

# eCH-0120 Hilfsmittel für Requirements Engineering & Management sowie plattform-unabhängige Modellierung

<b>Name</b>	Hilfsmittel für Requirements Engineering & Management sowie plattformunabhängige Modellierung
<b>Standard-Nummer</b>	eCH-0120
<b>Kategorie</b>	Hilfsmittel
<b>Reifegrad</b>	Definiert
<b>Version</b>	1.00
<b>Status</b>	Aufgehoben
<b>Genehmigt am</b>	2010-11-25
<b>Ausgabedatum</b>	2014-09-03
<b>Ersetzt Standard</b>	
<b>Sprachen</b>	Deutsch
<b>Autoren / Kontakt</b>	<b>Fachgruppe HERMES, Arbeitsgruppe REQMOD</b> Kontakt: Ralf Fahney, Fahney Anforderungsingenieurwesen GmbH, rf@fahney.com
<b>Herausgeber / Vertrieb</b>	Verein eCH, Mainaustrasse 30, Postfach, 8034 Zürich T 044 388 74 64, F 044 388 71 80 www.ech.ch / info@ech.ch

## Zusammenfassung

Dieses Hilfsmittel dokumentiert stichwortartig in der Praxis erprobte Verfahren, die das Vorgehen im Requirements Engineering & Management und in der plattformunabhängigen Modellierung systematisieren. Die Anwendung dieser Verfahren sichert Projekte ab, indem sie Transparenz und Klarheit des Projektumfangs fördert und dessen Abstimmung und Vereinbarung zwischen allen Stakeholdern unterstützt.

Die Anwendung dieser Verfahren garantiert nicht, dass ein spezifizierter Funktionsumfang dem tatsächlichen Bedarf entspricht. Allerdings erhöht die Anwendung dieser Verfahren erheblich die Wahrscheinlichkeit, dass die Spezifikation den tatsächlichen Bedarf beschreibt.

Die Autoren empfehlen, dass Projektleiter dieses Hilfsmittel als Checkliste für die Prüfung nutzen, inwieweit ihr Projekt die Erfolgsfaktoren auch tatsächlich nutzt.

## Inhaltsverzeichnis

<b>1</b>	<b>Status des Dokuments.....</b>	<b>3</b>
<b>2</b>	<b>Einleitung.....</b>	<b>3</b>
2.1	Überblick .....	3
2.2	Anwendungsgebiet .....	12
2.3	Vorteile .....	12
2.4	Schwerpunkte.....	12
<b>3</b>	<b>Best Practice für Requirements Engineering &amp; Management sowie plattformunabhängige Modellierung .....</b>	<b>13</b>
3.1	<sup>[ANF006]</sup> Planen Sie Ihr Requirements Engineering & Management .....	13
3.2	<sup>[ANF007]</sup> Anforderungen verwalten: Betrachten Sie jedes (jedes!) Ergebnis als eine Anforderungsspezifikation .....	13
3.3	<sup>[ANF009]</sup> Anforderungen verwalten: Unterscheiden Sie Ergebnisse auf verschiedenen Abstraktionsebenen .....	14
3.4	<sup>[ANF012]</sup> Anforderungen ermitteln/erheben: Legen Sie die Ermittlungstechniken ausdrücklich fest und wenden Sie diese dann auch an.....	14
3.5	<sup>[ANF008]</sup> Anforderungen verstehen und dokumentieren: Modellieren und spezifizieren Sie formal .....	14
3.6	<sup>[ANF011]</sup> Anforderungen verwalten: Dokumentieren sie alle Ergebnisse in einem zentralen Repository.....	15
3.7	<sup>[ANF001]</sup> Anforderungen abstimmen / vereinbaren: Entwickeln Sie jedes Ergebnis iterativ.....	18
3.8	<sup>[ANF010]</sup> Anforderungen aktuell halten: Führen Sie Change Requests. Führen Sie sie unabhängig von der Spezifikation.....	20
<b>4</b>	<b>Sicherheitsüberlegungen .....</b>	<b>21</b>
<b>5</b>	<b>Haftungsausschluss/Hinweise auf Rechte Dritter .....</b>	<b>21</b>
<b>6</b>	<b>Urheberrechte.....</b>	<b>21</b>
	<b>Anhang A – Referenzen &amp; Bibliographie .....</b>	<b>22</b>
	<b>Anhang B – Mitarbeit &amp; Überprüfung .....</b>	<b>24</b>
	<b>Anhang C – Abkürzungen.....</b>	<b>24</b>
	<b>Anhang D – Glossar .....</b>	<b>24</b>

## 1 Status des Dokuments

Das vorliegende Dokument wurde vom Expertenausschuss **aufgehoben**. Es hat für das definierte Einsatzgebiet im festgelegten Gültigkeitsbereich normative Kraft.

## 2 Einleitung

### 2.1 Überblick<sup>1</sup>

#### 2.1.1 Was versteht man unter Requirements Engineering & Management (RE&M)?

Unter Anforderungsingenieuren gibt es verschiedene Sichtweisen darüber, was Requirements Engineering, Requirements Management, Requirements Development und andere Begriffe bedeuten und in welchem Verhältnis sie zueinander stehen. Beispiele:

- Das International Requirements Engineering Board sieht Requirements Management als Teil des Requirements Engineering<sup>[22]</sup>
- Colin Hood (Mitglied des International Requirements Engineering Board) sieht Requirements Management jedoch als Disziplin *neben* dem Requirements Engineering<sup>[25]</sup>
- CMMI kennt den Begriff "Requirements Engineering" gar nicht, dafür jedoch die Begriffe „Requirements Management“ und „Requirements Development“<sup>[23]</sup>
- Das V-Modell XT benutzt den Begriff „Requirements Engineering“, definiert ihn jedoch weder explizit noch vollständig. Der Begriff wird beiläufig eingeführt mit der Definition der Rolle des Anforderungsanalytikers als „Anforderungserstellung und Anforderungsmanagement“. Die Bedeutung von „Anforderungsmanagement“ ergibt sich aus Bezug zu Werkzeugen für das Anforderungsmanagement. Die Bedeutung von „Anforderungserstellung“ bleibt offen<sup>[24]</sup>
- IEEE kennt alle diese Begriffe gar nicht<sup>[26]</sup>

In der Literatur gibt es weitere, sehr unterschiedliche Bedeutungen dieser Begriffe.

Dieses Hilfsmittel unterscheidet zwischen Requirements Engineering und Requirements Management. Es versteht unter Requirements Engineering sämtliche Tätigkeiten, welche erforderlich sind, um Anforderungen

- zu ermitteln / erheben
- zu verstehen / dokumentieren
- abzustimmen / zu vereinbaren

Dieses Hilfsmittel versteht unter Requirements Management sämtliche Tätigkeiten, welche erforderlich sind, um dokumentierte Anforderungen zu verwalten. Dies umfasst z.B.

---

<sup>1</sup> Dieser Abschnitt zitiert über weite Strecken aus [14], Abschnitte 1.1, 1.2 und 3, und berücksichtigt neueste Erkenntnisse aus [15]

- Anforderungen aktuell zu halten (Change Request Management),
- die Verfolgung von Anforderungsklärunge anhand von Statusattributen oder offenen Punkten,
- die Pflege von Beziehungen zwischen Anforderungen zum Zwecke der Rückverfolgbarkeit.

Mit diesem Verständnis orientiert sich dieses Hilfsmittel an den Definitionen von CMMI (Requirements Engineering gleichgesetzt mit Requirements Development) und Colin Hood.

