

# Wie viel Requirements Engineering steckt im Software Engineering?

Ralf Fahney

Freier Berater für  
Anforderungsingenieurwesen  
und Projektmanagement  
Helfrichstrasse 8  
D-82041 Oberhaching  
rf@fahney.com

Andrea Herrmann

Universität Heidelberg  
Institut für Informatik  
Neuenheimer Feld 326  
D-69120 Heidelberg  
andrea.herrmann@informatik.uni-heidelberg.de

Rüdiger Weißbach

IS-Manager bei einem  
Finanzdienstleister und  
Lehrbeauftragter an der  
Fachhochschule Hamburg  
D-22119 Hamburg  
r.weissbach@sh-home.de

**Abstract:** Die AutorInnen, Informatiker mit zusammen mehr als 40 Jahren Berufserfahrung in IT-Projekten, vertreten die These, dass das, was man üblicherweise als Software Engineering bezeichnet, auch in eher programmnahen Bereichen zu einem nicht unerheblichen Teil aus Requirements Engineering besteht. Sie begründen die These anhand gängiger Ergebnistypen des Software Engineering sowie den Rollenbeschreibungen des V-Modell XT. Als Konsequenz regen die AutorInnen an, die im Requirements Engineering üblichen Verfahren auch dort im Software Engineering anzuwenden, wo man sie bisher eher weniger vermutet, nämlich gerade in programmnahen Bereichen.

## 1 Was ist Requirements Engineering?

Literaturrecherchen wie z.B. in [HFRW05] fördern zu Tage, dass es keine allgemein anerkannte Definition davon gibt, was man unter Requirements Engineering (deutsch „Anforderungsingenieurwesen“) versteht. In Anlehnung an und Erweiterung von [VSH01] verstehen die AutorInnen unter Requirements Engineering,

... Anforderungen an ein Produkt, einen Service, ein Ergebnis,  
ein Projekt, einen Geschäftsprozess, ein was-auch-immer...

zu ermitteln  
aufzunehmen

**zu verstehen**

zu klären  
zu kommunizieren  
zu diskutieren  
wegzudiskutieren

abzuwägen  
abzulehnen  
einzufordern  
einzuklagen  
zu dokumentieren  
zu vereinbaren  
anzupassen  
zu verwalten  
zu pflegen

**im Groben wie im Detail**

**innerhalb und außerhalb von  
Projekten**

**unabhängig von der Art des  
Projektes**

Wer sich in einem Projekt damit beschäftigt, das richtige Projekt zu machen, arbeitet im Fachgebiet Requirements Engineering. Der ideale Anforderungsingenieur sollte demnach erfahren sein in

Kommunikationswissenschaften  
Psychologie  
Jura  
Buchhaltung

Die Bedeutung sozialer bzw. psychologischer Faktoren in der Projektarbeit ist allgemein anerkannt, was in Schlagworten wie „Menschen machen Projekte“ (z.B. als Leitmotiv der Tagung „Informatik2006“<sup>1</sup>) oder in dem Buchtitel „Peopleware“ von De Marco und Lister zum Ausdruck kommt. Die Relevanz von Jura und Buchhaltung mag hingegen auf den ersten Blick irritieren.

Die Bedeutung von Jura resultiert gleichermaßen aus der juristischen Komponente, die (zumindest größeren) Projekten innewohnt - die Spezifikation wird in der Regel Vertragsbestandteil - als auch aus der analytischen, „unparteiischen“ Funktion, die Anforderungsingenieure gegenüber den Anforderungen einnehmen.

Warum Buchhaltung? Ein Anforderungsingenieur sollte nach Auffassung der AutorInnen seine Anforderungen genauso sorgfältig verwalten und mit ihnen umgehen wie ein Buchhalter mit seinen Zahlen. *Aber Vorsicht:* Mit einem Buchhalter verbindet man auf den ersten Blick eher seine betriebswirtschaftlichen Kenntnisse. Solche zu besitzen, schadet einem Anforderungsingenieur nicht. Zur Untermauerung der zentralen These dieses Beitrags dient „Buchhaltung“ jedoch zur Verdeutlichung der Sorgfalt, welche ein Anforderungsingenieur in seine Tätigkeit einbringen sollte.

