



**Fraunhofer** Institut  
Experimentelles  
Software Engineering

# Das ReqMan Prozessrahmenwerk



**Autoren:**

Jörg Dörr  
Tom Koenig  
Thomas Olsson  
Sebastian Adam

IESE-Report Nr. 141.06/D  
Version 1.0  
1. Oktober 2006

---

Eine Publikation des Fraunhofer IESE



Das Fraunhofer IESE ist ein Institut der Fraunhofer-Gesellschaft. Das Institut transferiert innovative Software-Entwicklungstechniken, -Methoden und -Werkzeuge in die industrielle Praxis. Es hilft Unternehmen, bedarfsgerechte Software-Kompetenzen aufzubauen und eine wettbewerbsfähige Marktposition zu erlangen.

Das Fraunhofer IESE steht unter der Leitung von  
Prof. Dr. Dieter Rombach (geschäftsführend)  
Prof. Dr. Peter Liggesmeyer  
Fraunhofer-Platz 1  
67663 Kaiserslautern



## Abstract

Dieser Bericht beschreibt die theoretischen Grundlagen des ReqMan-Rahmenwerkes und die Motivation, die zu seiner Entwicklung geführt hat.

**Schlagworte:** ReqMan, requirements engineering, process improvement



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Verwandte Arbeiten</b>	<b>2</b>
<b>3</b>	<b>Motivation</b>	<b>4</b>
3.1	Hohe Komplexität existierender Ansätze	4
3.2	Existierende Ansätze sind zu unflexibel	5
3.3	Mehrwert der Prozessverbesserung ist unklar	5
3.4	Zusammenfassung	5
<b>4</b>	<b>Das Rahmenwerk</b>	<b>7</b>
4.1	Das Konzept der Praktiken und Techniken	7
4.2	Die Aktivitäten	8
4.3	Die Praktiken des Frameworks	9
4.4	Die Techniken	12
<b>5</b>	<b>Zusammenfassung</b>	<b>14</b>
	<b>Literatur</b>	<b>15</b>
	<b>Anhang A: Praktikbeschreibungen</b>	<b>17</b>





# 1 Einleitung

Anforderungen sind ein wichtiger Teil des Softwareentwicklungsprozesses, gleichzeitig jedoch schwer zu erheben, zu dokumentieren, zu modellieren, zu analysieren und in der weiteren Entwicklung zu verwalten. Anforderungen sind wichtig, da sie die Grundlage für die gesamte Entwicklung bilden und als Schnittstelle zwischen Kunden und Entwickler fungieren. Dabei ist es oft problematisch, dass viele Anforderungen nicht greifbar sind und sich oft nur in den Köpfen der Kunden und anderer Stakeholder befinden.

Das Sammeln und Wiederverwenden von Erfahrung ist oftmals ein wichtiges und hilfreiches Mittel. Auch im Bereich „Anforderungsmanagement“ gibt es inzwischen einen fundierten Satz von Erfahrungen aus Forschung und Praxis. So hat sich beispielsweise gezeigt, dass man durch systematisches Arbeiten schneller Ergebnisse mit weniger Fehlern erzielt, als mit einer unstrukturierten ad hoc Vorgehensweise.

In diesem Bericht wird ein Requirements Engineering-Rahmenwerk präsentiert, welches im Rahmen des ReqMan-Forschungsprojektes entwickelt wurde. Die Ziele dieses Rahmenwerks sind auf der einen Seite die wichtigsten Aufgaben und Praktiken des Requirements Engineering zu beschreiben und auf der anderen Seite zugehörige pragmatische Techniken zu sammeln, die bereits ausprobiert sind und im praktischen Einsatz Erfolg gezeigt haben.

## 2 Verwandte Arbeiten

Sowohl Ansätze welche versuchen, systematische Prozessverbesserung im Requirements Engineering Umfeld anzubieten als auch solche, die adaptierbare Prozess-Rahmenwerke zur Verfügung stellen, sind nicht neu.

Bei den Ansätzen, die eine gezielte Verbesserung der Requirements Engineering Prozesse zum Ziel haben, kann insbesondere R-CMM [Beech05] angeführt werden. Der Ansatz orientiert sich am etablierten CMM(I)-Standard und überträgt entsprechende Konzepte gezielt und entsprechend vertiefend auf die Ebene des Requirements Engineering. Hierdurch wird eine gezielte und systematische Bewertung (einschließlich der Bestimmung von Reifegraden) und daraus resultierende Verbesserung ermöglicht. Insbesondere für kleinere und mittelständige Unternehmen ist dieser Ansatz jedoch umfangreich und einschüchternd und bietet darüber hinaus wenig Unterstützung bei der Ausschöpfung der identifizierten Verbesserungspotentiale.

Innerhalb der Rahmenwerk-/Baukasten-orientierten Ansätze, sind vor allem der „Good Practice Guide“ [Somm97], das Open Process Framework [Firesm05] und der Rational Unified Process [Krucht99] von Bedeutung.

Der „Good Practice Guide“ von Sommerville gibt pragmatische Richtlinien und konkrete Praktiken, mit denen ein Requirements Engineering Prozess verbessert werden kann. Nachteilig ist jedoch, dass die Praktiken der sicherheitskritischen Domäne entsprechen und nur schwer auf andere Projektkontexte übertragbar sind.