RE&M findet grundsätzlich in jedem Projekt statt. Unterschiede bestehen darin, wie gezielt, bewusst, geplant und professionell RE&M zum Einsatz kommt.

Professionell ist in diesem Zusammenhang ein Vorgehen, welches in Kenntnis der gemäß dem Stand der Technik verfügbaren Prozesse, Methoden und Werkzeuge die der jeweiligen Projektsituation angemessenen Prozesse, Methoden und Werkzeuge für das RE&M einsetzt. Das professionelle Vorgehen unterstützt Ordnung in der Verwaltung der Anforderungen, ohne durch Bürokratie zu behindern. Es gibt dort Raum für Kreativität, wo dies erforderlich ist. Und es schafft dort das Fundament, wo die Projektarbeit stabilen Boden benötigt.

### **2.1.2 Warum sollten Projekte RE&M professionell durchführen?**

Die Projektmanagement Studie 2004 von GPM Deutsche Gesellschaft für Projektmanagement e.V. und PA Consulting Group<sup>[16]</sup> nennt „Unklare Anforderungen und Ziele“ mit knapp 70% als wichtigste Ursache für das Scheitern von Projekten. Im Jahr 2007 sind „Unklare Anforderungen und Ziele“ zwar nur noch zweitwichtigste Ursache für das Scheitern von Projekten<sup>[17]</sup>. Jedoch liegt der Prozentsatz immer noch bei knapp 60%. Dies legt den Schluss nahe, dass, wer Anforderungen und Ziele klar hat und klar hält, erheblich zur Minimierung von Projektrisiken beiträgt. Auch die Studien der Standish Group, die mehrere Tausend Projekte untersucht haben<sup>[18] [19] [20] [21]</sup>, kommen zu dem Schluss, dass einige Themen des systematischen RE&M wichtige Projekterfolgskriterien sind. So gehören zu den Top 10 die Einbindung der zukünftigen Produktnutzer, klare Geschäftsziele, minimierter Projektumfang und stabile Anforderungen.

Professionelles Requirements Engineering unterstützt das Projektmanagement, indem es die Qualität (z.B. Vollständigkeit, Klarheit, Verständlichkeit, Konsistenz) der erhobenen Rahmenbedingungen, Anforderungen und Abnahmekriterien für das innovative Produkt sicherstellt und erhöht.

Das professionelle Requirements Management unterstützt das Projektmanagement, indem es u.a. die Qualität von Abhängigkeitsanalysen sowie die Transparenz von Anforderungsänderungen verbessert. Zudem ermöglicht es, die Projektplanung auf Vollständigkeit gegen die Anforderungen zu prüfen

Vor der Inbetriebsetzung eines Systems bieten die dokumentierten Abnahmekriterien eine gute Basis für eine systematische und umfassende Qualitätsprüfung des Produkts. Wenn man den Erfolg des Produktes schon nicht mit Gewissheit vorhersagen kann, so erhält man am Ende doch wenigstens die Sicherheit, ein System entwickelt zu haben, welches die erhobenen Anforderungen erfüllt.

Professionelles RE&M unterstützt daher alle Beteiligten erheblich, sich z.B. auf jeweils einzelne Aspekte des neu zu entwickelnden Systems zu konzentrieren, sie zu analysieren, zu verstehen, Lösungen zu eruieren, durch Prototypen zu validieren etc., gleichzeitig den Zusammenhang und die Abhängigkeiten zu den anderen Aspekten des Systems nicht aus den Augen zu verlieren und somit zu jedem Zeitpunkt den Überblick über geplante, validierte, mögliche Eigenschaften und Qualität des neuen Systems zu behalten.

Professionelles RE&M anzuwenden wirkt ähnlich wie die sorgfältigen (und manchmal auch aufwändigen) Vorbereitungen, ein Segelboot vor einer Reise seefest zu machen: Man wendet Massnahmen präventiv an, weil viele Menschen gute Erfahrungen damit gemacht haben. Damit ist man zwar nicht gefeit davor, z.B. bei stürmischer See in risikoreiche Situationen zu geraten. Die Wahrscheinlichkeit dafür verringert sich jedoch deutlich, wenn man die einschlägigen Vorbereitungen sorgfältig trifft. Und wenn es wirklich ernst wird, ist man eher vorbereitet, mit der kritischen Situation geregelt und geordnet umzugehen. Daher ist RE&M aus Sicht des Projektleiters nicht nur eine produktive Tätigkeit, indem es die erste Grundlage für das zu erstellende System erstellt, sondern erfüllt im weiteren Projektverlauf dann wesentliche Funktionen im Risikomanagement.

### 2.1.3 Was ist eine Anforderung?

Beispiele für Anforderungen könnten sein:

- Das System soll dem Mitarbeiter im Kundenservice unmittelbar erkennbar anzeigen, dass zu einer Leistung ein Haftungsfall vorliegt.
- Das System soll dem Produktmanager die Möglichkeit bieten, einem Hotel-Baustein die Zimmerkategorien „Einfach“, „Gehoben“ und „Luxus“ zuzuordnen.

Weit verbreitet unter Anforderungsingenieuren ist folgende Definition von „Anforderung“<sup>[26]</sup>:

1. Eine Bedingung oder Fähigkeit, welche ein Stakeholder benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen.
2. Eine Bedingung oder Fähigkeit, welche ein System oder eine Systemkomponente besitzen oder einhalten muss, um einen Vertrag, einen Standard, eine Spezifikation oder andere formale Vorgaben zu erfüllen.
3. Eine dokumentierte Repräsentation einer Bedingung oder Fähigkeit gemäss (1) oder (2).

Formal umfasst diese Definition neben den Anforderungen an das zu erstellende Produkt genauso Termin- und Budgetvorgaben sowie Anforderungen an die Durchführung von Projekten wie beispielsweise Qualitätssicherungsmassnahmen und vom Projekt einzuhaltende Vorgaben und Standards. Ein Beispiel für eine Terminvorgabe als Anforderung: Dass ein Produkt vor der einschlägigen Entwicklungen des Wettbewerbs auf den Markt kommt ist eine Bedingung, welche der CEO des produzierenden Unternehmens benötigt, um das Ziel zu erreichen, sein Unternehmen gegenüber dem Wettbewerb hervorzuheben. Somit handelt es sich um eine Anforderung gemäss (1) der oben genannten Definition.

Formal umfasst diese Definition auch sämtliche programmiernahen Aspekte und diejenigen Themen, welche man üblicherweise unter „Software-Design“ und „IT-Architektur“<sup>[30]</sup>. Hieraus leiten die Autoren die Best Practice [ANF007] ab, sämtliche Ergebnisse des Entwicklungs-

prozesses als Dokumentation von Anforderungen zu verstehen, auch diejenigen Ergebnisse aus eher programmiernahen Bereichen.

Eine Anforderung gilt auch dann schon als eine Anforderung, wenn sie noch undokumentiert oder sogar unbekannt ist. Die Dokumentation von Anforderungen trägt jedoch sehr zu deren Qualität bei und ist die Basis für eine nachvollziehbare und verbindlich vereinbarte Arbeitsteilung und sowie die revisionstechnische Prüfbarkeit, u.a. durch die Aufdeckung von Missverständnissen zwischen Beteiligten.

Weit verbreitet sind zwei Formen der Dokumentation von Anforderungen:

- Textuell als „One Sentence-Requirement“
- Grafisch z.B. in einem UML- oder SysML-Diagramm.

