

Lutz Prechelt

# Softwaretechnik, SoSe21

## Übung 05

TutorIn: Samuel Domiks

Tutorium 02

Materialien: Latex, Skript

Jonny Lam & Thore Brehmer

16. Mai 2021

---

## 1 Aufgabe: Anwendungsfälle

a) *Was ist ein Anwendungsfall (Use Case) im Kontext der Softwareentwicklung (unabhängig von der UML) und wozu dient er?*

- Ein Use Case (Anwendungsfall) ist eine Vereinbarung (wenn die Vereinbarung fertig formuliert ist) zwischen den Beteiligten eines Projektes über das Verhalten des Systems (was das System tun soll, funktionale Anforderung) [1].
- Dabei beschreibt das Use Case, wie sich das System Verhalten soll, wenn ein Hauptakteur (Akteure nehmen eine bestimmte Rolle bzw. Perspektive ein, die zum Use Case gehört [2]) ein bestimmtes Ziel erreichen will [3].
- Im Rahmen des Use Case wird dieses Ziel erreicht, insoweit, dass darüber steht was erreicht werden soll und am Ende des Use Case ist es erreicht worden, es sei denn da ging was schief.
- Use Cases sind Bündelung von Fällen (Szenarien), des gleichen Ziels [4].

b) *Recherchieren Sie Prinzipien zum Verfassen guter Anwendungsfälle. Verwenden Sie mindestens zwei verschiedene Quellen (diese sind, wie immer, anzugeben).*

1. *Nehmen Sie dabei zuerst die „Produktsicht“ ein: Was macht einen guten Anwendungsfall eigentlich aus? Formulieren Sie mindestens drei Kriterien, die spezifisch für Anwendungsfälle sind (Kriterien wie „verständlich“ oder „vollständig“ zählen also nicht).*

“Ultimately, use cases are about clearly communicating detailed information to a very diverse audience and reaching the goal of creating successful development projects.” [16]

1. Ein guter Anwendungsfall soll zielführend sein, d.h. dass am Ende des Anwendungsfalls das Ziel erreicht sein sollte [5].
2. Ein guter Anwendungsfall sollte sich mit der Absicht des Akteurs beschäftigen und nicht mit der Benutzeroberfläche [5,6].
3. Ein guter Anwendungsfall sollte gut gegliedert sein, d.h. dass da am Anfang die oberflächige Information stehen sollte wie Akteur und Ziel und bis zum Ende hin mehr darauf eingegangen wird [5].

4. Ein guter Anwendungsfall sollte nicht zu lang sein (ist es zu lang liest sich das keiner mehr durch "gerade genau genug" [8]), falls der doch zu lang ist, kann es sein, dass man Abschnitte davon zu einem eigenen Use Case machen könnte. Dann könnte man darauf verweisen, wie z.B. „Siehe Use Case UC006“ [6].
5. "A use case describes functionality but it also must describe a result" [10]
2. *Wechseln Sie nun zur „Prozesssicht“: Wie schreibt man denn nun einen solchen guten Anwendungsfall? Suchen Sie nach Leitlinien zum Schreibprozess und erklären Sie mindestens die Kernaussage mindestens drei dieser Leitlinien mit eigenen Worten.*
  - Hier sollte man auch darauf achten, dass man den Use Case von niedriger Präzision zu höherer Präzision bearbeitet. Das bedeutet, dass am Anfang erstmal die Hauptinformationen stehen und danach näher darauf eingegangen wird [5]. Nach dem Schema:
    - Hauptakteur und sein Ziel (Sehr wichtig, so weiß man, wo man überhaupt hinwill)
    - Haupterfolgsszenario (Ist (meist) der einfachste Weg wie man das Ziel erreicht).
    - Erweiterungsbedingungen (Hier werden auf Probleme und Alternativen aufmerksam gemacht, die beim Haupterfolgsszenario vorkommen könnten)
    - Erweiterungsschritte (Hier werden die Alternativen und die Probleme behandelt, die im Hauptszenario entstehen könnten)
  - Leitlinie zum Schreibprozess
    1. Erstmal sollte man sich den Systemumfang klar machen, in dem man den Anwendungsfall aufstellen will. [5]
    2. Dann sollte man sich mithilfe von Brainstorming eine Übersicht der Hauptakteure machen und so deren Ziele herausfinden. [5]
    3. Hat man die Ziele so kann man ein Use Case erstellen und dessen Hauptszenario, wobei man bei 3-9 Schritten bleiben sollte [5] (Hier will man sichergehen, dass es nicht zu umfangreich wird, da Beteiligte nur an Informationen bzw. Überblicken interessiert sind [8]).
    4. Jetzt sollte man sich überlegen, ob man den Anwendungsfall erweitern kann [5] (wenn was nicht funktioniert oder wenn es einen alternativen Weg gibt), z.B. was passiert, wenn man kein Internet hat (bei einem login use case) oder wenn man sich über ein anderes Programm o.Ä. anmelden kann.
    5. Den Anwendungsfall kann man nun dementsprechend erweitern.
    6. Um das ganze übersichtlicher zu gestalten sollte man noch drüber schauen, ob man bestimmte Schritte zusammenfassen kann oder zu einem eigenen Anwendungsfall extrahieren kann [5].
    7. Am Ende kann man noch was hinzufügen, entfernen oder zusammenfassen, wenn nötig [5].
    8. Die Mehrheit der Akteure sollten Menschen sein (keine Systeme), weil "people are the most important use case element" [11]
    9. Wenn Sie (mit Ihrem Team) einen Anwendungsfallstil festlegen, können Sie diese schneller und einfacher schreiben und verstehen, da "Good use cases are not just paragraphs of clear and concise text. They have a structure that defines how the parts of the use case fit together"[11]
    10. A use case should have a single main flow and multiple alternative flows, as "Sometimes use case writers try to have multiple main flows, but this can dilute the understandability of the use case because it makes the reader work harder to understand the true goal" [12]