Das Open Process Framework bietet einen umfassenden, flexiblen Methodenbaukasten, um ganze Softwareentwicklungsprozesse, darunter auch den RE-Prozess aus etablierten, wiederverwendbaren Prozesskomponenten zusammenzusetzen. Aufgrund seiner Mächtigkeit ist er allerdings für kleinere und mittelständige Unternehmen schwer zu verstehen und anzupassen. Ähnliche Vor- und Nachteile besitzt auch der Rational Unified Process, welcher trotz seiner großen Akzeptanz in der Industrie insbesondere in kleineren Softwareunternehmen nur mit erheblichem Aufwand einzuführen ist.

Bei der Entwicklung des ReqMan-Rahmenwerkes wurden daher gezielt die existierenden Ansätze einschließlich ihrer Stärken und Schwächen berücksichtigt. Prozessionsstandards wie CMM, SPICE oder State-of-the-Art-Referenzen wie

SWEBOK<sup>1</sup> dienten als Strukturierungsmittel aber auch um sicherzustellen, dass das Rahmenwerk konsistent mit existierenden Ansätzen ist.

Angereichert um weitere State-of-the-Art-Lösungen zu verschiedenen Problemen und empirischen Praxiserfahrungen wurde daraus ein pragmatischer, aber auch konzeptionell fundierter Requirements Engineering Baukasten, der insbesondere die Bedürfnisse kleinerer und mittelständiger Unternehmen adressiert.

---

<sup>1</sup> [www.swebok.org](http://www.swebok.org)

## 3 Motivation

Kleine und mittelständige Unternehmen (KMU) sind sich größtenteils bewusst, dass die Grundlagen für konsequente Qualitätssicherung bereits in den frühen Phasen der Softwareentwicklung, insbesondere der Anforderungsphase, gelegt werden müssen. Trotz dieser Erkenntnis ist es meist schwer, Unternehmen generell (Hall 02; Juristo02; Rainer03) und insbesondere KMU (Kams98) davon zu überzeugen ihren Anforderungsprozess zu verbessern. An manchen Stellen wirken sich Rahmenbedingungen, wie etwa Skepsis der Entwickler gegenüber der Anforderungsthematik, nicht dokumentierte und meist wenig formelle Anforderungs- und Software-Entwicklungsprozesse, sowie nicht explizit definierte Rollen, negativ auf die Anwendung der Prozessverbesserung aus (Kams98). Diese Rahmenbedingungen sind aber meist nicht die Ursache für die zum Teil ablehnende Haltung gegenüber der Prozessverbesserung, vielmehr konnten die Aspekte in Abschnitt 3.1-3.3 erkannt werden. Dieser Aspekte dienen als Anforderung an ReqMan.

### 3.1 Hohe Komplexität existierender Ansätze

Existierende Prozessmodelle wie etwa der Rational Unified Process oder das V-Modell wurden bei vielen Unternehmen erfolgreich eingeführt. Allerdings sind viele dieser existierenden Ansätze äußerst komplex und abstrakt und somit schwer zu verstehen oder gar anzuwenden. Im Bereich der KMU leiten sich daraus meist zwei Problemstellungen ab, die dazu führen, einen solchen Ansatz nicht direkt anzuwenden: Zum einen fehlt einer KMU oft die Zeit sich vollständig in komplexe Ansätze einzuarbeiten und zum anderen mangelt es einer KMU teilweise auch an Erfahrung und theoretischem Hintergrundwissen um die Methode korrekt auf den Unternehmenskontext anzuwenden. Beides sind Randbedingungen, die oftmals als Voraussetzung für die Anwendung dieser Ansätze gelten (Nikula00; Hall02; Emam95). Eine indirekte Anwendung der erwähnten Ansätze durch externe Berater wird von vielen KMU ebenfalls als unakzeptabel angesehen, da Berater oftmals teuer sind oder aber das Vertrauen des Unternehmens nicht genießen.

Erforderliche Charakteristiken: Der Ansatz muss zum einen leicht verständlich sein, zum anderen sollte er auch schnell anzuwenden sein.

### 3.2 Existierende Ansätze sind zu unflexibel

Es gibt einen essentiellen Unterschied zwischen den theoretischen Grundlagen eines Standard-Rahmenwerks und der praktischen Anwendung im Kontext einer KMU, insbesondere bezogen auf den erforderlichen Anpassungsaufwand (Nikula00). Existierende Ansätze sind meist nicht auf den spezifischen Entwicklungskontext des Unternehmens zugeschnitten und erfordern aus diesem Grund meist einen hohen Anpassungsaufwand, um möglichst optimale Ergebnisse für eine KMU liefern zu können. Eine erfolgreiche Verbesserung des Anforderungsprozesses kann nicht losgelöst von der existierenden Entwicklungsumgebung durchgeführt werden.

Erforderliche Charakteristiken: Der Ansatz muss möglichst flexibel und anpassbar sein, um optimal auf unterschiedlichste Entwicklungskontexte reagieren zu können.

### 3.3 Mehrwert der Prozessverbesserung ist unklar

Es ist oftmals für ein Unternehmen schwer abzuschätzen, ob eine Verbesserungsmaßnahme wirklich zu dem gewünschten positiven Effekt bezogen auf den Anforderungsprozess führt. Insbesondere ist häufig unklar welches Ziel eine komplexe angeratene Verbesserungsmaßnahme genau verfolgt. Eine Prozessverbesserungsmethode sollte stets zu einer einfachen und verständlichen Prozessanpassung führen, die von Seiten aller Stakeholders akzeptiert wird (Kaindl02). Unsere Erfahrung zeigte, dass Unternehmen sich in diesem Bereich insbesondere auch erfahrungsbasierte Werte wünschen, die ihnen eine Einschätzung einer vorgeschlagenen Methode zur Verbesserung ermöglichen. Empirische Daten und Erfahrungswerte, im Hinblick auf die pragmatische Anwendung verschiedenster Verbesserungen, sind meist nicht verfügbar. In der Praxis könnten insbesondere solche Daten Unternehmen befähigen den positiven Effekt einer potentiellen Verbesserungsmaßnahme bereits a priori einschätzen zu können.

