# Tutorium: Erhebung von Produktanforderungen durch den Requirements Engineer

Stefan Jesse

Universität Stuttgart
Lehrstuhl für Betriebswirtschaftslehre und Wirtschaftsinformatik II
Breitscheidstr. 2c
70174 Stuttgart
jesse@wi.uni-stuttgart.de

**Abstract:** Das Requirements Engineering als erster Schritt der Systementwicklung entscheidet maßgeblich über den Erfolg oder Misserfolg eines Projektes. Die vier Haupttätigkeiten des Requirements Engineering sind: Erheben, Dokumentieren, Prüfen und Verwalten. Das Requirements Engineering ist eine Schlüsseldisziplin der Systementwicklung und viele weitere Disziplinen hängen von den Ergebnissen des Requirements Engineering ab. Den Stellenwert des Requirements Engineering zeigt der jährliche Bericht der Standish Group "The Scope of Software Development Project Failures". Rund die Hälfte der in dem Bericht untersuchten Projekte erreicht die angestrebten Ziele nicht aufgrund von Fehlern im Requirements Engineering. Ein professionelles Requirements Engineering trägt wesentlich dazu bei, Fehler in der Analyse zu beseitigen. Es beschreibt einen systematischen Weg von der Projektidee über die Ziele zu einem vollständigen Satz an Anforderungen.

## 1 Einführung

Die wesentlichen erfolgsbestimmenden Merkmale eines IT-Produktes werden im Rahmen der Produktdefinition festgelegt. Hier ist es Aufgabe des Requirements Engineers, die wesentlichen Anforderungen der Stakeholder zu bestimmen und Aufgabe des IT-Produktmanagements eine Lösung zu konzipieren, die in der Lage ist, diese Anforderungen unter Berücksichtigung der zu erzielenden Erlöse bzw. zur Verfügung stehenden Ressourcen zu erfüllen. Während die klassische Anforderungsanalyse in die Definition des Produktes und in den Übergang zur (Neu-)Entwicklung einzuordnen ist, ist heute das Verständnis von Anforderungs- und Lösungsmanagement umfassender: insbesondere auch Betrieb, Nutzung und Pflege müssen berücksichtigt werden. Betrieb und Nutzung können nicht zuletzt für die Weiterentwicklung und Pflege wichtige Impulse liefern. Zahlreiche aktuelle Techniken und Methoden in der Entwicklung von IT-Produkten zielen auf eine Verkürzung der Entwicklungsdauer, auf eine Steigerung der Wiederverwendung zur Kostensenkung und Qualitätsverbesserung und auf eine zumindest teilweise Automatisierung der Softwareentwicklung ab. Als Beispiele sind für die Softwarebestandteile Service-orientierte Architekturen, Modell-getriebene Entwicklung, Software-Produktlinien oder Aspekt-orientierte Programmierung und für die Hardwarebestandteile beispielsweise Simultaneous Engineering oder Plattformbauweise zu nennen.

Letztlich entscheidet offensichtlich nicht (nur) die bessere Prozess- bzw. Realisierungstechnologie, sondern (auch) der Stakeholder über den Erfolg eines Produktes. Hier gilt das Zitat von Meister: „Qualität ist das, was der Stakeholder (!) dafür hält“ [Me94]. Daraus kann man schließen, dass nur derjenige, der die Bedürfnisse der tatsächlichen und potentiellen Stakeholder kennt und versteht, in der Lage ist, diese planbar und wiederholbar in erfolgreiche Produkte umzusetzen.

## 2 Der Requirements Engineer

Das Requirements Engineering als erster Schritt der Systementwicklung entscheidet maßgeblich über den Erfolg oder Misserfolg eines Projektes. Die vier Haupttätigkeiten des Requirements Engineering sind das Erheben, Dokumentieren, Prüfen und Verwalten von Anforderungen. Das Requirements Engineering ist eine Schlüsseldisziplin der Systementwicklung und viele weitere Disziplinen hängen von den Ergebnissen des Requirements Engineering ab. Den Stellenwert des Requirements Engineering zeigt der jährliche Bericht der Standish Group “The Scope of Software Development Project Failures”. Rund die Hälfte der in dem Bericht untersuchten Projekte erreicht die angestrebten Ziele nicht aufgrund von Fehlern im Requirements Engineering.