Warum ist Software Engineering in den Kompetenzen nicht aufgezählt? Weil man für die Konstruktion eines Herzschrittmachers oder den Bau eines Schiffes *Software Engineering* nur dort benötigt, wo es um Software geht. Und dann hat man es wieder mit einem IT-Projekt zu tun, welches z.B. als Teilprojekt der Entwicklung des Herzschrittmachers oder des Schiffes durchgeführt wird. Requirements Engineering hingegen benötigt man bei derartigen Projekten auch dort, wo es um Hardware geht (z.B. die Eigenschaften der Umhüllung eines Herzschrittmachers)

In dieser Sichtweise ist Software Engineering die spezifische Lösungskompetenz in IT-Projekten, so wie Bauingenieurwesen die Lösungskompetenz in Bauprojekten ist, die Elektrotechnik in Halbleiterprojekten oder die Luft- und Raumfahrttechnik in der Flugzeugentwicklung. Abbildung 1 verdeutlicht diesen Zusammenhang und bringt gleichzeitig den Bezug zu Projektmanagement, Stakeholding und Lobbying ins Spiel. Details dazu folgen später, auch zur Einordnung des Qualitätsmanagements, das nicht ausdrücklich als querschnittliches Fachgebiet erwähnt ist. Die Autoren gehen auf diesen Umstand später in diesem Beitrag noch ausführlich ein. Auch die Begriffe „Stakeholding“ und „Lobbying“ werden später noch definiert.

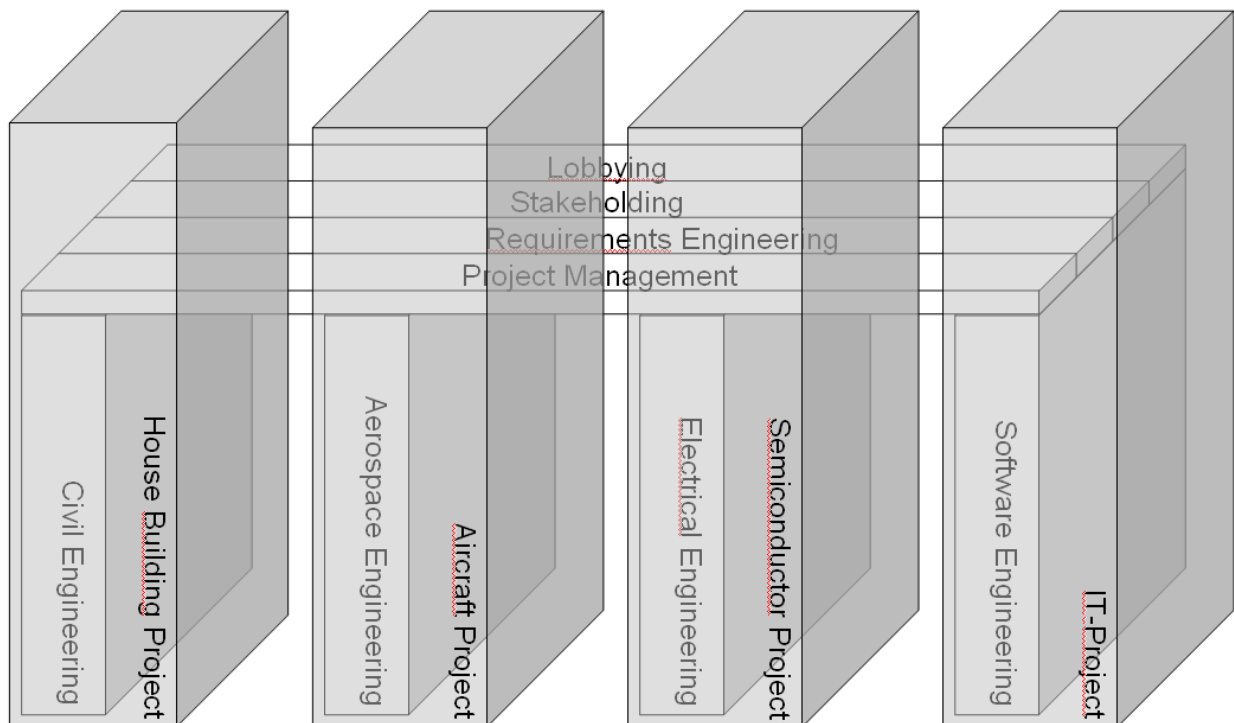


Abbildung 1: Beziehung zwischen Requirements Engineering und Lösungskompetenz

Man erkennt, dass es zwischen Requirements Engineering und den jeweiligen Lösungskompetenzen eine Schnittstelle gibt. Jedes Fachgebiet verwendet an dieser Schnittstelle seine eigenen Modelle. In einem

<sup>1</sup> <http://www.informatik2006.de/252.html>, abgerufen am 13.04.2007

Halbleiterprojekt sind es u.a. Pläne mit dem Chip-Layout. Im IT-Bereich sind es z.B. Use Cases und UML-Diagramme.