Erforderliche Charakteristiken: Die angeratenen Verbesserungen müssen leicht verständlich und anwendbar sein. Der Ansatz soll die Einschätzung bestimmter Maßnahmen durch Integration einer empirischen Erfahrungsbasis ermöglichen.

### 3.4 Zusammenfassung

Zusammengefasst ergeben sich folgende Anforderungen an ein Prozessverbesserungsrahmenwerk:

- Es muss leicht verständlich und nachvollziehbar sein,

- Es muss schnell anzuwenden sein,
- Es muss möglichst flexibel sein,
- Es soll Integration einer empirischen Erfahrungsbasis ermöglichen,
- Es muss nah am Entwicklungskontext sein und konkrete Verbesserungen anbieten,
- Es soll hohe Akzeptanz bei den Anwendenden haben.

## 4 Das Rahmenwerk

Die in Kapitel 3 genannten Probleme führen oftmals dazu, dass KMU eine eher ablehnende Haltung gegenüber der methodischen und systematischen Verbesserung ihres Anforderungsprozesses einnehmen. Häufig wird die Prozessverbesserung, falls überhaupt, eher unsystematisch und ad hoc durchgeführt (Kams98; Nikula00). Im Rahmen einer solchen ad hoc Verbesserung des Anforderungsprozesses birgt sich das Risiko, potentiell ungeeignete oder gar falsche Methoden auszuwählen und einzusetzen, die dem eigentlichen Zweck, der Prozessverbesserung, kontraproduktiv entgegenstehen.

Das zentrale Ziel des Reqman Ansatzes ist es, ein Unternehmen davon zu überzeugen, seine ablehnende Grundhaltung gegenüber der systematischen Verbesserung des Anforderungsprozesses abzulegen. Dieses Ziel wird verfolgt, in dem der Reqman Ansatz basierend auf den identifizierten Charakteristiken erstellt wurde.

### 4.1 Das Konzept der Praktiken und Techniken

Das ReqMan-Rahmenwerk ist ein pragmatischer Requirements Engineering Baukasten für kleine und mittelständige Unternehmen (KMU), welcher auf die Definition eines vorgefertigten Prozesses verzichtet. Vielmehr werden Praktiken und Techniken, sowie Hilfsmittel wie z.B. Templates bereitgestellt. Diese Elemente ermöglichen es, einen eigenen, an das jeweilige Unternehmen angepassten, Requirements Engineering-Prozess zusammen zu stellen.

Die Erfahrung zeigte, dass es gerade für KMU schwierig bis unmöglich ist, einen Standardprozess oder einen generischen Standardprozess zu erstellen. Typischerweise werden in einer KMU innerhalb des Requirements Engineering verschiedene Praktiken beherzigt und umgesetzt. Ein Prozessrahmenwerk mit aneinander logisch oder gar zeitlich gekoppelten Aktivitäten, welches viele Verzweigungen und alternative Wege aufweist, wirkt auf KMUs abschreckend, da sie meist den RE-Prozess als eine kleine Menge von Aktivitäten (meist nur 1 Aktivität) auffassen.

Das Rahmenwerk enthält 3 Konzepte (siehe auch Figure 1):

- Aktivitäten – übergreifende Einordnung in die Phasen eines typischen RE-Prozess
- Praktiken – abstrakte Aufgaben, die in den meisten Kontext zu einer qualitativen Verbesserung der RE-Prozess führen

- Techniken – konkretes Vorgehen, das direkt einsetzbar ist, um eine Praktik umzusetzen

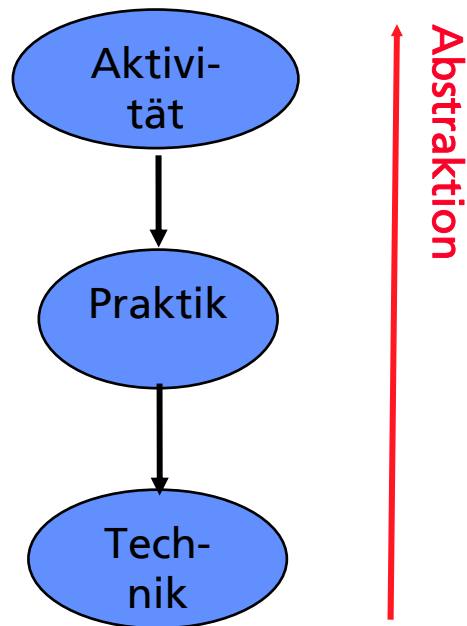


Figure 1

Zusammenhang Aktivitäten, Praktiken, Techniken

Basierend auf dieser Grundlage vermittelt unser Ansatz dem einzelnen Unternehmen nicht nur an welchen Stellen eine Prozessverbesserung erforderlich ist sondern auch, wie eine solche Verbesserung aussehen kann.