Anforderungen dienen allen am Systementwicklungsprozess Beteiligten als Grundlage für Kommunikation, Diskussion und Argumentation. Ihre Aufgabe ist es, das gemeinsame Verständnis und Wissen der Teammitglieder widerzuspiegeln und zu explizieren. Ein professionelles Requirements Engineering und die damit verbundene organisatorische Installation und Positionierung des Requirements Engineers trägt wesentlich dazu bei, Fehler in der Analyse zu beseitigen. Professionelles Requirements Engineering beschreibt einen systematischen Weg von der Projektidee über die Ziele zu einem vollständigen Satz an Anforderungen.

Zu den Fähigkeiten des Requirements Engineers zählen die klassischen „weichen“ Faktoren, die sogenannten Soft Skills. Requirements Engineers müssen und können nicht alle Aufgaben im Rahmen der Ermittlung, Beschreibung, Abstimmung und Prüfung übernehmen, sondern müssen den strukturierten und zielgeführten Prozess der Anforderungsspezifikation einhalten. Dazu können sie jederzeit auf die Fachkompetenzen und Ressourcen der Anwendungsdomänen zugreifen.

Zu den wichtigsten Fähigkeiten des Requirements Engineers zählen analytisches Denken, selbstbewusstes Auftreten, empathische Fähigkeiten, Moderationsfähigkeit, Methodenkompetenz, Überzeugungsfähigkeit, Kommunikationsfähigkeit und sprachliche Kompetenz.

## 3 Basis-, Leistungs- und Begeisterungsanforderungen

Die Erhebung von Produktanforderungen stellt eine der Hauptaufgaben innerhalb des Aufgabenspektrums des Requirements Engineers dar. Dabei sollte er beachten, dass nicht alle Merkmale eines Produktes gleichermaßen auf Stakeholderzufriedenheit und somit auf den Erfolg wirken. Der Zusammenhang zwischen verschiedenen Leistungskomponenten eines Produktes auf die wahrgenommene Produktqualität und Zufriedenheit der Stakeholder beschreibt das sogenannte – empirisch überprüfte – Kano-Modell [Ka84]. Kano unterscheidet dazu drei Kategorien von Produktanforderungen: Basis-, Leistungs- und Begeisterungsanforderungen. Die Aussagen des Kano-Modells können graphisch in einem zweidimensionalen Koordinatensystem wiedergegeben werden (siehe Abbildung 1).