Jede semantisch klare grafische Notation wie z.B. die UML lässt sich systematisch in eine Menge von semantisch gleichwertigen „One Sentence-Requirements“ umwandeln. Diese Menge von „One Sentence-Requirements“ und deren gegenseitige Abhängigkeiten sind dann sicher ungleich schwieriger zu überblicken als das entsprechende Diagramm. Das ändert jedoch nichts an der prinzipiellen Abbildbarkeit.

Die dokumentierte Repräsentation aller Anforderungen wird üblicherweise wesentlicher Bestandteil des Vertrages zwischen dem Auftraggeber des Projektes und dem Auftragnehmer. Der Auftragnehmer muss nicht der Projektleiter sein. Der Vertrag wirkt sich aber auf jeden Fall auf den Projektleiter aus. Genauso muss der Auftraggeber nicht unbedingt ein Endanwender des Produkts sein, sondern kann z.B. auch ein Produktmanager sein.

#### 2.1.4 Funktionale Anforderungen und Qualitätsanforderungen

Das Beispiel

*Das System soll dem Mitarbeiter im Front Office unmittelbar erkennbar anzeigen, dass zu einer Sendung ein Haftungsfall vorliegt.*

enthält neben dem sogenannten „funktionalen“ Aspekt

*„[...] anzeigen, dass zu einer Sendung ein Haftungsfall vorliegt“*

auch einen „qualitativen“ oder „nicht-funktionalen“ Aspekt:

*„[...] unmittelbar [...]“.*

Während man das Vorhandensein von funktionalen Eigenschaften recht einfach und objektiv prüfen kann (z.B. ist die Information an der Benutzerschnittstelle sichtbar oder nicht), ist es weitaus weniger leicht, den Grad der Erfüllung der Qualitätsanforderungen festzustellen. Die Beurteilung, ob die Information im Beispiel „unmittelbar“ erkennbar ist oder nicht, ist subjektiv: Für den einen Benutzer mag dies bei einer bestimmten Implementierung so sein, für einen anderen Benutzer nicht. Dies gilt für eine ganze Reihe weiterer qualitativer Aspekte wie „Einfache Bedienbarkeit“, „Flexibel in der Anpassbarkeit“, „Ausfallsicherheit“, „Robustheit“ oder Ähnlichem.

Daher kommt der Quantifizierung von Qualitätsanforderungen eine erhebliche Bedeutung zu: Wie kann man objektiv feststellen, dass eine Qualitätsanforderung erfüllt ist oder nicht? Hier ein Beispiel:

Anforderung	Die Software soll für die Mitarbeiter der Produktion einfach und intuitiv zu bedienen sein und ein ‚learning by doing‘ ermöglichen.
Begründung	Die Mitarbeiter der Produktion arbeiten selten mit der Software. Ein hoher Schulungsaufwand ist daher nicht zu rechtfertigen
Abnahmekriterium	An 7 Standorten arbeiten je 5 Mitarbeiter (in Summe 35 Personen) je eine Viertelstunde lang mit dem System. Die Mitarbeiter sollen das System noch nicht kennen und noch keine Information über das System erhalten haben. 80% der Testpersonen sollen das System mit mindestens „3“ auf einer Skala von „1 – sehr gut“ bis „7 – absolut unbrauchbar“ bewerten.

[27] beschreibt einen Prozess, wie man Qualitätsanforderungen systematisch quantifizieren kann, aber auch die Goal-Question-Metric Methode GQM<sup>[28]</sup> unterstützt dies.

### 2.1.5 Anforderung und Lösung – eine Frage der Perspektive<sup>2</sup>

Anforderungen sollen lösungsneutral sein. Aber was bedeutet „lösungsneutral“, wenn beispielsweise das BABOK Geschäftsprozesse als Teil der Lösung ansieht, welche ein Unternehmen für Anforderungen des Marktes und seiner Kunden anbietet, gleichzeitig aber die Literatur zum Software Engineering Geschäftsprozesse als Anforderungen für die Softwaresysteme zur Unterstützung dieser Geschäftsprozesse versteht? Sind die Geschäftsprozesse nun Lösung oder sind sie Anforderung? Antwort: Sie sind beides. Anforderung und Lösung sind zueinander dual wie Welle und Teilchen in der Quantenphysik. Je nach Perspektive und untersuchtem Problem schaut man durch die eine oder durch die andere Brille.

Anders formuliert: Anforderungen bezeichnen Ziele und Randbedingungen, die man bei der Entwicklung einer Lösung erreichen respektive einhalten muss. Somit ist der Geschäftsprozess je nach Sichtweise eine Anforderung oder ein Teil der Lösung. z.B.:

a) Ziel: Das Geschäft soll gesamthaft effizienter werden:

- „Geschäftsprozesse optimieren“ ist Teil der Lösung
- Softwaresystem muss vermutlich die obigen Geschäftsprozesse unterstützen. Somit sind die Geschäftsprozesse auch Anforderungen, jedoch nur für das Subprojekt „Software“.

Es gibt da eine interessante Rückkopplung: Die Möglichkeiten der Software erlauben ganz andere Prozesse. Dies erklärt, dass die Lösung und die Anforderungen sich iterativ gegenseitig beeinflussen.

b) Das Softwaresystem muss das Geschäft unterstützen:

- Geschäftsprozesse bestehen bereits und sollen nicht verändert werden. Also sind sie eine Anforderung, aber nicht Teil der Lösung.

Im Folgenden wird dieser Zusammenhang theoretisch erläutert anhand des V-Modell und des Zachman Framework.

<sup>2</sup> Dieser Abschnitt führt die in [38] und [39] veröffentlichten Ideen weiter



Das V-Modell (Abbildung 1) wurde in den 1990er Jahren in den USA<sup>[35]</sup> und in Deutschland<sup>[36]</sup> parallel zueinander, aber unabhängig voneinander entwickelt<sup>[37]</sup>. Seinen Namen bezieht es aus der V-förmigen Darstellung.

Eine wesentliche Aussage des V-Modells ist, dass man es im Systementwicklungsprozess auf unterschiedlichen Ebenen mit Anforderungen und Lösungen zu tun hat. Die Anforderungen einer lösungsferneren Ebene werden im Rahmen des Lösungsentwurfs transformiert in Anforderungen an Komponenten auf einer lösungsnäheren Ebene. Diese Transformation kann auf verschiedene Weise erfolgen:

- Lösungingenieure (Menschen) können dies leisten, sofern sie Lösungskompetenz auf der jeweiligen Ebene der Anforderungen besitzen (z.B. die Unternehmensleitung auf der Ebene Geschäftskontext, Business Analysten auf den Ebenen CIM und PIM, Softwareingenieure auf den Ebenen PSM und Implementierung)
- Softwareprodukte können dies leisten, z.B. bei Einsatz von domänenspezifischen Sprachen (DSL)<sup>[5] [6] [31]</sup> oder Ansätzen wie Model Driven Development (MDD)<sup>[11]</sup>.

Eine weitere wesentliche Aussage des Modells ist (und dies ist die eigentliche Wurzel des Modells), dass es zu jeder Anforderung eine Abnahmetätigkeit gibt, welche auf der gleichen Ebene stattfindet, zu welcher die Anforderung gehört. Unter der Voraussetzung, dass man zuvor die richtigen Anforderungen vollständig ermittelt hat, fördert dieses Vorgehen erheblich die Sicherheit, das richtige System gebaut zu haben.