## 4.2 Die Aktivitäten

Das Rahmenwerk unterscheidet fünf Hauptaktivitäten des Requirements Engineering, in denen die einzelnen Praktiken eingeordnet sind. Für die Bezeichnung "Aktivität" verwenden wir alternativ auch den Begriff "Prozessphase". Dieser ist allerdings etwas irreführend, da die Aktivitäten sich nicht strikt zueinander anordnen lassen. Der Versuch sie gar sequentiell anzuordnen bildet die Realität in KMU meist nicht ab. Beispielsweise findet während eines Workshops mit dem Kunden meist eine Mischform aus Anforderungserhebung, -analyse und -spezifikation statt.



### **Anforderungserhebung**

Der Anforderungsingenieur identifiziert die für das Projekt/Produkt relevanten Stakeholder und Anforderungsquellen. Er wendet Praktiken und Techniken an, um Kundenwünsche und –ziele zu identifizieren und gewünschte Produktfeatures sowie funktionale und nichtfunktionale Anforderungen zu erheben und mit dem Kunden abzustimmen.

### **Anforderungsanalyse**

In dieser Aktivität werden die Anforderungen analysiert und dazu, wenn nötig, auch modelliert (nicht zum Zwecke der Spezifikation). Machbarkeitsanalysen sowie Geschäftsprozessanalysen sind in dieser Aktivität genauso anzusiedeln wie eine einfache Klassifikation der Anforderungen.

### **Anforderungsspezifikation**

Die Kunden- und Entwickleranforderungen werden hier ggf. unter Verwendung von Standards und Dokumententemplates dokumentiert. Dies kann eine vielseitige Dokumentation mit verschiedenen Sichten auf die Anforderungen sein oder wie aus dem Extreme Programming bekannt, einfache User Stories in Karteikarten. Weiterhin können die Anforderungen in Worddokumenten oder komplexeren Requirements Management Tools dokumentiert sein.

### **Anforderungsverifikation und -validierung**

In dieser Aktivität werden die Anforderungen einer Qualitätssicherung unterzogen. Dabei werden verschiedene Qualitätsmerkmale (z.B. Verständlichkeit, Messbarkeit, Vollständigkeit, Widerspruchsfreiheit, Benutzbarkeit) begutachtet.

### **Anforderungsmanagement**

Diese Aktivität beschäftigt sich mit der Verwaltung der Anforderungen. Gerade für KMU verschmelzen hier häufig Projektmanagement und Anforderungsmanagement. Für Reqman, wird daher Anforderungsmanagement etwas breiter ausgelegt, als es in der Literatur üblich ist. Praktiken des Anforderungsmanagement umfassen beispielsweise die Auswahl von Technologien für die anderen vier Aktivitäten, Risikoevaluation, Kosten- und Zeitschätzung sowie Verfolgbarkeit einzuführen und Änderungen an den Anforderungen zu verwalten.

## **4.3 Die Praktiken des Frameworks**

Die zentralen Bausteine des erstellten Rahmenwerks sind jedoch so genannte Praktiken und Techniken. Eine Praktik ist dabei eine abstrakte Aktivität, zum Beispiel „Ziele erheben“, die man prinzipiell in jedem Anforderungskontext verfolgen kann. Wichtig ist, dass eine Praktik keine Aussage darüber macht wie eine bestimmte Aktivität durchzuführen ist. Diesbezüglich wurde das Konzept der

Technik eingeführt. Eine Technik dokumentiert eine konkrete Realisierung einer oder mehrerer Praktiken, so zum Beispiel „Use Case Spezifikation“ als Technik die zur Realisierung der konkreten Praktik „Funktionale Anforderungen erheben“ dient. Dieser Abschnitt behandelt die Praktiken.

Ziel ist es mit Hilfe der Praktiken KMU in die Lage zu versetzen, ein Gefühl für ihre eigene Situation zu bekommen (Welche Praktiken beherrsche ich bis jetzt?) und zu sehen, welche Praktiken noch sinnvoll eingeführt werden sollten. Figure 2 gibt eine Übersicht über die einzelnen Praktiken.

