW		
GAMES		★	☁	STATUS	METASCORE	LAST PLAYED
	Amnesia: The Dark Descent			Not installed	85	New 08.06.2012
	Antichamber			Not installed	82	23.02.2014
	Aquaria			Not installed	82	New 29.09.2011
	Audiosurf 2		☁	Not installed	76	01.12.2014
	The Banner Saga		☁	Not installed	80	15.12.2015
	Bastion		☁	Not installed	86	23.06.2012
	BioShock Infinite		☁	Not installed	94	07.12.2014
	Borderlands 2		☁	Not installed	89	25.11.2014
	Braid		☁	Not installed	90	02.02.2014
	Broforce		☁	Not installed	83	26.03.2017
	Chaos on Deponia		☁	Not installed	78	17.10.2015
	Counter-Strike			Not installed	88	02.01.1970
	Counter-Strike: Global Offensive		☁	Not installed	83	25.06.2016
	Counter-Strike: Source		☁	Not installed	88	02.10.2014
	Darkest Dungeon		☁	Not installed	84	31.10.2017
	Day of Defeat			Not installed	79	02.01.1970
	Dead Island		☁	Not installed	80	New 01.08.2013
	Deathmatch Classic			Not installed		New 26.11.2003
	Deponia		☁	Not installed	74	08.10.2015

# Requirements Engineering Prozess

Requirements Engineering						
Requirements Analysis					Requirements Management	
Elicitation	Interpretation	Negotiation	Documentation	Validation	Change Management	Tracing
<ul style="list-style-type: none"> <li>➤ Identification of stakeholders and other sources of requirements (e.g., old existing systems)</li> <li>➤ Gathering of raw requirements</li> <li>➤ Target analysis</li> </ul>	<ul style="list-style-type: none"> <li>➤ Identification of requirements</li> <li>➤ Structuring of requirements (merging, grouping, classification)</li> <li>➤ Refinement (to satisfiable criteria)</li> </ul>	<ul style="list-style-type: none"> <li>➤ Identification of dependencies</li> <li>➤ Identification of inconsistencies</li> <li>➤ Resolution of inconsistencies</li> <li>➤ Prioritization</li> </ul>	<ul style="list-style-type: none"> <li>➤ Specification of requirements (incl. intermediate results and assumptions)</li> </ul>	<ul style="list-style-type: none"> <li>➤ Contentual validation (correctness, completeness, consistency)</li> <li>➤ Formal verification</li> </ul>	<ul style="list-style-type: none"> <li>➤ Handling of change requests</li> <li>➤ Management of different versions of the requirements (history)</li> <li>➤ Propagation of changes</li> </ul>	<ul style="list-style-type: none"> <li>➤ Recording of assumptions and requirement sources</li> <li>➤ Recording of how requirements are implemented</li> </ul>



# Requirements Elicitation (1)

- Identifizierte Stakeholder
  - Spielehersteller, Benutzer, VAULT, Tester, „Hacker“, Infrastruktur-Ingenieur, ...
- Priorisieren der Stakeholder
- Identifizierte zusätzliche Quellen
  - GOG Galaxy, Epic Games Launcher, ...

# Requirements Elicitation (2)

- Identifizierte Requirements (u.a.)
  - Req-1: Benutzern sollte es möglich sein Spiele zu erwerben
  - Req-2: Der Abschluss des Kaufes eines Spiels sollte in weniger als 5 Sekunden abgeschlossen sein
  - Req-3: Das System soll eine Web Oberfläche und ein GUI bieten





# Requirements Interpretation (1)

Req-1: Benutzern sollte es möglich sein Spiele zu erwerben

Req-2: Der Abschluss des Kaufes eines Spiels sollte in weniger als 5 Sekunden abgeschlossen sein

Req-3: Das System soll eine Web Oberfläche und ein GUI bieten

## ■ Identifizierte Beziehungen

- Req-1 wird für Req-2 benötigt

- Req-3 muss separiert werden

  - Req-3: Das System soll eine Web Oberfläche besitzen

  - Req-4: Das System soll eine Desktop GUI besitzen

# Requirements Interpretation (2)

Req-1: Benutzern sollte es möglich sein Spiele zu erwerben

Req-2: Der Abschluss des Kaufes eines Spiels sollte in weniger als 5 Sekunden abgeschlossen sein