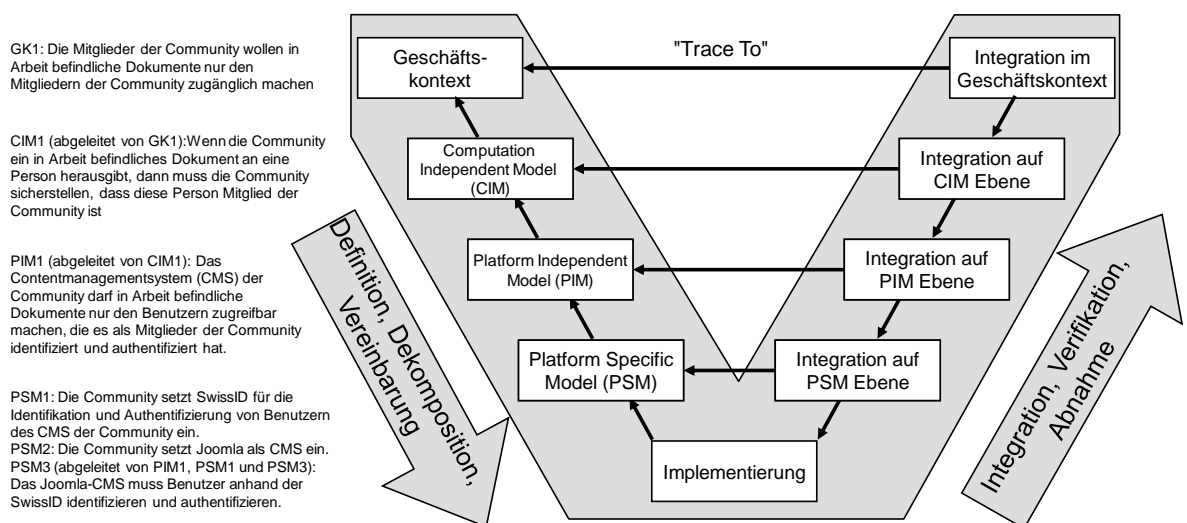


Abbildung 1: V-Modell mit Beispielen für Anforderungen auf den verschiedenen Ebenen am Beispiel einer virtuellen Community, wobei die Benennung der Ebenen an die Terminologie der OMG<sup>[11]</sup> angeglichen ist. Der Unterschied zwischen GK1 und CIM1 besteht darin, dass GK1 eine Aussage über eine Rahmenbedingung für die Community macht, während CIM1 eine Aussage über die (Geschäfts-)Abläufe innerhalb der Community trifft.

Das Zachman Framework<sup>[8][9]</sup> (Abbildung 2) ist eine Matrix. Es kategorisiert die Ergebnisse des Systementwicklungsprozesses anhand der Fragen, die diese Ergebnisse bezogen auf das zu entwickelnde System beantworten (Spalten der Matrix), und anhand der Ebenen, auf



welcher ein Ergebnis eine Frage beantwortet (Zeilen der Matrix). Die folgende Tabelle erläutert die Spalten der Matrix:

Frage	Ergebniskategorie
<i>Was</i> ist Gegenstand der Verarbeitung	Daten
<i>Wie</i> werden die Gegenstände verarbeitet	Prozess
<i>Wo</i> werden die Gegenstände verarbeitet	Netzwerk
<i>Wer</i> verarbeitet die Gegenstände	Akteur
<i>Wann</i> werden die Gegenstände verarbeitet	Zeit
<i>Warum</i> werden die Gegenstände verarbeitet	Anforderungen

Das Zachman Framework trifft die folgenden wesentlichen Aussagen:

- Die erwähnten Fragen muss man auf unterschiedlichen Ebenen beantworten, um ein Problem vollständig verstanden und eine Lösung für das Problem entwickelt zu haben. Die Antwort auf das „Warum“ beschreibt die Anforderungen auf jeder Ebene. Die Antworten auf alle anderen Fragen beschreiben die Lösung auf der jeweiligen Ebene.
- Die Ergebnisse aus der Beantwortung der Fragen auf den verschiedenen Ebenen können innerhalb der gleichen Ebene beliebig voneinander abhängen. Ebenenübergreifend hängen Ergebnisse einer tieferen Ebene immer von den Ergebnissen der höheren Ebene ab, jedoch nicht umgekehrt.
- Es ist egal, an welcher Frage und auf welcher Ebene man zu arbeiten beginnt und in welcher Reihenfolge man die Fragen beantwortet. Wenn man das richtige System entwickeln möchte, müssen am Ende alle Fragen auf allen Ebenen beantwortet sein.

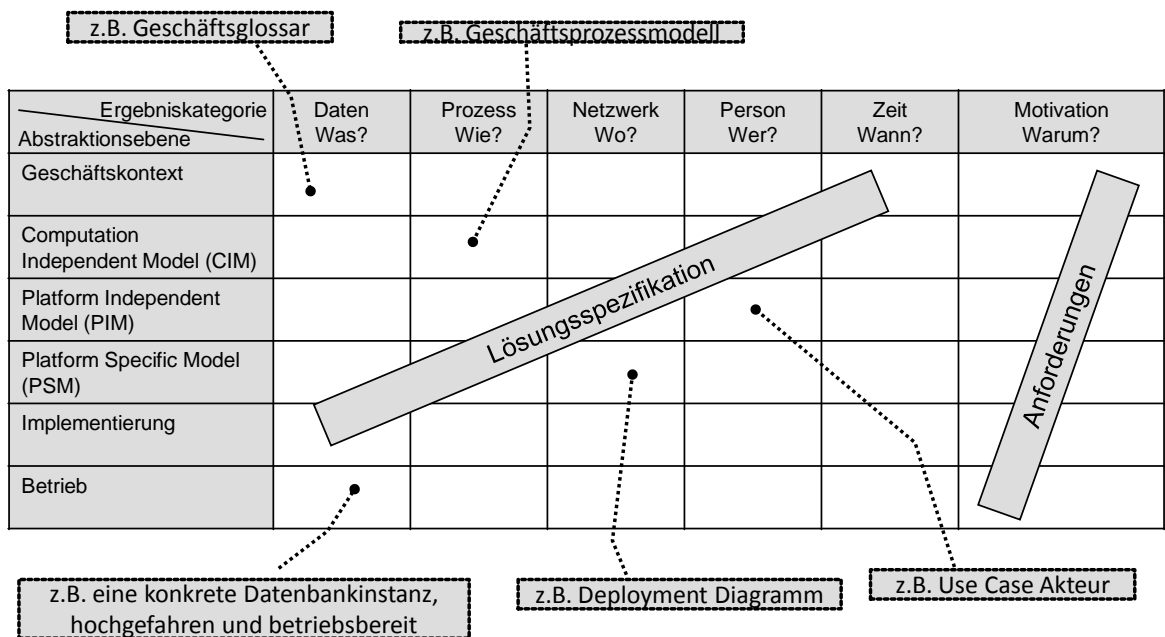


Abbildung 2: Zachman Framework mit Modellbeispielen für verschiedene Ebenen und Fragen, wobei die Benennung der Ebenen an die Terminologie der OMG<sup>[1][12]</sup> angeglichen ist.