## 2 Ergebnistypen in IT-Projekten und ihr Bezug zu Anforderungen

Ergebnistypen können als Produkte einzelner Aktivitäten begriffen werden. Die folgende Tabelle enthält typische Ergebnistypen aus einem IT-Projekt:

|  |   |
|--|---|
| Projektauftrag                             | = Vereinbarung mit dem Auftraggeber                                       |
| Stakeholderliste                           | = Analyse der Betroffenen   |
| Geschäftsprozessmodell und Kontextdiagramm | = Analyse des Kontextes, in welchem das IT-System zum Einsatz kommen soll |
| Glossar                                    | = Spezifikation von Begriffen   |
| System-, HW- und SW-Spezifikation          | = Anforderungen an das IT-System  |
| Schnittstellenspezifikation                | = Vereinbarung mit einem Nachbarsystem                                    |
| Komponentenspezifikation                   | = Anforderungen an eine Komponente  |
| Dialogspezifikation                        | = Anforderungen an einen Dialog   |
| Modulspezifikation                         | = Anforderungen an ein Modul  |
| Change Request                             | = Anforderung zur Änderung von Anforderungen                              |
| Spezifikation einer Plausibilitätsprüfung  | = Anforderungen an ein Maskenfeld   |
| Klassendiagramm                            | = Anforderungen an eine oder mehrere Klassen                              |
| Anforderungen an Nachbarabteilungen        | z.B. hinsichtlich der zukünftigen Organisation                            |
| Anforderungen an das Projekt               | z.B. Einsatz eines bestimmten Vorgehens                                   |
| Style Guide                                | = Anforderungen an die Gestaltung der Benutzeroberfläche                  |
| Projektplan                                | = terminliche Anforderungen an die Arbeitspakete                          |

Jeder erwähnte Ergebnistyp dokumentiert Anforderungen. Selbstverständlich dokumentiert z.B. der Projektauftrag eine andere Art von Anforderungen als eine Modulspezifikation. Über eine Modulspezifikation sprechen andere Personen miteinander als über einen Projektauftrag. Und man spricht über einen Projektauftrag auch zu einem ganz anderen Zeitpunkt im Projekt als über eine Modulspezifikation. Trotzdem handelt es sich in diesen beiden und auch in den anderen Fällen um eine **Dokumentation von Anforderungen**.

Diese Beobachtung führt zu folgender These:

**Das, was man üblicherweise als Software Engineering bezeichnet, besteht zu einem nicht unerheblichen Teil aus Requirements Engineering.**

Die folgenden Ergebnistypen dokumentieren keine Anforderungen:

|              |   |
|--------------|---|
| Programmcode | = Ausführbare Befehle an die Hardware <sup>2</sup>                  |
| Testergebnis | = Das Ergebnis der Ausführung der Testfälle und deren Vergleich mit |

<sup>2</sup> Obwohl man, würde man es auf die Spitze treiben wollen, bei einem Programmcode auch von Anforderungen sprechen könnte, welche der Computer erfüllen soll, sozusagen dem Vertrag zwischen dem Entwickler und dem Computer. Eine vergleichbar konsequente terminologische Zuspitzung finden wir beispielsweise in der (technischen) Kommunikation von Objekten untereinander. Ein Programmcode ist jedoch keine Anforderung an einen anderen menschlichen Akteur.

|                    |   |
|--------------------|---|
|                    | den Anforderungen   |
| Aufwandsschätzung  | = Eine Schätzung des Aufwands für die Durchführung einer Aktivität, bevor sie durchgeführt wird   |
| Designentscheidung | = Entscheidung für eine von mehreren möglichen Lösungsmöglichkeiten, als Ergebnis einer Bewertung der Lösungsmöglichkeiten in Bezug auf deren Erfüllung der Anforderungen |

Es fällt nicht leicht, weitere Ergebnistypen zu finden, an deren Erstellung zwingend Softwareingenieure beteiligt sind und die keine Anforderungen dokumentieren.