11. Vermeide if Klauseln im Anwendungsfall, da “Problems arise when there are one or more if statements in a flow, because it usually means that you are specifying multiple requirements” [13]
12. Erzwingen Sie keinen Ereignisfluss, wenn eigentlich gar keiner vorhanden ist. “The point is, you don’t want to constrain the designers of the system by having implicit temporal requirements when they are not necessary” [14]
13. Vergessen Sie nicht, dass Sie Anwendungsfälle für den Endbenutzer schreiben. “If use cases are requirements (and they are), the designers and developers will be constrained by what they say. Remember this, take pity on the actor and write your use cases in a way that makes the system as easy as possible for the user.” [15]

c) *Worin besteht der Unterschied zwischen einem Szenario und einem Anwendungsfall?*

- Szenarien können viele verschiedene Fälle sein, die zu einem Anwendungsfall gehören. Zum Beispiel gibt es zu einem Anwendungsfall meist ein Hauptszenario, wo alle Schritte stehen, um das Ziel zu erreichen. Dann kann es passieren, dass ein Schritt abweicht bzw. ein Problem auftritt, da kommen andere alternativen Szenarien im Spiel, die zum Ziel führen können bzw. die auch zum Misserfolg führen oder auch helfen zum Hauptszenario zurückzukommen. [9]

## Quellen

- [1] Vorlesung 6; Seite 4; 1. Abschnitt
- [2] Vorlesung 6; Seite 14; 1. Abschnitt
- [3] Vorlesung 6; Seite 4; 2. Abschnitt
- [4] Vorlesung 6; Seite 4; 3. Abschnitt
- [5] A. Cockburn, „Use case guidelines” <http://faculty.washington.edu/jtenenbg/courses/360/f02/project/usecaseguidelines.html> . [aufgerufen am 14.05.2021]
- [6] <http://www.sysflow.com/blog/5-rules-for-writing-effective-use-cases/> [aufgerufen am 14.05.2021]
- [7] A. Cockburn, „Writing Effective Use Cases” <https://www.infor.uva.es/~mlaguna/is1/materiales/BookDraft1.pdf> [aufgerufen am 14.05.2021]
- [8] Vorlesung 6; Seite 2
- [9] <https://de.wikipedia.org/wiki/Anwendungsfall> [aufgerufen am 14.05.2021]
- [10] “Tips for writing good use cases”, <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/RAW14023-USEN-00.pdf> [aufgerufen am 16.05.2021], pp.4
- [11] “Tips for writing good use cases”, <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/RAW14023-USEN-00.pdf> [aufgerufen am 16.05.2021], pp.6
- [12] “Tips for writing good use cases”, <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/RAW14023-USEN-00.pdf> [aufgerufen am 16.05.2021], pp.7
- [13] “Tips for writing good use cases”, <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/RAW14023-USEN-00.pdf> [aufgerufen am 16.05.2021], pp.10
- [14] “Tips for writing good use cases”, <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/RAW14023-USEN-00.pdf> [aufgerufen am 16.05.2021], pp.13
- [15] “Tips for writing good use cases”, <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/RAW14023-USEN-00.pdf> [aufgerufen am 16.05.2021], pp.14
- [16] “Tips for writing good use cases”, <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/RAW14023-USEN-00.pdf> [aufgerufen am 16.05.2021], pp.2