Kombiniert man nun die Darstellung des V-Modell mit den Konzepten des Zachman Framework (Abbildung 3; beide Modelle haben identische Ebenen<sup>3</sup>), so werden folgende Aspekte deutlich:

- Auf jeder Ebene der Systementwicklung gibt es Anforderungen und Lösungen. Plattformneutral formulierte Use Cases sind aus Sicht der Geschäftsprozessanalyse z.B. Teil der Lösung, aus Sicht der Softwareentwicklung dokumentieren sie jedoch Anforderungen.
- Im Verlauf der Systementwicklung wechseln das RE&M und die Lösungsfindung und –spezifikation einander ab (Abbildung 4, Ziffern 2 und 3). Würde man die für iterativ-inkrementelles oder agiles Vorgehen typischen Rückkopplungsschleifen berücksichtigen, dann würde man erkennen, dass RE&M und die Lösungsfindung und –spezifikation auf mannigfache Art und Weise miteinander verwoben sind. Um die Konzentration auf die wesentlichen Kernaussagen zu fördern, sind diese Rückkopplungsschleifen in Abbildung 4 nicht dargestellt.
- Eine Lösungsspezifikation auf einer Ebene zu erstellen bedeutet gleichzeitig, die Anforderungen für die darunterliegende Ebene zu formulieren. **Damit ist begründet, weswegen sowohl Geschäftsprozessanalysten als auch IT-Architekten im RE&M ausgebildet sein müssen!**

<sup>3</sup> Die Ebene „Betrieb“ des Zachman Framework bezeichnet den tatsächlichen Betrieb des Systems im Gegensatz zur Spezifikation und Entwicklung des Systems auf den darüberliegenden Ebenen. Die Ebene „Betrieb“ bezeichnet *nicht* die Anforderungen, welche das System aus Sicht der IT Betriebsabteilung erfüllen muss. Daher ist die Ebene „Betrieb“ des Zachman Framework für den Systementwicklungsprozess nicht relevant.

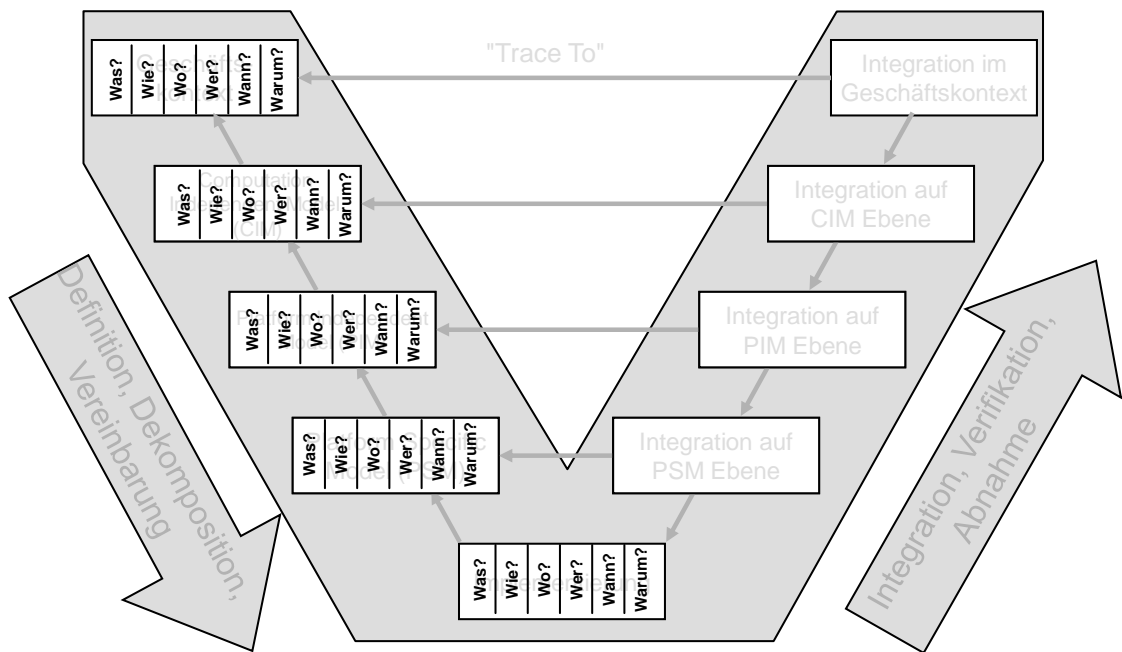


Abbildung 3: V-Modell und Zachman Framework miteinander kombiniert



### **3 Best Practice für Requirements Engineering & Management sowie plattformunabhängige Modellierung**

[ANF049] Jede Anforderung hat einen eindeutigen Identifikator <sup>[ANFxxx]</sup>. Diesen behält die Anforderung auch dann, wenn sie zwischen Abschnitten einer Spezifikation verschoben wird. Daher entspricht die Reihenfolge von Absätzen einer Spezifikation nicht zwingend der Nummerierung der Identifikatoren.

In diesem Dokument ist jede der aufgeführten Best Practices mit einem solchen Identifikator versehen. Denn aus Sicht der Autoren dieser Best Practices ist jede Best Practice gleichzeitig eine Anforderung, welches jedes Projekt erfüllen sollte.

#### **3.1 <sup>[ANF006]</sup> Planen Sie Ihr Requirements Engineering & Management**

[ANF014] Schreiben Sie auf, wie Sie im Requirements Engineering & Management und in der Modellierung vorgehen wollen.

[ANF015] Stimmen Sie das aufgeschriebene Vorgehen mit allen Stakeholdern ab und vereinbaren Sie es mit allen Stakeholdern.

[ANF050] Schulen Sie Ihre Mitarbeiter im Requirements Engineering & Management. Lassen Sie Ihre Mitarbeiter zertifizieren, z.B. zum IREB Certified Professional for Requirements Engineering <sup>[32]</sup>

[ANF051] Rechnen Sie mit einem Aufwand zwischen 15% und 20% für das Requirements Engineering & Management <sup>[34]</sup>

#### **3.2 <sup>[ANF007]</sup> Anforderungen verwalten: Betrachten Sie jedes (jedes!) Ergebnis als eine Anforderungsspezifikation**

Von sehr wenigen Ausnahmen wie z.B. einem Benutzerhandbuch abgesehen, dokumentieren alle Ergebnisse Anforderungen oder anforderungsähnliche Sachverhalte.

[ANF002] Planen Sie nur Arbeitspakete die relevant für das Ziel und die Ergebnisse sind. Gestalten Sie diese Planung als iterativen Prozess (Am Anfang ist das Ziel). Referenzieren Sie eindeutig von jedem Arbeitspaket auf die im Arbeitspaket zu dokumentierenden Ergebnisse.

[ANF004] Lassen Sie ihre Ergebnisse frühzeitig auf ihre Machbarkeit verifizieren. Dokumentieren Sie Lösungsideen bei den jeweiligen Ergebnissen. Requirements Engineering & Management muss das Ziel haben, dass Requirements effizient implementiert werden können.

[ANF005] Legen Sie Verantwortlichkeiten eindeutig fest  
Benennen Sie für jedes Ergebnis genau eine Person (und nicht zwei und nicht keine), die für dieses Ergebnis zuständig ist. Diese Person muss das Ergebnis kennen und den Lebenslauf des Ergebnisses steuern. Diese Zuständigkeit gilt unabhängig davon, wer ggf. Veränderungen an dem Ergebnis vornimmt.

### 3.3 [ANF009] Anforderungen verwalten: Unterscheiden Sie Ergebnisse auf verschiedenen Abstraktionsebenen

[ANF025] Definieren sie Ihre Abstraktionsebenen z.B. in Anlehnung an das Zachman Framework<sup>[8] [9] [10] [12] [13]</sup> und die Standards der OMG (Geschäftskontext, Computation Independent Model – CIM, Platform Independent Model -PIM, Platform Specific Model - PSM, Implementierung)<sup>[11] [12]</sup>.

