Anforderungsanalyse (Requirement Engineering)

* „Verstehen und beschreiben, was die Kunden wünschen oder brauchen.“[[1]](#footnote-1)
* Identifizierung der Aufgabe eines Software-Systems und seinem Nutzungskontext
* Beschreibung der Fähigkeiten der Software sowie der Bedingungen, die die Software erfüllen muss.

Die Anforderungen werden gegliedert in die ***funktionalen Anforderungen*** und die ***nicht-funktionalen Anforderungen***, beispielsweise gemäß der FURPS+-Technik:

* **Functiona**l:   
  Angemessenheit, Sicherheit, Interoperabilität, Konformität, Ordnungsmäßigkeit,   
  Features, Fähigkeiten ==> mündet in Use-Cases (siehe unten)
* **Usability** / Benutzerfreundlichkeit: Human factors, Dokumentation   
  (Attraktivität, Bedienbarkeit, Erlernbarkeit, Konformität, Verständlichkeit)
* **Reliability** / Zuverlässigkeit:   
  Häufigkeit von Fehlern, Fehlererholung, Stabilität, Testbarkeit, Reife, Wiederherstellbarkeit
* **Performance** / Effizienz:   
  Antwortzeiten, Durchsatz, Verfügbarkeit, Kosten, Verbrauchsverhalten
* **Supportability** / Wartbarkeit:   
  Analysierbarkeit, Modifizierbarkeit, Internationalisierbarkeit
* „+“ steht für weitere Anforderungen wie: Implementation (Endliche Resourcen, Sprachen und Tools, Hardware), Interface (Schnittstellen für andere Systeme), Operations (Systemmanagement), Packaging (Strukturierung), Legal (Lizenzen usw.)

Diese Anforderungen Beschreiben die Qualität einer Software, wie sie Beispielsweise in der ISO/IEC 25000 (vormals ISO/IEC 9126) beschrieben wird.

Use-Cases (Anwendungsfälle)

Ein Anwendungsfall beschreibt jeweils eine vollständige und begrenzte Aufgabe des *Systems*, die mit einer Aktion eines *Aktors* beginnt und mit Beendigung dieser Aufgabe in Form eines messbaren Ergebnisses endet. Anwendungsfälle dürfen nicht zu klein (Einzelschritte) und nicht zu groß (Gesamtsystem) gefasst werden.

*Ein Anwendungsfall ist eine Menge von verhaltensverwandten Sequenzen von Transaktionen, die durch ein System ausgeführt werden und ein messbares Ergebnis liefern. (Jacobson)*

* UCs beschreiben nur **funktionale Anforderungen** an ein System
* UC werden immer aus Benutzerperspektive geschrieben
* UCs dienen zur Einigung mit dem Auftraggeber auf präzisen Leistungsumfang
* UCs erhöhen die spätere Anwenderakzeptanz
* UC gehen nicht auf Details des Systemverhaltens ein

Aktoren

Aktoren sind Benutzer, andere Systeme oder Geräte, die von außen Informationen mit dem System austauschen. Aktoren werden durch die Rolle, die sie gegen über dem System einnehmen, charakterisiert

Ein Szenario...

* ist die erzählende Beschreibung dessen, was Leute beim Versuch, Rechnersysteme und deren Anwendungen zu nutzen, tun und erleben.
* ist konkret, fokussiert, informell
* beschreibt ein einzelnes Systemmerkmal
* beschreibt aus Sicht eines einzelnen Akteurs
* enthält keine Verallgemeinerungen, keine Fallunterscheidungen (dafür werden mehrere Szenarien benötigt!)
* ist für einzelne Nutzer / Kunden verständlich formuliert[[2]](#footnote-2)

Fragen zur Erstellung von Anwendungsfällen:

1. Wer sind die beteiligten Akteure?
   * Welche Benutzergruppen werden vom System bei der Ausübung ihrer Arbeit unterstützt?
   * • Welche Benutzergruppen nutzen die Hauptfunktionen des Systems?
   * • Welche Benutzergruppen führen Sekundärfunktionen wie Wartung und Verwaltung durch?
   * Mit welcher externen Hardware oder Software wird das System interagieren?
   * Zu welchen Rollen können diese Akteure zusammengefasst werden?
2. Welche Szenarien / Transaktionen führen die Akteure mit dem System aus?
   * Zusammenfassung der einzelnen Szenarien zu Anwendungsfällen
3. Identifikation von Anwendungsfällen (use cases)
   * Diese bestimmen die Systemfunktionalität