Dass Software Engineering nun doch etwas deutlich anderes ist als Requirements Engineering, soll die folgende Definition aufzeigen, welche den Begriff „Designentscheidung“ aufgreift:

(Software-)Design ist ...

- ... die kreative Leistung,
- mit der eine Menge von Anforderungen an eine Lösung (z.B. eine Systemspezifikation)
- um eine geeignete zweite Menge von Anforderungen ergänzt wird, (z.B. mehrere Modulspezifikationen zusammen mit der Lösungsbeschreibung. Die Lösungsbeschreibung wird nach der Entscheidung für die gewählte Lösung zur Rahmenbedingung des Projektes und ist deswegen in sich selbst eine Anforderung)
- mit welcher die gemäß der ersten Menge möglichen Lösungen sinnvoll weiter eingeschränkt werden, (würde man andere Modulspezifikationen wählen, sähe die Lösung anders aus)
- ... sowie die Dokumentation der Entscheidungen, welche gerade zu dieser zweiten und nicht zu einer anderen zweiten Menge von Anforderungen geführt haben. (z.B. das Designdokument)

Um die „zweite Menge“ geeignet festlegen zu können, benötigt man Lösungskompetenz. Ein Marketing-Designer würde sich z.B. vermutlich eher schwer tun, die geeigneten Software-Module für ein IT-System zur Unterstützung der Vertriebskanalsteuerung einer Luftfahrtgesellschaft zu bestimmen. Dies ist die Domäne des Software Engineering.

Weshalb könnte es nützlich sein, diese anforderungs-orientierte Sicht auf Software Engineering einzunehmen? Weil man dann zu jedem Zeitpunkt klar darüber ist, wo man es mit Vereinbarungen, Verträgen und Ähnlichem zu tun hat und etablierte Verfahren des Requirements Engineering überall dort einsetzen kann, wo sie Nutzen stiften, z.B.

- Verständigung über die Anforderungen auch in programmnahen Bereichen, z.B. durch Vereinbarungen zwischen einem IT-Teamleiter und seinen Softwareingenieuren, wobei die Softwareingenieure ihren Teamleiter als Stakeholder ansehen sollten.
- Dokumentation der Anforderungen widerspruchsfrei, konsistent, unmissverständlich
- Verknüpfung von Anforderungen durch Beziehungen zum Zwecke der Nachvollziehbarkeit und Unterstützung von Impact-Analyse bei Changes.

Somit wird man zu jedem Zeitpunkt sicher sein, das richtige Projekt zu machen.

### 3 Anforderungen im V-Modell XT

Die HTML-Fassung der Version 1.2 des V-Modell XT [VMXT06] besteht aus 1203 HTML-Dokumenten. 427 davon, das sind 35%, enthalten den Begriff „Anforderung“. 143 enthalten den Begriff „Vertrag“. Somit geht es bei

einem Drittel bis der Hälfte der HTML-Dokumente mindestens auch, wenn nicht sogar überwiegend um Anforderungen, Vereinbarungen und Verträge.

Für den Ergebnistyp „SW-Spezifikation“ des V-Modell XT ist die Rolle „SW-Architekt“ verantwortlich. Mitwirkend sind SW-Entwickler, Logistikentwickler, Ergonomieverantwortlicher, Prüfer und Systemsicherheitsbeauftragter. Folgende Zitate aus [VMXT06] zeigen auf, wie ausführlich es bei der Erstellung der „SW-Spezifikation“ um Anforderungen geht:

„Die SW-Spezifikation beschreibt alle funktionalen und nicht-funktionalen Anforderungen an ein SW-Element (SW-Einheit, SW-Komponente oder SW-Modul).“

Es fällt auf, dass nicht nur das System als Ganzes, sondern auch die Teile davon mit SW-Spezifikationen beschrieben werden.

„Zur Erstellung der Spezifikation werden die Anforderungen aus den Spezifikationen übergeordneter Systemelemente beziehungsweise SW-Elemente abgeleitet.“

Die Ableitung ist die „geeignete Festlegung der zweiten Menge von Anforderungen“ gemäß der Definition von „Software-Design“ im vorherigen Kapitel.