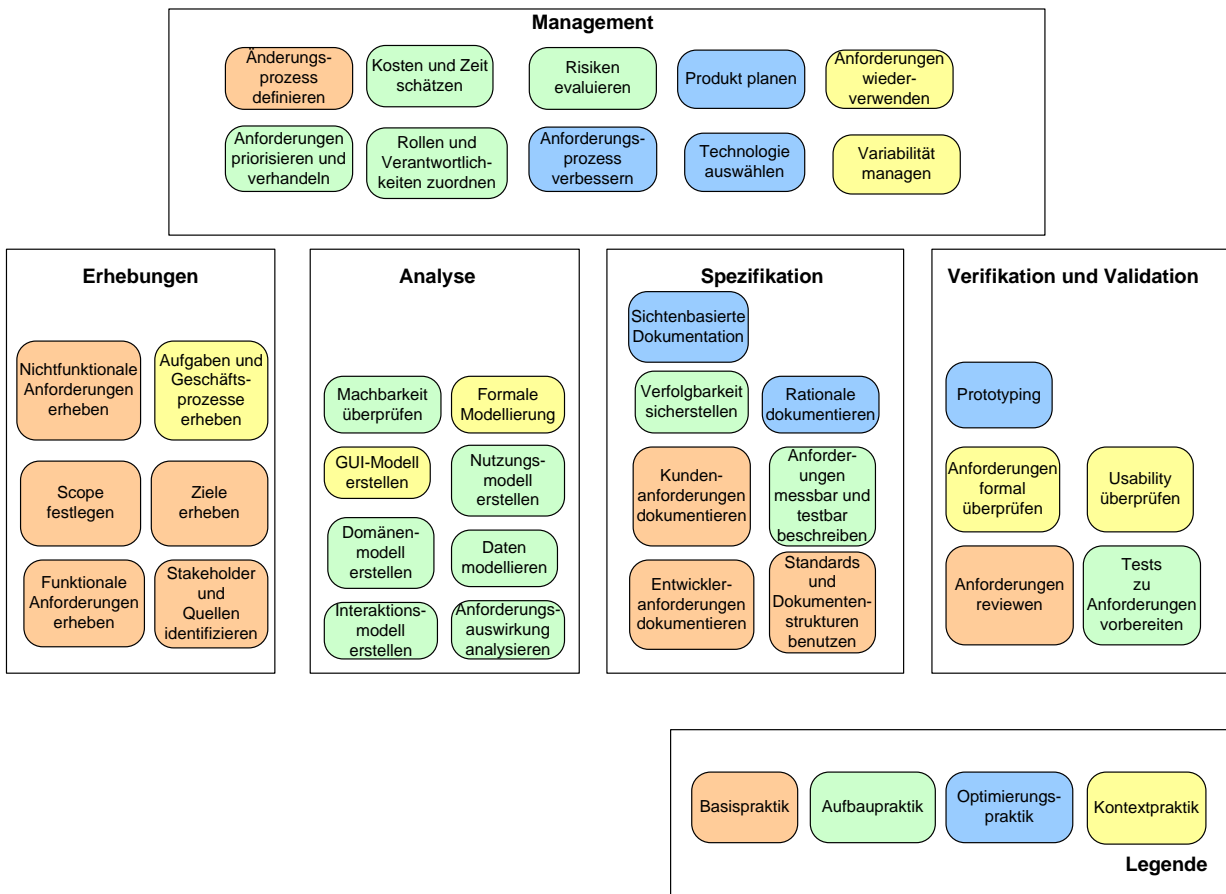


Figure 2 Übersicht von Praktiken

Die Praktiken sind nach ihrer Wichtigkeit klassifiziert:

- Basis-Praktiken: diese Praktiken sollen unabhängig vom Kontext immer beherzigt werden und versprechen einen hohen Nutzen.
- Aufbau-Praktiken: diese Praktiken können unabhängig vom Kontext eingesetzt werden, bauen jedoch auf anderen, meist Basis-Praktiken auf.

- Optimierungs-Praktiken: diese Praktiken sind per se nicht notwendig, bieten aber einen zusätzlichen, optimierenden Nutzen unabhängig vom Kontext.
- Kontext-Praktiken: diese Praktiken hängen vom konkreten Kontext der KMU und des Projekts ab.

Die Klassifizierung ist in Figure 3 dargestellt. Je nach Kontext wird die Kontextpraktik als Basis, Aufbau oder Optimierungspraktik eingestuft.

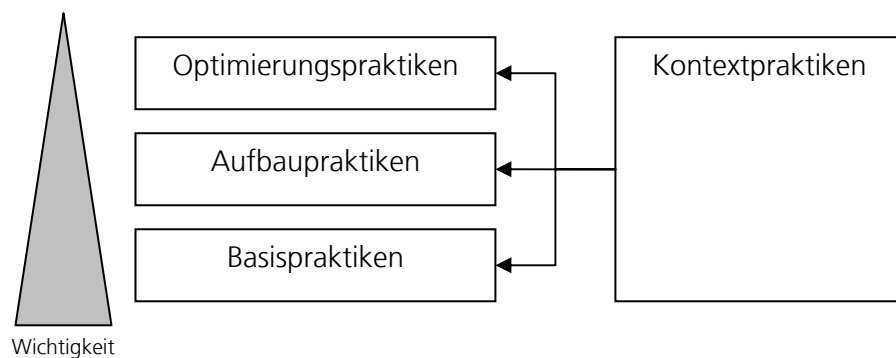


Figure 3

Praktikenklassifikation

Da die Praktiken zumeist einen bestimmten Teilbereich des Requirements Engineering adressieren, sind sie weiterhin den in 4.2 genannten Hauptaktivitäten zugeordnet (siehe Figure 2)

Die Beschreibung der einzelnen Praktiken, die als Zugang zu unserem Rahmenwerk dienen, ist leicht verständlich und kurz gehalten, so dass der Zugang schnell und einfach möglich ist. Zu jeder Praktik wird angegeben welches Ziel durch die Integration der Praktik im Rahmen des Anforderungsprozesses verfolgt wird. Eine übersichtliche Darstellung der Praktiken mit ihren zugeordneten Zielen ermöglicht somit eine schnelle Einschätzung welche Praktiken für ein Unternehmen sinnvoll sein können (siehe Figure 4). Diese Einschätzung ist vor allem auch Personen ohne detailliertes Anforderungswissen möglich.

Manchmal fällt es schwer zu entscheiden, ob eine Praktik eher eine konkrete Aktivität oder einen Grundsatz darstellt („Anforderungen testbar und messbar beschreiben“ könnte eine Subaktivität von „Entwickleranforderungen dokumentieren“ sein, oder ein Grundsatz, den man bei der Spezifikation berücksichtigen muss.) Eine pragmatische Vorgehensweise dabei ist, bereits etablierte Aktivitäten (Elicitation, Analysis) als Aktivitäten zu deklarieren und innovative bzw. nicht etablierte Ideen als zu berücksichtigende Grundsätze anzusehen. Praktiken können Kompetenzen sein, falls man das zugehörige Wissen (z.B. konkrete Techniken) dafür hat.