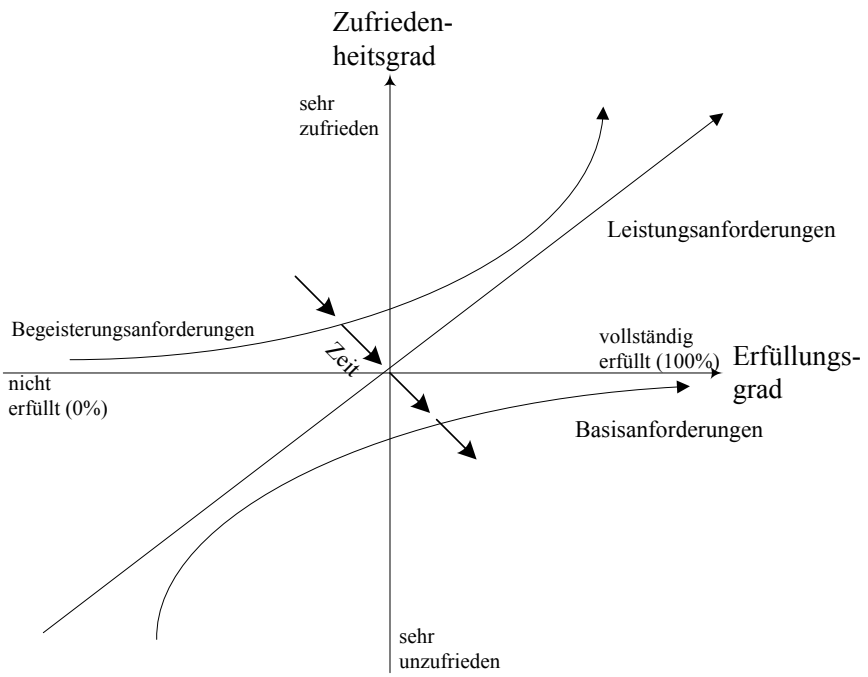


Abbildung 1: Arten von Produktanforderungen im Kano-Modell

Auf der horizontalen Achse wird der Erfüllungsgrad einer Stakeholderanforderung durch die Realisierung entsprechender Produktanforderungen von „nicht erfüllt“ bis zu „vollständig erfüllt“ abgetragen, auf der vertikalen Achse der dadurch bedingte Zufriedenheitsgrad mit der Erfüllung der Anforderung von „sehr unzufrieden“ bis „sehr zufrieden“. Die drei eingezeichneten Graphen repräsentieren die oben genannten Kategorien von Produktanforderungen. Unter diesen spiegeln nur die *Leistungsanforderungen* solche Anforderungen wider, die vom Stakeholder explizit verlangt werden. Sie werden deswegen auch mit „gewünschter“ oder „eindimensionaler Qualität“ gleichgesetzt, denn die Zufriedenheit steigt gleichmäßig mit zunehmendem Erfüllungsgrad.

Die *Basisanforderungen* entsprechen der „erwarteten Qualität“ des Produktes. Der Stakeholder verlangt oftmals nicht explizit nach diesen Produktmerkmalen, die Erfüllung der sich dahinter verbergenden Anforderungen wird als selbstverständlich vorausgesetzt. Deswegen fallen diese Charakteristika des fertigen Produktes nur bei unzureichender Umsetzung als „Quellen der Unzufriedenheit“ auf, werden allerdings auch bei nahezu perfekter Abdeckung auf der Zufriedenheitsskala nicht besonders gewürdigt.

B*egeisterungsanforderungen* beziehen sich auf Produktmerkmale, die vom Stakeholder nicht erwartet werden und in erster Linie der Kreativität und Innovationskraft der Entwickler entstammen. Oftmals geben sie unbewusste und versteckte Wünsche der Stakeholder wieder, die bei unzureichender Umsetzung nicht zu Unzufriedenheit führen, durch ihre Existenz allerdings zu „Quellen hoher Zufriedenheit“ werden können. Daher sind sie auch nicht mit „nice to have“-Funktionen zu verwechseln, da diese zwar vom Stakeholder auch nicht zwangsweise erwartet werden, aber eben auch nicht zu besonderer Begeisterung beim Stakeholder führen, wenn sie enthalten sind.

Ebenso muss betont werden, dass bei weitem nicht alle Produktmerkmale, die der Kreativität der Entwickler entspringen, wirklich vom Stakeholder gewünscht werden. Folglich ist die methodisch unterstützte Rückversicherung, welche Produktmerkmale Begeisterungsanforderungen erfüllen, von großer Bedeutung.

Zu berücksichtigen ist ferner noch der Zeitfaktor. Produktanforderungen, die zum heutigen Zeitpunkt Begeisterungsanforderungen darstellen, werden im nächsten Release vielleicht schon explizit von den Stakeholdern gefordert und in noch fernerer Zukunft als selbstverständlich vorausgesetzt. In den Kategorien von Produktanforderungen spiegelt sich also, in Kenntnis der oft hohen Eigendynamik des Softwaremarktes, der Zwang zur ständigen Verbesserung wider.

## 4 Erfolgsfaktor Stakeholderorientierung

Der Grundstein für erfolgreiche IT-Produkte wird bereits in der Produktentwicklungsphase gelegt. Entspricht schon die Konzeption nicht den Erwartungen der Stakeholder, kann auch die Gestaltung der nachfolgenden Produktions- und Absatzprozesse an der voraussichtlich entstehenden Stakeholderunzufriedenheit kaum etwas ändern. Die Produktentwicklung in Deutschland weist diesbezüglich jedoch noch zahlreiche Schwächen auf, die sich z. B. in der mangelnden Kommunikation der Produktentwicklung mit den Stakeholder- bzw. absatznahen Bereichen oder in dem Hang zur technischen Perfektion statt der Befriedigung von Stakeholderbedürfnissen manifestieren.