„Wesentliche Inhalte der SW-Spezifikation sind die Beschreibung der Anforderungen an das SW-Element sowie die Festlegung der Schnittstellen, die es zu bedienen hat. Zusätzlich wird die Verfeinerung und Zuordnung von Anforderungen und Schnittstellen zu untergeordneten SW-Elementen beschrieben. Im Rahmen der Anforderungsverfolgung wird sichergestellt, dass alle Anforderungen an das Element bei der Verfeinerung auf die nächste Hierarchieebene berücksichtigt werden.“

Um dies zu leisten, wendet der SW-Architekt etablierte Verfahren des Requirements Engineering an.

„Die Erstellung der SW-Spezifikationen erfolgt Hand in Hand mit dem Architektorentwurf der SW-Einheiten. Zur Sicherstellung der Konsistenz zwischen Spezifikationen und Architektur ist der SW-Architekt verantwortlich für die Erstellung beider Produkte.“

Dieses Zitat betont wieder den Designaspekt in der Tätigkeit des SW-Architekten.

Der Arbeitskreis „Requirements Engineering & Projektmanagement“ der GI-Fachgruppe „Requirements Engineering“ hat fünf Fachgebiete identifiziert, welche zu differenzieren sind, um die Beziehung zwischen Requirements Engineering und Projektmanagement zu untersuchen [GIREPM05]<sup>3</sup>. Vorsicht: Diese Fachgebiete sind nicht zwangweise deckungsgleich mit den in ihnen arbeitenden Personen und auch nicht mit Rollen, welche diese Personen in einem Projekt innehaben:

|                                |   |
|--------------------------------|---|
| SH – Stakeholding              | Im Fachgebiet Stakeholding zu arbeiten bedeutet, relevante und verlässliche Entscheidungen zu treffen. Ein Beispiel dafür sind sämtliche Entscheidungen, welche den Projektumfang betreffen   |
| LO - Lobbying                  | Im Fachgebiet Lobbying zu arbeiten bedeutet, Einfluss auf den Projektumfang zu nehmen oder sich um Einflussnahme zu bemühen unabhängig davon warum man dies tut. Ein Vertriebsmitarbeiter, der sich um Beauftragung eines Projektes bemüht, betreibt dabei genauso Lobbying wie ein Angestellter, der heimlich ein Projekt boykottiert, weil er fürchtet, sein Arbeitsplatz könne mit Erreichen der Projekteziele überflüssig werden. |
| PM - Project Management        | Im Fachgebiet Projektmanagement zu arbeiten bedeutet, ein Projekt oder Aktivitäten in einem Projekt zu planen, zu führen, zu koordinieren, zu steuern und zu kontrollieren.   |
| RE – Requirements Engineering: | Siehe Definition in Kapitel 1. Mit dieser Definition wird deutlich, das RE den Projektumfang dokumentiert und managet. RE ist der “single point of entry,, des Projektes für jegliche Art von Information, welche den Projektumfang betrifft  |

<sup>3</sup> Mit diesen 5 Fachgebieten soll lediglich die Schnittstelle zwischen Projektmanagement und Requirements Engineering beschrieben werden; daher sind sie in Bezug auf Anforderungen detaillierter aufgeschlüsselt, während das Solution Engineering vielfältige Aktivitäten umfasst wie Design, Realisierung und Qualitätssicherung. In der hier vorgestellten Form können und sollen sie nicht dazu dienen, ein vollständiges Projektmodell zu definieren.

SE - Software Engineering  
 - Solution Engineering:

Im Fachgebiet Software Engineering (als konkrete Form von Solution Engineering) zu arbeiten bedeutet, eine (IT-)Lösung zu entwerfen und zu implementieren, welche die Anforderungen erfüllt. Aus Sicht der Autoren umfasst dies auch Aufwandsschätzung und Qualitätsmanagement.

Der Arbeitskreis untersuchte die Rollen des V-Modell XT daraufhin, mit welchem Anteil jede Rolle in den erwähnten fünf Fachgebieten tätig werden muss. Vier AK-Mitglieder haben ausgehend von ihren individuellen Erfahrungen die Festlegungen des V-Modell XT geprüft und eine Schätzung abgegeben. Daraus hat der AK pro Rolle und Fachgebiet Mittelwert und Standardabweichung errechnet. Die folgende Tabelle zeigt für ausgewählte Rollen die Mittelwerte pro Fachgebiet. Die letzte Spalte enthält pro Rolle die Summe über die Standardabweichungen.