## 2 Aufgabe: UML-Anwendungsfalldiagramme

- a) *Was ist ein Anwendungsfalldiagramm im Gegensatz zum Anwendungsfall? Worin bestehen Vor- und Nachteile bei der Darstellung mit UML-Anwendungsfalldiagrammen? Nennen Sie jeweils mindestens zwei.*

Ein Anwendungsfalldiagramm visualisiert die Zusammenhänge und Beziehungen zwischen einer Menge von Use Cases und den involvierten Akteuren. [3] Sie beschreiben die Funktionalität, die zu erbringenden Dienste und Leistungen, aus Anwendersicht. [8] Damit eignet es sich sehr gut zur Anforderungsanalyse, also zur Ermittlung oder Verfeinerung von Anforderungen und bieten einen sehr guten Überblick über das Gesamtsystem [3][8]

- **Vorteile**

1. Generell nützlich für Kontext Diagramme und table of contents. [2]
2. Bietet eine gute Zusammenfassung über das gesamte System/Software in einem einzelnen Diagramm. [1]
3. Bietet einen zusätzlichen Blickwinkel auf das System und auf Zusammenhänge zwischen einzelnen Anwendungsfällen. [3]
4. Hilft bei der Strukturierung eines zu entwickelnden Systems und bei der Festlegung der Systemgrenzen. [3]
5. Gibt eine Übersicht, welche Akteure an welchen Use Cases Teilnehmen. [5]

- **Nachteile**

1. Bietet nur wenig Informationen. [5]
2. Es gibt eine Lernkurve bzw. es ist anfangs Unverständlich für Klienten und Entwickler [1]
3. Bei größeren Systemen kann es schnell unübersichtlich werden. (Eigen Annahme)

- b) *Was repräsentiert das Modellelement Akteurin einem Anwendungsfalldiagramm?*

- Akteure sind Personen oder Systeme außerhalb des beschriebenen Systems, mit welchem sie interagieren (z.B. durch Austausch von Signalen oder Daten). [6][7] Sie haben Anforderungen an das oder ein Interesse am System und sind entsprechend an den Ergebnissen interessiert.[8] Im Normalfall werden sie als Strichmännchen dargestellt. [3]

- c) *Welche weiteren Modell-Elemente kennt die UML für Anwendungsfalldiagramme neben dem Akteur und dem Use Case selbst? Nennen und erläutern Sie drei.*

1. **System (System Boundary).** “Das System ist an sich kein logisches Modellelement, sondern grenzt den Kontext ab.” [3] Eine Systemgrenze eines Anwendungsfalldiagramms definiert also die Grenzen Umfang des Systems. Die Systemgrenze wird als Rechteck angezeigt, das alle Anwendungsfälle im System umfasst. [9]
2. **Beziehungen.** Beziehungen(Assoziation) zwischen Akteuren und Anwendungsfällen werden normalerweise mit einer Linien modelliert. Wenn eine solche Beziehung existiert, heißt das, dass der Akteur den Anwendungsfall ausführt bzw. mit diesem kommuniziert. [8]
3. **Beschreibungen und Notizen.** Notizen können an Akteure und Anwendungsfällen mithilfe einer geschrichelten Linie verbunden werden. Mithilfe der Notiz kann man dem Element weitere Informationen geben bzw. dieses Beschreiben. [3] [8]

- d) *Welche Arten von Beziehung können zwei Use Cases in einem UML-Anwendungsfalldiagramm zueinander haben? Nennen und vergleichen Sie mindestens zwei.*

1. **Enthält-Beziehung (Include).** “Teile von Anwendungsfällen, die in mehreren Use Cases in identischer Weise vorkommen, können in einem eigenen Anwendungsfall ausgelagert und per Enthält-Beziehung wieder eingebunden werden, um so eine redundante Beschreibung der identischen Teile zu vermeiden.” [8] Sie wird durch einen gestrichelten Pfeil mit offener Pfeilspitze dargestellt.
2. **Erweiterungsbeziehung (Extend).** In einer Erweiterungsbeziehung zwischen zwei Anwendungsfällen erweitert das Kind den Parent um Funktionalitäten. Sie wird durch einen gestrichelten Pfeil dargestellt. [3][9]
3. **Spezialisierung (Generalisierung).** Eine Generalisierung kann zwischen Akteuren oder zwischen Anwendungsfällen vorkommen. Sie wird durch einen gerichteten Pfeil mit Dreieck Spitze dargestellt. In einer Generalisierung, kann man das Child als Spezialisierung vom Parent beschreiben. Es ist wie eine “IS-A” Beziehung in ER-Diagramm. [3][8][9]