Dass Stakeholderorientierung ein sehr wichtiges Thema für Softwareunternehmen ist, bestätigt eine aktuelle Umfrage unter Verantwortlichen für das Thema Produktmanagement in deutschen Softwareunternehmen. Befragt, wie sehr wettbewerbsentscheidend verschiedene Faktoren sind, wurde die Passgenauigkeit ihrer Lösung/ihres Software-Produktes hinsichtlich der Anforderungen der Stakeholder von den Teilnehmern am stärksten gewichtet. Knapp dahinter und quasi gleichbedeutend landeten die technische Qualität der Software, die richtige Identifikation unklarer Stakeholderanforderungen, die Flexibilität der Software sowie der Gesamtumfang der Funktionalität der Software. Service und Support landeten auf dem vorletzten, der Preis auf dem letzten Platz [HHK07].

Hier ist der Produktmanager gefordert, die Stimme der Stakeholder ins Unternehmen zu tragen und der Requirements Engineer muss sicherstellen, dass die Stakeholderanforderungen richtig erkannt und umgesetzt werden. Der Requirements Engineer muss das Produktmanagement durch die Auswahl geeigneter Befragungs- und Beobachtungstechniken unterstützen, um implizite wie explizite Bedürfnisse der Stakeholder aufzunehmen.

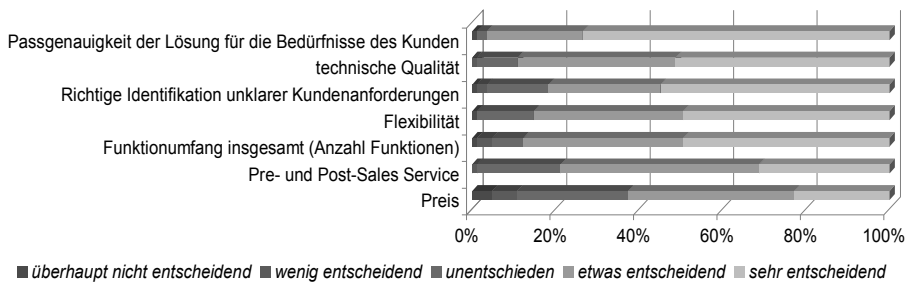


Abbildung 2: Bedeutung verschiedener Faktoren im Wettbewerb [HHK07]

## 5 Erhebung von Anwenderanforderungen

Theoretisch ist die Erhebung und Verwaltung von Anwenderanforderungen einfach: Man muss die Anwender des Produktes nur fragen, was sie haben wollen, das entsprechend im Produkt umsetzen und anschließend die Wünsche zur Verbesserung, die in aller Regel automatisch kommen, in der Weiterentwicklung und Verbesserung des Produktes berücksichtigen.

Wäre dies in der Praxis genauso einfach, wäre das Leben der meisten Requirements Engineers oder auch IT-Produktmanager viel einfacher. Stattdessen kämpfen diese damit, dass das Management von Stakeholderanforderungen deutlich schwieriger ist. Die Passgenauigkeit der eigenen Lösung für die Bedürfnisse der Stakeholder, aber auch die richtige Identifikation unklarer Stakeholderanforderungen werden von den Teilnehmern der Studie als deutlich wichtiger empfunden als z. B. der Preis. Da die Funktion des Produktmanagers ja gerade sein soll, die Stimme des Stakeholder in das Unternehmen zu tragen und dafür zu sorgen, dass die angebotenen Produkte den Stakeholderwünschen entsprechen, ist das Management von Stakeholderanforderungen eine der wichtigsten Herausforderungen, denen Requirements Engineers sich stellen müssen.

Die Erhebung von Anforderungen ist dabei diejenige Teilaufgabe, die für die Ermittlung der „qualitativen und quantitativen Eigenschaften eines Produkts aus der Sicht des Auftraggebers“ [Ba01] zuständig ist. Das Management von Anforderungen umfasst zusätzlich noch Maßnahmen zur Steuerung, Kontrolle und Verwaltung der Anforderungen.