In Anhang A sind alle Praktiken beschrieben.

<b>Template zur Beschreibung einer Praktik</b>	
<b>Praktik:</b>	Name der Praktik
<b>Ziel:</b>	Ziel, das mit Einführung bzw. Erfüllen der Praktik erreicht wird.
<b>Beschreibung:</b>	Kurzbeschreibung der Praktik
<b>Phase im Framework:</b>	Anforderungserhebung, Anforderungsanalyse, Anforderungsspezifikation, Anforderungvalidierung oder Anforderungsmanagement
<b>Mögliche Techniken zur Realisierung der Praktik:</b>	<ul style="list-style-type: none"><li>• Technik1</li><li>• Technik2</li></ul>
<b>Praktiken, die vorher ausgeführt werden müssen:</b>	<ul style="list-style-type: none"><li>• Praktik1</li><li>• Praktik2</li></ul>
<b>Praktiken, die eine Unterstützung zur vorliegenden Praktik bieten:</b>	<ul style="list-style-type: none"><li>• Praktik1</li><li>• Praktik2</li></ul>
<b>Praktiken, die durch die vorliegende Praktik unterstützt werden:</b>	<ul style="list-style-type: none"><li>• Praktik1</li><li>• Praktik2</li></ul>
<b>Inkompatible Praktiken:</b>	<ul style="list-style-type: none"><li>• Praktik1</li><li>• Praktik2</li></ul>
<b>Nutzen:</b>	Basis-Praktik (von hohem Nutzen), Aufbau-Praktik (von mittlerem Nutzen), Optimierungs-Praktik (von zusätzlichem Nutzen) oder Kontext-Praktik (in einem bestimmten Kontext von Nutzen)
<b>Prinzip:</b>	Ja / Nein

Figure 4

Vorlage für eine Praktik

#### 4.4 Die Techniken

Den eher abstrakten Praktiken sind jeweils konkrete Techniken zugeordnet, mit denen man eine Praktik erfüllen, beziehungsweise umsetzen kann.

Die Techniken werden gemäß folgendem Template (siehe Figure 5) dokumentiert. Die Dokumentation sollte ebenfalls kurz sein, kann jedoch mehr als eine DinA4 Seite umfassen. Gegebenenfalls sollte im entsprechenden Unterpunkt der Beschreibung auf weiterführende Literatur verwiesen werden.

<b>Template zur Beschreibung einer Technik</b>	
<b>Technik:</b>	Name der Technik
<b>Ziel:</b>	Ziel, welches durch Nutzung der Technik erreicht werden soll
<b>Beschreibung:</b>	Beschreibung der Technik
<b>Ansprechpartner:</b>	Für weitere Informationen bezüglich der Technik steht Ihnen jederzeit XXX zur Verfügung.
<b>Anwendungskontext:</b>	Die Technik ist nur in dem XXX Kontext anwendbar /nicht spezifisch.
<b>Art / Typ:</b>	Methode, Prozess, Template, Algorithmus
<b>Erfüllt (zum Teil) folgende Praktiken:</b>	<ul style="list-style-type: none"><li>• Praktik 1</li><li>• Praktik 2</li></ul>
<b>Techniken, die vorher ausgeführt werden müssen:</b>	<ul style="list-style-type: none"><li>• Technik1</li><li>• Technik2</li></ul>
<b>Techniken, die eine Unterstützung zur vorliegenden Technik bieten:</b>	<ul style="list-style-type: none"><li>• Technik1</li><li>• Technik2</li></ul>
<b>Techniken, die durch die vorliegende Technik unterstützt werden:</b>	<ul style="list-style-type: none"><li>• Technik1</li><li>• Technik2</li></ul>
<b>Inkompatible Techniken:</b>	<ul style="list-style-type: none"><li>• Technik1</li><li>• Technik2</li></ul>
<b>Weiterführende Informationen zur Methodik:</b>	<ul style="list-style-type: none"><li>• Buch1</li><li>• Paper1</li><li>• Schulungsmaterial</li></ul>

Figure 5

Technikvorlage

## 5 Zusammenfassung

In diesem Bericht wurden das ReqMan Rahmenwerk, verwandte und zugrunde liegende Arbeiten sowie die wesentlichen Kernkonzepte vorgestellt.

Schwerpunkt des Rahmenwerks sind die Praktiken und Techniken. Die Praktiken stellen abstrakte Aufgaben dar, die in verschiedenen Kontexten Sinn machen. Eine Technik ist hingegen ein konkretes und durchführbares Vorgehen innerhalb des Anforderungsprozesses.

Aufgrund ihres abstrakten Charakters, können die Praktiken bei einer Bewertung des Ist-Standes innerhalb einer Organisation helfen. Die Techniken geben darauf aufbauend konkrete Umsetzung von Verbesserungsmaßnahmen.