[ANF026] Ordnen Sie jedes Ergebnis eindeutig genau einer Abstraktionsebene zu.

[ANF027] Legen Sie für jede Abstraktionsebene die Dokumentationsmittel fest  
Auf jeder Abstraktionsebene kann es erforderlich sein, andere Dokumentationsmittel als auf den anderen Abstraktionsebenen zu verwenden (z.B. mehr Prosa bei der Strategie, sehr umfassende Modellierung ohne Prosa in der plattformspezifischen Modellierung)

[ANF028] Definieren Sie Ermittlungs-, Erhebungs- und Transformationsmethodik  
Sie müssen Stakeholder, Fachbereich und andere Wissensträger so systematisch befragen, dass eine gemeinsam getragene Modell-Instanz als Essenz entsteht. Diese Modell-Instanz ist das Ergebnis einer Transformation von Anwendungsbe-  
reich / Problembereich / übergeordnete Abstraktionsebene in Richtung des Lösungsbereichs/ in Richtung der aktuellen Abstraktionsebene. Die Modell-Instanz ist dann Essenz, wenn die weitere Transformation in die nächstniedrigere Abstraktionsebene möglich ist. Eine solche Essenz benötigen Sie auf jeder Abstraktionsebene.

### 3.4 [ANF012] Anforderungen ermitteln/erheben: Legen Sie die Ermittlungstechniken ausdrücklich fest und wenden Sie diese dann auch an

Je nach Projektsituation sind unterschiedliche Vorgehensweisen bei der Erhebung und Ermittlung von Anforderungen nützlich.

### 3.5 [ANF008] Anforderungen verstehen und dokumentieren: Modellieren und spezifizieren Sie formal

Hierzu gibt es eine ganze Palette an Hilfsmitteln: Anforderungssyntaxschablonen<sup>[3]</sup>, Domainspezifische Sprachen<sup>[5] [6]</sup>, Standardsoftware für Modellierung.

[ANF016] Formulieren Sie mit Anforderungssyntaxschablonen  
Wenn Sie Prosa-Texte schreiben, dann formulieren Sie jeden Satz in Anlehnung an die Anforderungssyntaxschablone<sup>[3]</sup> oder SCRUM User Stories<sup>[4]</sup>

[ANF017] Erstellen Sie ein Domain Modell  
Domain Modelle umfassen statische (z.B. Konzepte und Beziehungen zwischen diesen Konzepten) und dynamische Sicht (z.B. Prozessmodelle und Use Case-Modelle) auf das System. Domain Modelle enthalten Prosa als Beschreibung für jedes Modellelement. Domain Modelle sind sehr hilfreich, die verwendete Terminologie zu strukturieren resp. zu modularisieren und um Konzepte zu hierarchisieren. Ein Domain Modell drückt die Essenz aus der gemeinsamen Sicht aller involvierten Fachexperten.

- [ANF018] Pflegen Sie ein domänenspezifisches Vokabular (= Ontologie, Semantik, Glossar) Stimmen sie Terminologie und Glossar mit den Fachexperten ab. Verlangen Sie, dass diese Begriffe eine lange Lebensdauer haben. Dies umfasst ein Glossar, in dem nicht nur Substantive, sondern auch Verben definiert sind. Stimmen sie das Vokabular mit allen Beteiligten ab.
- [ANF019] Definieren Sie ihre Modellierungssprache so früh wie möglich. Modellierungssprachen können sein z.B. UML, DSLs <sup>[5] [6] [31]</sup>, BPMN/BPML <sup>[7]</sup>, PLanguage <sup>[1] [2]</sup> u.a. Wählen Sie Ihre Modellierungssprache so, dass ihre fachlichen Auftraggeber ihre Modellinstanzen verstehen.
- [ANF020] Jede Modellinstanz ist ein Ergebnis = Anforderungsspezifikation im oben erwähnten Sinn.
- [ANF021] Ermitteln Sie die nichtfunktionalen Anforderungen.
- [ANF022] Die Gesamtheit aller Modellinstanzen und nichtfunktionalen Anforderungen ist die Spezifikation.
- [ANF054] Dokumentieren / modellieren Sie klar, was zum System gehört und was nicht. Unterscheiden Sie zwischen System und Systemumgebung. Dokumentieren Sie Abgrenzungen.

### 3.6 <sup>[ANF011]</sup> Anforderungen verwalten: Dokumentieren sie alle Ergebnisse in einem zentralen Repository

Mit "Repository" ist eine Lösung zur Verwaltung von Ergebnissen gemeint, welche Auswertungen ermöglicht. Das bietet ein Repository:

- Aufnahme, Modifikation, Wiedergabe verschiedenster Ergebnistypen
- Jedes Ergebnis hat einen eindeutigen Identifikator, den das Repository bei der Erzeugung des Ergebnisses automatisch vergibt und den man im Lebenszyklus des Ergebnisses nicht ändern kann.
- Pro Veränderung an einem Ergebnis erhält das Ergebnis eine neue Versionsnummer. Der Identifikator bleibt jedoch unverändert
- Die Möglichkeit, Metadaten oder Attribute für ein Ergebnis zu pflegen. Diese Metadaten oder Attribute benötigen Sie für die Organisation, Planung und Verwaltung Ihrer Ergebnisse du Ihres Projekts
- Zu jedem Ergebnis eine beliebige Zahl von Anhängen hinzuzufügen.

[ANF056] Entscheiden Sie bewusst, was Sie dokumentieren (lassen) und was nicht.

Die Ergebnisse sollen vollständig sein. Dies bedeutet jedoch nicht, dass eine vollständige Dokumentation in jedem Fall nötig ist, solange die Ergebnisse im Kreis der relevanten Personen gut genug präsent sind. In manchen Fällen kann es sogar aus „politischen“ Gründen kontraproduktiv sein, bestimmte Ergebnisse zu dokumentieren.

Auf der anderen Seite muss – oft auch dann, wenn Auftraggeber und Auftragnehmer der Produktentwicklung zur selben Organisation gehören - ein Vertrag zur Entwicklung eines Systems den geforderten Leistungsumfang vollständig beschreiben.



Daher ist eine geeignete Mischung zwischen Dokumentation und Nicht-Dokumentation von Ergebnissen erforderlich.

[ANF029] Erfassen und bearbeiten Sie Ergebnisse nur im Repository.

[ANF030] Dokumentieren sie sowohl alle Texte (Anforderungen, Konzeptpapiere, Protokolle, offene Punkte, Risiken) als auch alle Modelle und alle anderen Arten von Grafiken als Ergebnis in dem zentralen Repository. Diese Dokumentation umfasst allenfalls die Dokumentation von Abhängigkeiten des Ergebnisse von und zu anderen Ergebnissen (Traceability)

[ANF053] Verwenden Sie Traceability nur dann, wenn Sie sie tatsächlich konsistent und vollständig führen. Aber Vorsicht: Verfolgbarkeit konsistent und vollständig zu setzen und zu verwalten kann einen erheblichen Aufwand erfordern. Verfolgbarkeit bietet ihren Nutzen aber nur dann, wenn sie tatsächlich konsistent und vollständig gepflegt ist (vgl. z.B. [33], S. 360).

[ANF031] Verwenden Sie keinen anderen Ort als dieses Repository zur Dokumentation von Ergebnissen.

[ANF052] Führen Sie für jedes Ergebnis einen Bearbeitungsstatus im Repository

[ANF032] Jedes Ergebnis muss notwendig sein.