Fast alle Aufgaben von Anforderungsanalyse und Anforderungsmanagement sind problembehaftet:

1. Definition der Anforderungen: Hier sind Ist-Analysen (also sowohl Beschreibung als auch Bewertung des Ist-Zustands) durchzuführen, die Anwendungsziele zu bestimmen und schließlich System- oder Produktanforderungen festzulegen,

2. Beschreibung der Anforderungen: Die Anforderungen müssen systematisiert und priorisiert werden sowie die Abhängigkeiten zwischen den Anforderungen aufgezeigt werden,

3. Analyse der Anforderungen: Die Anforderungen müssen konsistent, vollständig und richtig beschrieben werden und realisierbar sein,

4. Dokumentation der Anforderungen: Schließlich müssen die Anforderungen noch in einer Anforderungsspezifikation dokumentiert werden, damit man später auf sie zugreifen kann.

5. Verfolgung und Pflege der Anforderungen: Die Anforderungen müssen auch projekt- bzw. produktübergreifend verwaltet und abgestimmt werden.

Im Folgenden werden einige wichtige Probleme, die bei der Erfüllung der o. g. Aufgaben häufig auftreten, noch etwas genauer erläutert; beispielsweise ist es alleine schon schwierig, „den Stakeholder“ zu identifizieren: in aller Regel werden IT-Produkte von verschiedenen Stakeholdergruppen, welche das Produkt zu ganz unterschiedlichen Zwecken einsetzen, verwendet. Besonders groß ist dieses Problem naturgemäß beim Standardprodukt, aber auch bei der Entwicklung einer Individualsoftwarelösung existieren mehrere Stakeholdergruppen, so zum Beispiel häufige Nutzer, unregelmäßige Nutzer, Buying Center (welche die Software nicht nutzen, aber die Anbieterauswahl betreuen), das Management (welches die Software auch nicht nutzt, aber das Budget freigibt) und die IT-Abteilung (welche die Software betreibt und administriert). Sind all diese Stakeholdergruppen identifiziert und beginnt man, ihre Anforderungen an das Produkt zu erheben, so wird man in aller Regel feststellen, dass sie ganz unterschiedliche Anforderungen stellen. Beispielsweise wird der häufige Nutzer tendenziell viele Funktionen fordern, evtl. auch die Möglichkeit, die gesamte Anwendung rein mit Tastenkombinationen zu bedienen[1], der unregelmäßige Nutzer wird vor allem eine übersichtliche Benutzeroberfläche fordern (was einer großen Zahl von Funktionen eher widerspricht), Buying Center und Management werden stark auf den Preis schauen, die IT-Abteilung schließlich achtet auf geringe Systemanforderungen, Schnittstellen und einfache Administration.

Wie man sieht, gilt es, eine Vielzahl an – sich teilweise widersprechenden – Anforderungen unter einen Hut zu bringen. Dabei stellt sich insbesondere auch die Frage, wie wichtig verschiedene Anforderungen sind oder wie man bei sich widersprechenden Anforderungen vorzugehen hat. Diese Priorisierung kann nur durch die Stakeholder geschehen und sollte auch verwendet werden, um bei knappem Budget oder Terminplan entscheiden zu können, welche Funktionen nicht mehr umgesetzt werden können.

Ein weiteres Problem besteht darin, dass Anforderungen oftmals nicht eindeutig formuliert sind, beispielsweise die Anforderung: „Diese Funktion bewirkt, dass alle europäischen Umsätze des letzen Quartals zusammengezählt und dem Benutzer zurückgemeldet werden.“ Während diese Anforderung im ersten Moment relativ klar klingt, wird spätestens dem Entwickler beim Versuch, dies zu implementieren auffallen, dass einige Fragen offen sind: Umsätze aus welchen Ländern sollen berücksichtigt werden? Zählt das Unternehmen nur EU-Mitglieder zu Europa, was ist beispielsweise mit der Türkei (liegt bekanntlich mehrheitlich auf asiatischem Boden) und vor allem auch: in welcher Währung soll der Umsatz berechnet werden? Welcher Wechselkurs ist anzusetzen? In der Praxis kommt es leider oftmals vor, dass der Entwickler hier nicht nachfragt, sondern selbst eine Entscheidung trifft, die aber leider nicht dem entspricht, was der Stakeholder haben wollte. Daher wäre nachfragen besser, nur stellt sich bei einem normalen Pflichtenheft, das schnell fünfzig und mehr oder gar hunderte von Seiten umfasst, die Frage, bei wem man denn nachfragen soll. Folglich ist die Nachverfolgbarkeit von Anforderungen (warum? und wer wollte das so?) von großer Bedeutung.