Bei besonders hohen Standardabweichungen haben die Mitglieder des AK ihre individuellen Bewertungen gegenseitig begründet, um Missverständnisse in der Interpretation der im V-Modell XT dokumentierten Festlegungen nach Möglichkeit auszuschließen. Dieser Abstimmungsprozess ist noch nicht vollständig abgeschlossen. Trotzdem gibt der AK einen Zwischenstand bekannt und versteht die verbleibende Standardabweichung als Maß für den im V-Modell XT 1.2 vorhandenen Interpretationsspielraum:

|                              | SH | PM | RE | SE | LO | $\sigma$ |
|------------------------------|----|----|----|----|----|----------|
| Akquisiteur                  |    | 11 | 15 | 8  | 66 | 19       |
| Änderungsverantwortlicher    |    | 14 | 36 | 50 |    | 33       |
| Anforderungsanalytiker (AG)  | 14 | 9  | 70 | 5  | 3  | 32       |
| Anforderungsanalytiker (AN)  |    | 10 | 75 | 15 |    | 15       |
| Anwender                     | 81 |    | 16 | 1  | 1  | 16       |
| Ergonomieverantwortlicher    | 10 | 8  | 40 | 35 | 8  | 37       |
| Projektkaufmann              |    | 74 | 6  | 18 | 3  | 49       |
| Projektleiter                |    | 65 | 16 | 11 | 8  | 19       |
| Projektmanager               | 3  | 66 | 11 | 5  | 15 | 35       |
| SW-Architekt                 |    | 14 | 16 | 70 |    | 27       |
| SW-Entwickler                |    | 9  | 9  | 83 |    | 7        |
| Systemicherheitsbeauftragter | 8  | 20 | 45 | 25 | 3  | 49       |

Es fällt auf, dass SW-Architekt und SW-Entwickler zu einem nennenswerten Anteil im Fachgebiet Requirements Engineering tätig sind. Bezogen auf die Rollen HW-Architekt, HW-Entwickler, Logistikverantwortlicher und Logistikentwickler kam der Arbeitskreis zu ähnlichen Einschätzungen.

## 4 Zusammenfassung

Anhand gängiger Ergebnistypen aus dem Software Engineering haben die AutorInnen aufgezeigt, dass das, was man üblicherweise als Software Engineering bezeichnet, zu einem erheblichen Teil aus Requirements Engineering besteht. Die Analyse des V-Modell XT untermauert diese These.

Die aufgezeigten Sachverhalte legen daher die Vermutung nahe, dass es erheblich zur Absicherung des Projekterfolgs beiträgt, wenn Softwareingenieure mit den Methoden und Verfahren des Requirements Engineering vertraut sind und die in der Einführung genannten Erfahrungen mitbringen. Zum einen erleichtert die Kenntnis anderer Arbeitsbereiche die Zusammenarbeit. Man vermeidet z.B. Kommunikationsprobleme, die durch unterschiedliche Terminologien und Semantiken entstehen können. Zum anderen aber gilt dies auch gerade in den Fällen, in denen Softwareingenieure nicht ausdrücklich als Anforderungsingenieur arbeiten, der in Zusammenarbeit mit dem Projektauftraggeber und anderen Stakeholdern die Benutzeranforderungen aufnimmt und das Pflichtenheft erstellt. Es hat sich gezeigt, dass klare und unmissverständliche Formulierungen und Vereinbarungen generell in einem Projekt wichtig sind. Dies gilt sowohl für lineare Vorgehensmodellen als auch für agile, in denen die Eindeutigkeit der Vereinbarungen im Dialog der Akteure erzeugt werden muss.

## Literaturverzeichnis

- [HFRW05] Herrmann, A., Fahney, R., Rückert, C., Weißbach, R.: Clear Role and Process Definitions as a Means to Analyze and Understand Conflicts between Project Management and Requirements Engineering. Beitrag zum "Workshop on the Interplay of Requirements Engineering and Project Management in Software Projects", der 13<sup>th</sup> IEEE Requirements Engineering conference, 2005.
- [GIREPM05] Arbeitskreis „Requirements Engineering & Projektmanagement“ der Fachgruppe 2.1.6 „Requirements Engineering“ der Gesellschaft für Informatik. [www.repm.de](http://www.repm.de)

- [VMXT06] V-Modell XT Version 1.2  
<http://ftp.uni-kl.de/pub/v-modell-xt/Release-1.2/CD/VMXT-CD-R1.2.iso>
- [VSH01] Versteegen, G., Salomon, K., Heinold, R.: Change Management bei Software Projekten. Springer-Verlag, Berlin, Heidelberg, New York, 2001.