Tabellarische Austellung eines Anwendungsfalls (Beispiel: Geldautomat)

|  |  |
| --- | --- |
| Name | Geld am Geldautomaten abheben |
| Kurzbeschreibung  (in 1 – 2 Sätzen) | Ein Benutzer hebt am Geldautomaten mit Hilfe seiner  Kreditkarte Geld von seinem Konto ab. |
| Hauptakteur | Benutzer (Bankkunde) |
| Nebenakteure | Konsortium (Bank) |
| Auslöser  (Warum führt der Akteur den Use-Case aus) | Benutzer aktiviert den Bankautomaten. |
| Vorbedingungen  (für eine erfolgreiche Ausführung) | Der Geldautomat ist bereit für einen Benutzer eine Transaktion durchzuführen. |
| Erfolgszustand  (Zustand nach erfolgreicher Ausführung, auch: *„Nachbedingung“)* | Der Benutzer hat die Kreditkarte, das Geld und den Beleg entnommen. Der Geldautomat ist erneut bereit eine Transaktion durchzuführen. |
| Fehlerzustand | – |
| Hauptszenario  (Schritte bzw. Interaktionen, die im Normalfall bei Ausführung des Anwendungsfalls durchlaufen werden) | 1. Der Benutzer gibt seine Kreditkarte ein.  2. Der Geldautomat liest die Kreditkarte und fordert daraufhin die Geheimzahl an.  3. Der Benutzer gibt die Geheimzahl ein.  4. Der Geldautomat liest die Geheimzahl, überprüft sie und lässt dann die BLZ und die Kartennummer beim Konsortium überprüfen.  5. Das Konsortium überprüft die BLZ, gleicht die Kartennummer mit der Bank des Kunden ab und gibt dem ATM sein OK.  6. Der Geldautomat fordert den Benutzer auf die Transaktionsform (Abhebung, Einzahlung, Überweisung, Kontoauszug) zu wählen.  7. Der Benutzer wählt „Abhebung“, woraufhin der Geldautomat den Betrag erfragt.  8. Der Benutzer gibt den gewünschten Betrag ein.  9. Der Geldautomat liest den Betrag, überprüft, ob er innerhalb vordefinierter Grenzen liegt, und fordert dann das Konsortium auf die Transaktion zu verarbeiten.  10. Das Konsortium leitet die Anforderung an die Bank weiter, die den Kontostand aktualisiert und die Ausführung bestätigt.  11. Das Konsortium teilt dem ATM den erfolgreichen Abschluss der Transaktion mit.  12. Der Geldautomat gibt den gewünschten Betrag Bargeld aus und fordert den Benutzer auf es zu entnehmen.  13. Der Benutzer entnimmt das Bargeld, woraufhin der Geldautomat fragt, ob der Benutzer eine weitere Transaktion durchführen will.  14. Der Benutzer verneint.  15. Der Geldautomat druckt einen Beleg, gibt die Karte aus und fordert den Benutzer auf sie zu entnehmen.  16. Der Benutzer entnimmt den Beleg und die Karte.  17. Der Geldautomat fordert den nächsten Benutzer auf eine Kreditkarte einzugeben. |
| Nebenszenarien  (bei Fehlerfällen oder Optionen) | Karte gesperrt, Transaktion gescheitert, Abbruch durch den Benutzer, Karte nicht lesbar, Falsche Geheimzahl, Falsche Bankleitzahl, Grenzen überschritten, Karte abgelaufen, Kein Geld im ATM, Netzwerk unterbrochen  (für jedes Nebenszenario muss entsprechend dem Hauptszenario ein Ablauf erstellt werden). |

Mit Hilfe des UML-Anwendungsfall-Diagramms lässt sich eine Übersicht aller Anwendungsfälle darstellen. Die Schritte einzelner Anwendungsfälle lassen sich beispielsweise mit dem UML-Aktivitätendiagramm oder dem UML-Zustandsautomaten darstellen.

1. (Gause & Weinberg, 1989) nach <https://www.techfak.uni-bielefeld.de/~iluetkeb/2005/session04.pdf> [↑](#footnote-ref-1)
2. Brügge, Dutoit: zitiert nach Prof. Dr. Peter Knauber, FH Mannheim, <http://www.informatik.hs-mannheim.de/~knauber/BCSc-SE/07-f.pdf> [↑](#footnote-ref-2)