[1] Während gelegentliche Nutzer eine grafische Oberfläche mit Mausbedienung in aller Regel vorziehen, sind häufige Nutzer oftmals schneller, wenn sie die Software komplett mit der Tastatur bedienen können, also nicht ständig zwischen Tastatur und Maus wechseln müssen.

Neben den primären Aufgaben von Anforderungen (Kommunikationsgrundlage, Vertragsgrundlage, Grundlage für Design und Implementierung, Grundlage für die Systemarchitektur sowie Grundlage für Installation und Wartung) existieren aber auch sekundäre Aufgaben von Anforderungen, die oftmals vergessen werden: insbesondere der angestrebte Geschäftsnutzen der Nutzung des IT-Produktes sollte (quasi als Vision) in der Anforderungsspezifikation mit aufgeführt sein. Beispielsweise sollen durch die Einführung eines neuen IT-Produktes, sei es eine CRM-Software oder mobile Endgeräte in der Lagerverwaltung, im Unternehmen oftmals Rationalisierungspotenziale gehoben werden. Eventuell liegt der hauptsächliche Stakeholdernutzen aber auch in der Integration bislang auf verschiedene Systeme verteilter Daten oder in der Erhöhung der Mitarbeiterzufriedenheit (wenn die Mitarbeiter von stupiden und repetitiven Tätigkeiten entlastet werden).

## 6 Einsatz von Befragungs- und Beobachtungstechniken zur Ergebung von Anwenderanforderungen

Da das Ermitteln von Anforderungen zu den wichtigsten Aufgaben des Requirements Engineers gehört, werden im Folgenden einige Instrumente zur Unterstützung des Requirements Engineers vorgestellt. Wie bereits im Rahmen des KANO-Modells erläutert, äußern Stakeholder explizit nur Leistungsanforderungen, d. h. Anforderungen die bei Erfüllung Stakeholderzufriedenheit hervorrufen und bei Nichterfüllung zu Stakeholderunzufriedenheit führen. Diese Leistungsanforderungen können gut von den Stakeholdern erfragt werden, da diese Art von Anforderungen dem Stakeholder bewusst sind.

Basisfaktoren sind dem Stakeholder ebenfalls bewusst, aber im Unterschied zu den Leistungsanforderungen sind diese dem Stakeholder oftmals zu selbstverständlich oder zu „trivial“, um explizit verlangt zu werden. Da das Vorhandensein von Basisfaktoren im Produkt jedoch zwingend ist, da ihr Vorhandensein bestenfalls zu einem Zustand der Nicht-Unzufriedenheit führt und das Vergessen von Basisanforderungen zu massiver Unzufriedenheit oder Produktablehnung führt, müssen diese Faktoren ebenfalls vom Requirements Engineer erhoben werden. Zur Erhebung, Definition und genauen Bestimmung von Basis- und Leistungsfaktoren stehen dem Requirements Engineer Beobachtungs- und Befragungstechniken zur Auswahl. Zu den Beobachtungstechniken zählen beispielsweise die Feldbeobachtung sowie das Apprenticing, zu den Befragungstechniken zählen u. a. Fragebogen und Interviews [Ru07].

**Beobachtungstechnik Feldbeobachtung**

Bei der Feldbeobachtung werden die Stakeholder an ihren Arbeitsplätzen beobachtet. Während des Beobachtens erkennt der außen stehende Betrachter nicht optimale Arbeitsschritte und Fehler, welche dem Stakeholder aus Gewohnheit nicht mehr bewusst sind. Beobachten lassen sich dabei natürlich nur Arbeitsabläufe, die bereits gelebt werden und vom Stakeholder auch tatsächlich durchgeführt werden. Von daher kann diese Technik nicht für völlig neue Prozesse angewandt werden. Vorteil der Methode ist, dass der Analytiker einen unverfälschten Eindruck der Sprache und Terminologie des Fachgebietes gewinnt. Darüber hinaus kann die Methode leicht durch Audio- oder Video-Aufzeichnungen unterstützt werden.