#### Quelle:

- [1] “Advantages and disadvantages of USE CASE diagram ” , <https://mansansaar.blogspot.com/2015/08/advantages-and-disadvantages-of-use.html>, [Accessed 16.05.2021]
- [2] “Alistair Cockburn Writing Effective Use Cases”, [https://www.academia.edu/32227372/Alistair\\_Cockburn\\_Writing\\_Effective\\_Use\\_Cases](https://www.academia.edu/32227372/Alistair_Cockburn_Writing_Effective_Use_Cases), [Accessed 16.05.2021], pp. 245
- [3] “Use Case Diagramm”, <https://t2informatik.de/wissen-kompakt/use-case-diagramm/>, [Accessed 16.05.2021]
- [4] “Disadvantages of use cases and scenarios”, <https://uxapprentice.wordpress.com/2011/11/29/disadvantages-of-use-cases-and-scenarios/>, [Accessed 16.05.2021]
- [5] “Vorlesung 6”, [Accessed 16.05.2021], pp. 40
- [6] “Aufbau eines Anwendungsfalls”, [https://de.wikipedia.org/wiki/Anwendungsfall#Aufbau\\_eines\\_Anwendungsfalls](https://de.wikipedia.org/wiki/Anwendungsfall#Aufbau_eines_Anwendungsfalls), [Accessed 16.05.2021]
- [7] “Use Case Analysis: How to Identify Actors?”, <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/how-to-identify-actors/>, [Accessed 16.05.2021]
- [8] “Anwendungsfalldiagramm (Use Case Diagram)”, <https://www.sparxsystems.de/ressourcen/literatur/leseprobe-zu-projektabwicklung-mit-uml-und-enterprise-architect/anwendungsfalldiagramm>, [Accessed 16.05.2021]
- [9] “Elements of a Use Case Diagram”, [https://www.e-education.psu.edu/geog468/18\\_p4.html](https://www.e-education.psu.edu/geog468/18_p4.html), [Accessed 16.05.2021]

### 3 Aufgabe: Anwendungsfall und UML-Diagramm entwickeln

- a) Formulieren Sie basierend auf einer Ihrer Anforderungen (Übungsaufgabe4-2) einen Anwendungsfall für Ihre Software aus. Wählen Sie keinen trivialen Anwendungsfall, d.h. sein Erfolgsszenario sollte mehr als drei Schritte beinhalten und er sollte mindestens eine Erweiterung besitzen.

Anwendungsfall	12. User_Searchs_Receipt
Beschreibung	12. Benutzer sucht eine Quittung.
Akteure	Benutzer und App
Annahmen	Quittung ist in der App vorhanden.
Schritte	1. Benutzer öffnet App. 2. Benutzer klickt auf den Button zum Suchen einer Quittung. 3. App öffnet ein Dialog mit Filter für die Suche. 4. Benutzer trägt benötigte Daten ein: Datum/Zeitraum, Kategorie, Markt 5. Benutzer bestätigt seine Eingaben. 6. App schließt Dialog und zeigt alle Quittungen mit diesem Filter an.
Variationen	4. Benutzer kann Daten weglassen 5. Benutzer bestätigt nicht und schließt den Dialog, alles bleibt so wie es ist.
Nicht-funktionales	Keine
Probleme	Falls Quittungen online gespeichert sind und kein Internetzugang besteht, werden die Quittungen nicht angezeigt.

Erweitern tun wir das, in dem wir die angezeigten Quittungen noch Sortieren. Es ist sehr wahrscheinlich, dass wir auch mit dem Filter sehr viele Quittungen haben, und deshalb ist es wichtig diese zu Sortieren.

Anwendungsfall	11. User_Sort_Receipt
Beschreibung	12. Benutzer kann seine Quittungen sortieren.
Akteure	Benutzer und App
Annahmen	Quittung ist in der App vorhanden.
Schritte	1. Benutzer klickt auf den Button zum Sortieren der Quittungen. 2. App öffnet ein Dialog mit Sortierungsmöglichkeiten. 3. Benutzer wählt Sortierungswunsch aus. 4. Benutzer bestätigt seine Eingaben. 5. App schließt Dialog und zeigt Quittungen mit dieser Sortierung an.
Variationen	3 Benutzer wählt nichts aus und alles bleibt so wie es ist. 4 Benutzer bestätigt nicht und schließt den Dialog, alles bleibt so wie es ist.
Nicht-funktionales	Keine
Probleme	Quittungen werden nicht alphabetisch sortiert, wenn man z.B. nach Datum aufsteigend sucht und beide gleichzeitig erstellt wurden, ist die Reihenfolge zufällig

- b) Erstellen Sie ein UML-Anwendungsfalldiagramm, das die Zusammenhänge von mindestens drei Anwendungsfällen Ihrer zu entwickelnden Software darstellt. (Diese Anwendungsfälle brauchen Sie nicht auszuformulieren.)