## Literatur

- [Beech05] Beecham, S., Hall T., Britton C., Cottee M, Rainer A.: Using an expert panel to validate a requirements process improvement model. In: Journal of System and Software 76(3), 251-275. 2005
- [Emam95] EL EMAM KE and MADHAVJU NH (1995) A field study of requirements engineering practices in information systems development. In Proceedings of the International Symposium on Requirements Engineering.
- [Firesm05] Firesmith, D: Open Process Framework (OPF), [www.donald-firesmith.com](http://www.donald-firesmith.com), last visited 2005-07-20
- [Hall05] HALL T, BEECHAM S and RAINER A (2002) Requirements problems in twelve software companies: An empirical analysis. IEE Proceedings Software, 149(5), 153-160.
- [Juristo02] JURISTO N, MORENO AM and SILVA A (2002) Is the European Industry Moving towards Solving Re-quirements Problems. IEEE Software, 19(6), 70-77.
- [Kaindl02] KAINDL H, BRINKKEMPER S, BUBENKO J, FARBEY B, GREENSPAN S, et al. (2002) Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda. Requirements Engineering, 7(3), 113-123.
- [Kamst98] KAMSTIES E, HÖRMANN K and SCHLICH M (1998) Requirements Engineering in Small and Medium Enterprises. Requirements Engineering, 3(2), 84-90.
- [Krucht99] Kruchten, P.: The Rational Unified Process, An Introduction. Addison Wesley, 1999
- [Nikula00] NIKULA U, SAJANIEMI J and KÄLVIÄINEN H (2000) Management View on Current Requirements Engineering Practices in Small and Medium Enterprises. In Proceedings of The Australian Workshop on Requirements Engineering.
- [Rainer03] RAINER A, HALL T and BADDOO N (2003) Persuading developers to 'buy into' software process improvement: Local opinion and empirical evidence. In Proceedings of the International Symposium on Empirical Software Engineering.

[re-wissen] [www.re-wissen.de](http://www.re-wissen.de)

[Somm97] Sommerville, I., Sawyer, P.: Requirements Engineering: A Good Practice Guide. Wiley, 1997



## Anhang A: Praktikbeschreibungen

Name	Aktivität	Typ	Ziel
Ziele erheben	Erhebung	Basis	Ziel ist es, die Ziele der verschiedenen Stakeholder, die mit Hilfe eines Systems unterstützt werden können, festzulegen.
Scope festlegen	Erhebung	Basis	Ziel ist es, sicher zu stellen, dass nur die zur Erstellung eines Systems notwendigen Anforderungen im Rahmen des Projekts umgesetzt werden.
Funktionale Anforderungen erheben	Erhebung	Basis	Ziel ist es, die Funktionalen Anforderungen an ein System zu erheben, um eine optimale Unterstützung der Benutzenden bei der Erfüllung individueller Aufgaben zu ermöglichen.
Aufgaben und Geschäftsprozesse erheben	Erhebung	Kontext	Ziel ist es, die Geschäftsprozesse eines Unternehmens oder die Aufgaben einzelner Benutzer zu verstehen, um eine optimale Unterstützung mit der zu entwickelnden Software ermöglichen zu können.
Nichtfunktionale Anforderungen erheben	Erhebung	Basis	Ziel ist es, die Nichtfunktionalen Anforderungen zu ermitteln, um somit klare Qualitätsvorgaben an die durch ein System zu unterstützenden Geschäftsprozesse und Benutzeraufgaben umsetzen zu können.
Stakeholder und Quellen identifizieren	Erhebung	Basis	Ziel ist es, alle möglichen Stakeholder und Informationsquellen, die eine direkte oder indirekte Verbindung zu einem System oder zur Systementwicklung aufweisen, in den Anforderungsprozess zu integrieren, um eine möglichst vollständige Erfassung der Anforderungen zu gewährleisten.
Domänenmodell erstellen	Analyse	Aufbau	Ziel ist es, die Anwendungsdomäne zu analysieren und zu verstehen.
Machbarkeit überprüfen	Analyse	Aufbau	Ziel ist es, sicherzustellen, dass alle Anforderungen an ein System erfüllt werden können.
Daten modellieren	Analyse	Aufbau	Ziel ist es, die relevanten Daten der Anwendungsdomäne zu analysieren und zu verstehen.
Anforderungsauswirkungen analysieren	Analyse	Aufbau	Ziel ist es, die Auswirkungen von Anforderungen und Anforderungsänderungen auf andere Anforderungen sowie die Systemarchitektur abschätzen zu können.

Name	Aktivität	Typ	Ziel
Interaktionsmodell erstellen	Analyse	Aufbau	Ziel ist es, alle möglichen Interaktionen zwischen Benutzern und einem System oder zwischen einem System und anderen Systemen zu verstehen.
GUI-Modell erstellen	Analyse	Kontext	Ziel ist es, die graphische Benutzerschnittstelle eines Systems im Hinblick auf eine hohe Gebrauchstauglichkeit zu entwerfen.
Formal modellieren	Analyse	Kontext	Ziel ist es, Anforderungen so zu modellieren, dass eine mathematische Überprüfung, aber auch Ableitung in spätere Softwareprodukte ermöglicht werden kann.
Nutzungsmodell erstellen	Analyse	Aufbau	Ziel ist es, das Nutzungsverhalten der Benutzer zu modellieren.
Kundenanforderungen dokumentieren	Spezifikation	Basis	Ziel ist es, die Anforderungen der Kunden zu dokumentieren.
Entwickleranforderungen dokumentieren	Spezifikation	Basis	Ziel ist es, die Anforderungen aus Entwicklersicht zu dokumentieren, um so eine optimale Kommunikation mit den Entwicklern bezüglich der Anforderungen zu ermöglichen.
Sichtenbasierte Dokumentation	Spezifikation	Optimierung	Ziel ist es, eine effiziente und effektive Dokumentation der Anforderungen zu ermöglichen.
Anforderungen messbar und testbar beschreiben	Spezifikation	Aufbau	Ziel ist es, Anforderungen überprüfbar zu spezifizieren.
Rationale dokumentieren	Spezifikation	Optimierung	Ziel ist es, den Ursprung und die Begründung für Anforderungen zu dokumentieren, um diese besser verstehen und in ihrer Intention nachvollziehen zu können.
Verfolgbarkeit sicherstellen	Spezifikation	Aufbau	Ziel ist es, die Verfolgbarkeit von Anforderungen sicherzustellen, um deren Verfeinerung / Realisierung in späteren Entwicklungsprodukten nachvollziehen zu können.
Standards und Dokumentenstruktur benutzen	Spezifikation	Basis	Ziel ist es, durch Verwendung einer Dokumentenstruktur und Erfüllung von Dokumenten-Standards die Qualität und insbesondere die Verständlichkeit und Vollständigkeit eines Anforderungsdokuments zu erhöhen.
Usability überprüfen	Verifikation und Validation	Kontext	Ziel ist es, die Gebrauchstauglichkeit eines Systems sicherzustellen.
Anforderungen revidieren	Verifikation und Validation	Basis	Ziel ist es, die Anforderungen systematisch zu inspizieren, um die Qualität eines Systems bereits in frühen Phasen der Softwareentwicklung sicherzustellen.