**Beobachtungstechnik Apprenticing**

Beim Apprenticing führt der Analytiker die Tätigkeiten der Stakeholder konkret aus. Wie ein Lehrling ist der Analytiker gefordert, unklare und unverständliche Handlungsschritte sofort zu hinterfragen, um in diesem Bereich Erfahrung zu sammeln. Der Analytiker kann hierdurch Anforderungen erfahren, welche für den Stakeholder so selbstverständlich wurden, dass er sie nicht mehr äußert. Einschränkend ist, dass die Methode in einem sicherheitskritischen Umfeld nicht uneingeschränkt einsetzbar ist. Bei der Apprenticing-Methode überbrückt der Analytiker durch eigene Erfahrung mangelnde kommunikative Fähigkeiten des Stakeholders.

**Befragungstechnik Interview**

Beim Interview kann vom Stakeholder eine genaue Aussage über die Anforderungen an das zu untersuchende System ermittelt werden. Beim Interview stellt der Analytiker einem oder mehreren Stakeholdern vorgegebene Fragen und protokolliert deren Antworten. Interviews setzen voraus, dass der Interviewte sein Wissen explizit ausdrücken kann. Gerade zu Beginn der Anforderungsermittlung eignen sich vor allem persönliche Interviews, in denen gemeinsam grobe Anforderungen an das System erarbeitet werden. Nachteil ist, dass (Einzel-)Interviews sehr zeitaufwändig sind und eine sehr persönliche Färbung des Systems durch die jeweiligen Befragten beinhalten.

**Befragungstechnik Fragebogen**

Bei dieser Befragsungstechnik werden Fragen zur Beantwortung durch den Stakeholder gesammelt. Fragebögen sind insbesondere bei einer Produktentwicklung gut geeignet, geplante Eigenschaften des Produkts von einer großen Zahl von Testpersonen bewerten zu lassen und in offenen Fragen Verbesserungsmöglichkeiten zu ermitteln. Insbesondere für größere Stakeholder-Anzahlen ist diese Befragungstechnik gut geeignet, setzt jedoch voraus, dass der Befragte sein Wissen explizit ausdrücken kann. Der Befragte muss weiterhin bereit sein, Zeit und Engagement in die Ermittlung zu investieren. Problemtisch ist, dass der Analytiker die zu beantwortenden Fragen vorgibt. Anliegen der Stakeholder können unter Umständen verdrängt, vergessen oder vernachlässigt werden.

## 7 Schwerpunkte des Tutoriums

- Vorstellung des Zertifizierungsprogramms „Certified Professional Requirements Engineerings“
- Grundlagen des Requirements Engineering
  - Einflussfaktoren des Requirements Engineering; Profil des Requirements Engineer
- Scope, Kontext und Schnittstellen
- Anforderungsarten
  - Überblick über Anforderungsarten; Anforderungen an das Verhalten
  - Daten/Informationsanforderungen; Nicht funktionale Anforderungen
  - Schnittstellen-Anforderungen
- Anforderungsdokumente
  - Dokumentengestaltung; Glossar
  - Repräsentationsformen für Anforderungen
- Anforderungen gestalten
  - Quellen für Anforderungen; Ermittlung von Anforderungen; Prüfen von Anforderungen
  - Stakeholder
- Werkzeuge für das Requirements Engineering
- Informationen über neue Entwicklungen im Requirements Engineering
- Checklisten und Templates für Requirements Engineering

## Literaturverzeichnis


[Me94] Meister, H.: Qualität ist das, was Kunden (!) dafür halten, in: Marketing Journal, 1994

[Ka84] Kano, N., Seraku, N., Takahashi, F. und Tsuji, S.: Attractive quality and must-be quality, in: Quality, Vol.14, 1984, No.2, S.39-44.

[HHK07] Helferich, A., Hiller, K.; Kneher, S.: Variantenmanagement in der Softwareentwicklung - Ergebnisse einer Umfrage unter deutschen Softwareanbietern, Stuttgarter Schriften zur Unternehmenssoftware Nr. 1, Stuttgart 2007

[Ba01] Balzert, H.: Lehrbuch der Software-Technik – Bd. 1: Software-Entwicklung, 2. Aufl., Heidelberg u. a. 2001

[Ru07] Rupp, C.: Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis, 4. Aufl., München 2007