Verwalten Sie Ergebnisse nur so lange, wie es einen Nutzen und Zweck für die Ergebnisse gibt. Es macht keinen Sinn, veraltete Ergebnisse ewig in der aktuellen Sicht auf das Modell zu halten.

Notwendigkeit eines Ergebnisses entsteht aus folgenden Gründen (nach Scott Ambler):

1. Eindeutiger und wertschöpfender Grund warum das Ergebnis ein permanenter Teil der Projektdokumentation ist
2. Es gibt ein Publikum, das dieses Ergebnis als wertvoll erachtet
3. Es gibt Stakeholder, die bereit sind in das Ergebnis zu investieren.

Die folgenden Fragen helfen Ihnen, die Notwendigkeit eines Ergebnisses zu überprüfen:

- was (ist das Ergebnis)
- wann (unter welchen Bedingungen ist das Ergebnis gültig)
- wer (ist von dem Ergebnis betroffen)
- wie (sehen Details des Ergebnisses und Lösungen aus)
- warum (existiert das Ergebnis)

[ANF033] Markieren Sie Ergebnisse, die mehr nicht relevant sind, aber löschen Sie diese Ergebnisse nicht

In jedem Projekt gibt es Ergebnisse, von denen man anfänglich glaubt, sie zu brauchen, jedoch hinterher feststellt, dass man sie nicht mehr braucht. Je nach Status sollten Sie Ergebnisse und ihre Beziehungen aus der aktuellen Sicht auf das Modell und aus Auswertungen des Modells nehmen.

- [ANF034] Trennen sie Inhalt eines Ergebnisses von der Präsentation des Ergebnisses und automatisieren sie die Entstehung der Präsentation  
Inhalte sind nicht an eine Form gebunden. Dann sollten sie auch medienneutral und formal gespeichert werden können. Der gleiche Inhalt kann verschieden präsentiert werden, je nachdem welchen Zweck man mit dem Inhalt erreichen will. Insofern darf der Inhalt keine präsentationsspezifischen Eigenschaften enthalten.
- [ANF035] Erzwingen Sie nach Möglichkeit durch Werkzeugumgebung die Trennung von Präsentation und Inhalt.
- [ANF036] Verwenden sie Office-Werkzeuge zur Ausgabe von Reports  
WYSIWYG-Editoren: Verwenden Sie WYSIWYG-Editoren allenfalls für die Erzeugung von Reports aus dem Repository, jedoch nicht als Werkzeug zur Dokumentation von Ergebnissen der Anforderungserhebung.  
Tabellenkalkulationswerkzeuge: Verwenden Sie Tabellenkalkulationswerkzeuge allenfalls für die Dokumentation von Anhängen zu anderen Ergebnissen oder als Werkzeug zur Aufbereitung von Auswertungen aus dem Repository.  
Präsentationswerkzeuge: Verwenden Sie Präsentationswerkzeuge allenfalls für die Präsentation von Ergebnissen, jedoch nicht als Werkzeug zur Dokumentation von Ergebnissen.
- [ANF037] Achten Sie auf Versionierung und Dokumentation von Änderungen an Ergebnissen  
Da ein Ergebnis eine atomare Einheit der Arbeit im allgemeinen Spezifikationsprozess ist, sollte es nur von einer Person zu jedem gegebenen Zeitpunkt editierbar sein. Exklusive Sperrmechanismen (Locks) verhindern Synchronisierungs- und Konsistenzprobleme in einer großen Teamumgebung.
- [ANF038] Legen Sie die Granularität Ihrer Lieferungen fest  
Nicht jede neue Version eines Ergebnisses muss zu einer neuen Lieferung an den Auftraggeber führen. Legen Sie stattdessen fest, welche Liefereinheiten es gibt (z.B. eine komplette Use Case-Spezifikation einschliesslich aller Szenarien oder die vollständige Spezifikation einer Komponente einschliesslich aller Modellinstanzen, die zu dieser Komponente gehören.
- [ANF039] Es gibt Ausnahmen von der Pflicht, Ergebnisse im Repository zu verwalten
- [ANF040] Wenn Sie um Verwendung von Ergebnissen in Form von Dateien im Dateisystem nicht umhin kommen (z.B. wenn Sie Projektpläne mittels MS Project dokumentieren): Speichern Sie jede Datei in einem Werkzeug für Konfigurationsmanagement, um eine eindeutige Referenz auf dieses Ergebnis und seine Versionen zu erhalten und zu ermöglichen.
- [ANF041] Verwalten Sie im Repository eine Referenz auf die Ausnahme (z.B. Datei)
- [ANF042] Verwalten Sie Ausnahmen im Repository durch Eindeutigkeit bei Namen, Speicherort, Version und Status.
- [ANF043] Vermeiden Sie Datumsangaben und Versionsnummern im Dateinamen.  
Aber: In seltenen Ausnahmefälle (z.B. Releases von Komponenten) kann es

erforderlich sein, dass Sie Datumsangaben und/oder Versionsnummern im Dateinamen verwenden müssen.

[ANF044] Verwalten Sie den Zustand der Ausnahme in der Referenz auf die Ausnahme. Führen Sie für die Datei einen Bearbeitungsstatus im Repository wie für jedes andere Ergebnis auch.

[ANF045] Verwalten Sie offene Punkte, Risiken sowie deren Beziehungen zu Ergebnissen im Repository. Stellen Sie stets Bezüge her zwischen den offenen Punkten / Risiken und den Ergebnissen, aus welchen die offenen Punkte / Risiken entstehen bzw. auf welche sich die offenen Punkte / Risiken beziehen.

[ANF046] Ermitteln Sie Kennzahlen durch Auswertung des Repositorys. Werten Sie den Bearbeitungsstatus der Ergebnisse aus und leiten Sie daraus Kennzahlen für das Projektmanagement ab.

[ANF047] Rechnen Sie mit Werkzeugbrüchen. Aus technischen Gründen (z.B. unzureichende Integration zwischen Standardsoftware verschiedener Hersteller) kann das Repository aus mehreren technisch unterschiedlich implementierten Datenbanken bestehen. Beispiel: Eine Standardsoftware für Anforderungsmanagement für die Prosa und eine Standardsoftware für UML-Diagramme für Modelle. Je mehr Werkzeugbrüche Sie haben, umso schwieriger wird es, automatisiert Reports und Auswertungen aus dem Repository zu erhalten.

### 3.7 [ANF001] Anforderungen abstimmen / vereinbaren: Entwickeln Sie jedes Ergebnis iterativ

[ANF023] Iterativ entwickeln heisst: Erbitten Sie frühe und mehrfache Rückmeldung zu Ihren Ergebnissen.

[ANF013] Nutzen Sie Prüfungsverfahren als Mittel zur Kommunikation von Ergebnissen und als Verfahren zur Abstimmung und Vereinbarung von Ergebnissen. Gemeint ist eine Prüfung von Ergebnissen im Vorfeld einer Vereinbarung des detaillierten Projektumfangs. Gemeint ist nicht eine Prüfung, ob ein geliefertes System einer zuvor vereinbarten Spezifikation entspricht.

[ANF024] Lassen Sie die Ergebnisse von denjenigen Personen prüfen, die die Ergebnisse anschliessend weiter verwenden (z.B. Architekten, Tester, Entwickler, Auftraggeber, ggf. weitere Stakeholder)

[ANF055] Gegenstand der Prüfung / Abstimmung / Vereinbarung sollen nur die Anforderungen an das System sein. Die Dokumentation der Systemumgebung sollte nicht Gegenstand der Prüfung / Abstimmung / Vereinbarung sein.