Req-3: Das System soll eine Web Oberfläche besitzen

Req-4: Das System soll eine Desktop GUI besitzen

## ■ Klassifikation der Requirements

- ☐ Req-1: Funktional
- ☐ Req-2: Nicht funktional
- ☐ Req-3: Funktional
- ☐ Req-4: Funktional

# Requirements Interpretation (3)

Req-1: Benutzern sollte es möglich sein Spiele zu erwerben

Req-2: Der Abschluss des Kaufes eines Spiels sollte in weniger als 5 Sekunden abgeschlossen sein

Req-3: Das System soll eine Web Oberfläche besitzen

Req-4: Das System soll eine Desktop GUI besitzen

## ■ Gruppierung der Requirements

- Spieleerwerb: Req-1, Req-2
- User Interface: Req-3, Req-4

# Requirements Interpretation (4)

Req-2: Der Abschluss des Kaufes eines Spiels sollte in weniger als 5 Sekunden abgeschlossen sein

- Konkretisierung der Requirements mit Kunden

- Req-2: Was heißt „Abschluss des Kaufes“?

- Der gesamte Prozess von Auswahl bis zur Eintragung in die Spielebibliothek?
    - Der letzte Schritt vom Drücken eines „Kaufen“ Buttons bis in die Eintragung in die Bibliothek?

- Glossar



# Requirements Negotiation

- Bei den Verhandlungen sind keine großen Streitpunkte aufgetreten
- Identifizierte Bedingungen
  - Req-1 wird für Req-2 benötigt
- Keine Inkonsistenten in den Requirements vorhanden



# Requirements Negotiation

Req-1: Benutzern sollte es möglich sein Spiele zu erwerben

Req-2: Der Abschluss des Kaufes eines Spiels sollte in weniger als 5 Sekunden abgeschlossen sein