Name	Aktivität	Typ	Ziel
Tests zu Anforderungen vorbereiten	Verifikation und Validation	Aufbau	Ziel ist es, Testfälle für die Testphase erstellen.
Anforderungen formal überprüfen	Verifikation und Validation	Kontext	Ziel ist es, Anforderungen mit Hilfe formaler, mathematischer Techniken zu überprüfen, um dadurch insbesondere Konsistenz und Vollständigkeit sicherstellen zu können.
Prototyping	Verifikation und Validation	Optimierung	Ziel ist es, eines spätere System oder Teile eines späteren Systems (z.B. Benutzerschnittstelle) exemplarisch zu implementieren, um hiermit frühzeitig Qualitätseigenschaften überprüfen und die Kommunikation mit den Stakeholdern unterstützen zu können.
Rollen und Verantwortlichkeiten zuordnen	Management	Aufbau	Ziel ist es, den Anforderungsprozess systematisch zu strukturieren, um einen reibungslosen Ablauf in überprüfbarer und optimierbarer Form zu gewährleisten.
Kosten und Zeitschätzen	Management	Aufbau	Ziel ist es, möglichst zutreffende Informationen über die Kosten, Aufwände und benötigten Zeitrahmen, die mit einem Projekt verbunden sind, zu erhalten, um Angebote erstellen oder das finanzielle Risiko bei den Vertragsverhandlungen minimieren zu können.
Anforderungsprozess verbessern	Management	Optimierung	Ziel ist es, Probleme und Verbesserungspotential bezüglich des Anforderungsprozesses in einem Unternehmen aufzudecken, um durch Lösung der Probleme die Qualität des Anforderungsprozesses zu steigern.
Risiken evaluieren	Management	Aufbau	Ziel ist es, Probleme und ihre Auswirkungen, die während der Umsetzung einer Anforderung auftreten können, zu ermitteln, um optimale Vorbereitungen zur Bewältigung der daraus resultierenden Risiken treffen zu können.
Produkte planen	Management	Optimierung	Ziel ist es, die Produkte in Bezug auf die zu realisierenden technischen und organisatorischen Anforderungen zu planen, um die Entwicklungsprojekte auf diese Vorgaben optimieren zu können.
Variabilität managen	Management	Kontext	Das Ziel ist es, in einer Umgebung mit mehreren ähnlichen Produkten, Gemeinsamkeiten und Variabilitäten zu managen, um das Wiederverwendungspotential besser und effizienter ausschöpfen zu können.
Technologie auswählen	Management	Optimierung	Ziel ist es, systematisch Techniken, Werkzeuge und Vorgehensweisen auszuwählen, um die Aktivitäten des Anforderungsmanagements kontextoptimal durchführen zu können.

Name	Aktivität	Typ	Ziel
Anforderungen priorisieren und verhandeln	Management	Aufbau	Ziel ist es, die von den einzelnen Stakeholdern erwünschten Anforderungen zu analysieren und zu priorisieren, um die tatsächlich zu entwickelnden Anforderungen an das System festlegen zu können.
Änderungsprozess definieren	Management	Basis	Ziel ist es, durch die explizite Definition eines an das Unternehmen angepassten Änderungsprozesses die zuverlässige und erfolgreiche Berücksichtigung von Änderungswünschen zu ermöglichen.
Anforderungen wiederverwenden	Management	Kontext	Ziel ist es, Anforderungen wiederzuverwenden, um in ähnlichen Systemen unnötigen Mehrfachaufwand für Erhebung und Spezifikation zu reduzieren.

# Dokumenten Information

Titel: Das ReqMan Prozessrahmenwerk

Datum: 1. Oktober 2006

Report: IESE-141.06/D

Status: Final

Klassifikation: Öffentlich

Copyright 2006, Fraunhofer IESE.

Alle Rechte vorbehalten. Diese Veröffentlichung darf für kommerzielle Zwecke ohne vorherige schriftliche Erlaubnis des Herausgebers in keiner Weise, auch nicht auszugsweise, insbesondere elektronisch oder mechanisch, als Fotokopie oder als Aufnahme oder sonstwie vervielfältigt, gespeichert oder übertragen werden. Eine schriftliche Genehmigung ist nicht erforderlich für die Vervielfältigung oder Verteilung der Veröffentlichung von bzw. an Personen zu privaten Zwecken.