Req-3: Das System soll eine Web Oberfläche besitzen

Req-4: Das System soll eine Desktop GUI besitzen

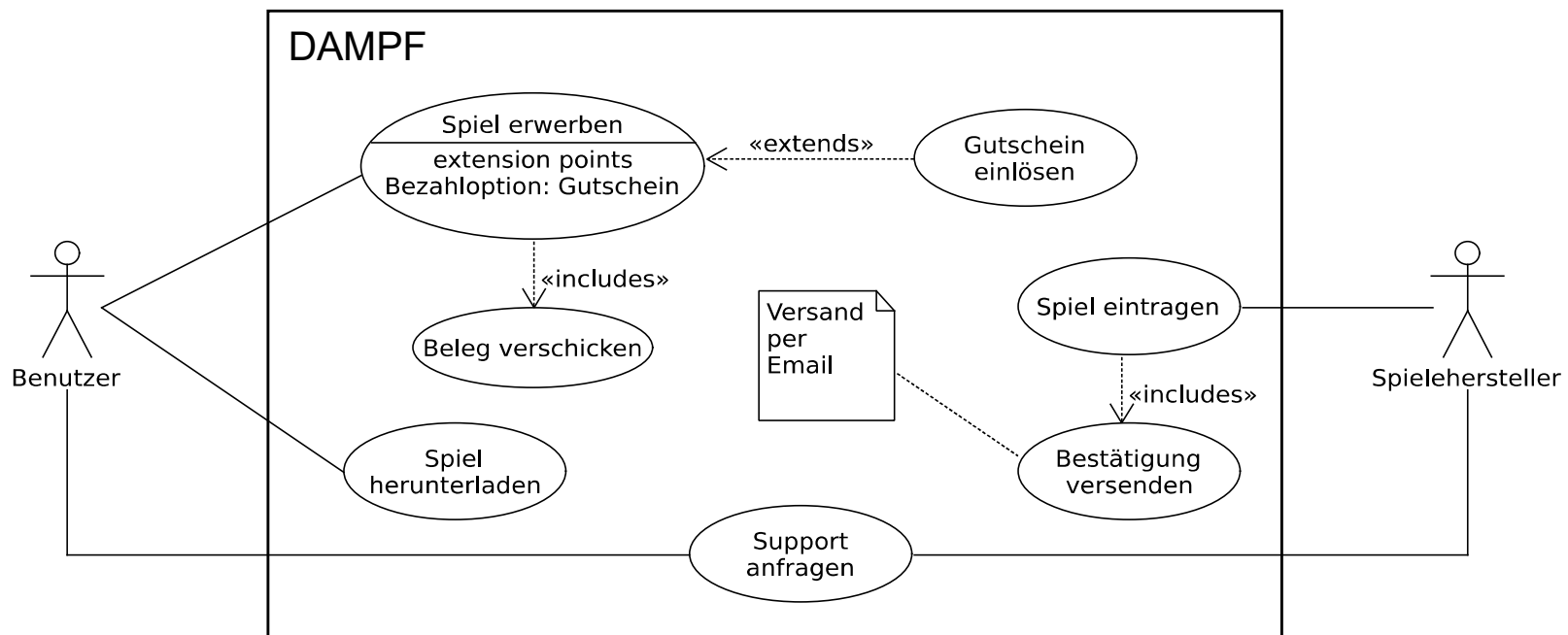
## ■ Priorisierung nach Kundenabsprache

- ☐ Req-1, Req-4: Must-have

- ☐ Req-2, Req-3: Optional

# Requirements Dokumentation (1)

- Es wurde Entschieden einen Mix aus natürlicher Sprache und UML-Anwendungsfalldiagrammen zu benutzen



# Requirements Dokumentation

## (2)

**Anwendungsfall:**

Spiel herunterladen

**Zusammenfassung:**

Prozess eines Benutzers um ein Spiel herunterzuladen. Hierfür sucht sich der Benutzer ein noch nicht heruntergeladenes Spiel aus seiner Bibliothek für erworbene Spiele aus.

**Akteure:**

Benutzer

**Auslöser:**

Benutzer möchte ein erworbenes Spiel herunterladen.

**Vorbedingungen:**

Spiel muss in der Bibliothek für erworbene Spiele vorhanden sein.



# Requirements Dokumentation

## (3)

### **Standardablauf:**

1. Benutzer startet DAMPF.
2. Benutzer loggt sich ein.
3. Benutzer navigiert zu seiner Bibliothek.
4. Benutzer wählt zu heruntergeladenes Spiel aus.
5. Benutzer betätigt herunterladen Button.
  - 5.1. Benutzer gibt Download-Verzeichnis an.
  - 5.2. Benutzer bestätigt Start des Downloads.

### **Nachbedingungen:**

Spiel wird heruntergeladen.

### **Ausnahmen und Varianten:**

Zu 5.1: Download-Verzeichnis bietet nicht genügend Speicherplatz: Wdh. Schritt 5.1.



# Requirements Dokumentation

## (4)

### **Ergebnis:**

Spiel ist heruntergeladen.

### **Häufigkeit:**

5 mal pro Monat.

### **Verweise:**

Glossar


### **Anmerkungen:**

Download kann jederzeit abgebrochen werden.



# Requirements Validation

- Alle erstellten Dokumente haben einen dedizierten Reviewprozess durchlaufen
- Es wurde überprüft ob die Wünsche der Kunden berücksichtigt wurden



# Requirements Change Management and Tracing



# Change Management

- Verwalten von angefragten Änderungen
- Sollte systematisch gehandhabt werden
- Bspw. durch Workflow der den Ablauf einer Änderungsanfrage durchläuft
- Vielzahl von existierenden Tools
  - Erlaubt erstellen von Change Request Workflows
  - Bspw: Jira

# Tracing

- Nach/Rückverfolgung von Anforderungen
- Zeigt abdeckung der Anforderungen
- Verschieden Methoden:
  - Bspw: Traceability Matrix

Req No.	Beschreibung	Testcase	Status
1	Benutzer sollen Spiele erwerben	TC3,TC5	TC3 pass, TC5 fail
2	Abschluss des Spielekaufs in weniger als 5 Sekunden	TC2, TC4	TC2 no run, TC4 pass
3	Das System soll eine GUI bieten	TC1	TC1 pass,