[ANF057] Nutzen Sie Prüfungsverfahren nicht nur zur Kommunikation von Ergebnissen, sondern gleichzeitig auch zur Qualitätssicherung. Prüfen Sie Ergebnisse anhand der Qualitätskriterien für Anforderungen (abgeleitet von [29]):

Identifizierbar      Siehe [ANF049]

Eindeutig	Es gibt nur eine mögliche Interpretation des Ergebnisses
Konsistent	Keine Widersprüche (z.B. derselbe Begriff wird für verschiedene Inhalte verwendet oder verschiedene Begriffe für denselben Inhalt), Terminologie passend zum Glossar, Testfälle und andere Projektdokumente verwenden die gleiche Terminologie
Verständlich	Klar strukturiert, präzise Beschreibungen
Vollständig	In sich abgeschlossen, alle Erwartungen den Stakeholder abdeckend, offene Punkte sind ausdrücklich erwähnt
Korrekt	Zutreffende Darstellung des Wissens der Stakeholder, ordnungsgemäß abgeleitet von anderen Ergebnissen (z.B. auf höheren Detaillierungsebenen)
Atomar	Gilt für Prosa: Nur ein Aspekt pro Anforderung
Machbar	Innerhalb der technischen und betriebswirtschaftlichen Rahmenbedingungen (welche ihrerseits Anforderungen sind, welche das Projekt einhalten muss)
Priorisiert	Von allen betroffenen Stakeholdern. Eine Priorisierung ist meistens erst dann möglich, wenn Aufwände und Kosten für die Umsetzung von Ergebnissen geschätzt sind. Die Erfahrung lehrt, dass Auftraggeber häufig erst dann auf Umsetzung von Ergebnissen verzichten, wenn sie erfahren, dass der zu erwartende Nutzen in einem ungünstigen Verhältnis zu den Kosten steht.
Testbar	Es ist möglich, ein Abnahmekriterium zu formulieren.  Es macht Sinn, zumindest das Abnahmeverfahren frühzeitig zu spezifizieren, weil sich aus dem Abnahmeverfahren in der Regel auch Anforderungen an das zu erstellende System ergeben. Für Performance-Messungen muss ein IT-System z.B. an bestimmten Stellen im Programm Zeitstempel in eine Logdatei schreiben. Und ein Steuergerät in einem Kfz (z.B. ABS oder ACC) benötigt evtl. eine spezielle Schnittstelle, um Statusinformation während der Abnahmetests abfragen zu können.
Rückverfolgbar	Durch Dokumentation der Entwicklungsgeschichte des Ergebnisses, ihrer Quelle sowie der Beziehungen zu anderen Ergebnissen
Nützlich	Siehe [ANF032]

**3.8 [ANF010] Anforderungen aktuell halten: Führen Sie Change Requests. Führen Sie sie unabhängig von der Spezifikation**

[ANF048] Wenn sich der Projektumfang ändert, ändern Sie nicht sofort in der Spezifikation, sondern dokumentieren Sie den Änderungsbedarf, lassen Sie über den Änderungsbedarf entscheiden, und dann ändern Sie die Spezifikation.

## 4 Sicherheitsüberlegungen

Aus Sicht der Autoren sind während der Ausarbeitung dieses Hilfsmittels keine Fragestellungen aufgetreten, welche Sicherheitsüberlegungen erforderlich machen.

## 5 Haftungsausschluss/Hinweise auf Rechte Dritter

**eCH**-Standards, welche der Verein **eCH** dem Benutzer zur unentgeltlichen Nutzung zur Verfügung stellt, oder welche **eCH** referenziert, haben nur den Status von Empfehlungen. Der Verein **eCH** haftet in keinem Fall für Entscheidungen oder Massnahmen, welche der Benutzer auf Grund dieser Dokumente trifft und / oder ergreift. Der Benutzer ist verpflichtet, die Dokumente vor deren Nutzung selbst zu überprüfen und sich gegebenenfalls beraten zu lassen. **eCH**-Standards können und sollen die technische, organisatorische oder juristische Beratung im konkreten Einzelfall nicht ersetzen.

In **eCH**-Standards referenzierte Dokumente, Verfahren, Methoden, Produkte und Standards sind unter Umständen markenrechtlich, urheberrechtlich oder patentrechtlich geschützt. Es liegt in der ausschliesslichen Verantwortlichkeit des Benutzers, sich die allenfalls erforderlichen Rechte bei den jeweils berechtigten Personen und/oder Organisationen zu beschaffen.

Obwohl der Verein **eCH** all seine Sorgfalt darauf verwendet, die **eCH**-Standards sorgfältig auszuarbeiten, kann keine Zusicherung oder Garantie auf Aktualität, Vollständigkeit, Richtigkeit bzw. Fehlerfreiheit der zur Verfügung gestellten Informationen und Dokumente gegeben werden. Der Inhalt von **eCH**-Standards kann jederzeit und ohne Ankündigung geändert werden.

Jede Haftung für Schäden, welche dem Benutzer aus dem Gebrauch der **eCH**-Standards entstehen ist, soweit gesetzlich zulässig, wegbedungen.

## 6 Urheberrechte

Wer **eCH**-Standards erarbeitet, behält das geistige Eigentum an diesen. Allerdings verpflichtet sich der Erarbeitende sein betreffendes geistiges Eigentum oder seine Rechte an geistigem Eigentum anderer, sofern möglich, den jeweiligen Fachgruppen und dem Verein **eCH** kostenlos zur uneingeschränkten Nutzung und Weiterentwicklung im Rahmen des Vereinszweckes zur Verfügung zu stellen.

Die von den Fachgruppen erarbeiteten Standards können unter Nennung der jeweiligen Urheber von **eCH** unentgeltlich und uneingeschränkt genutzt, weiterverbreitet und weiterentwickelt werden.

**eCH**-Standards sind vollständig dokumentiert und frei von lizenz- und/oder patentrechtlichen Einschränkungen. Die dazugehörige Dokumentation kann unentgeltlich bezogen werden.

Diese Bestimmungen gelten ausschliesslich für die von **eCH** erarbeiteten Standards, nicht jedoch für Standards oder Produkte Dritter, auf welche in den **eCH**-Standards Bezug genommen wird. Die Standards enthalten die entsprechenden Hinweise auf die Rechte Dritter.

## Anhang A – Referenzen & Bibliographie

- [1] [http://www.re-wissen.de/opencms/Wissen/Techniken/Tagging\\_mit\\_Planguage.html](http://www.re-wissen.de/opencms/Wissen/Techniken/Tagging_mit_Planguage.html)
- [2] [http://en.wikipedia.org/wiki/Competitive\\_Engineering](http://en.wikipedia.org/wiki/Competitive_Engineering)
- [3] Rupp, C., SOPHISTen: Requirements-Engineering und Management. Professionelle, iterative Anforderungsanalyse für die Praxis. Carl Hanser Verlag München, 5., aktualisierte und erweiterte Auflage, 2009
- [4] <http://scrummethodology.com/scrum-user-stories/>
- [5] <http://www.dsmforum.org/>
- [6] <http://www.infoq.com/presentations/Truth-about-DSL>
- [7] eCH-0073: Dokumentation öffentlicher Leistungen und Prozesse (Dokumentationsstandard eGov CH)
- [8] Zachman, J.A., A framework for information systems architecture, IBM Systems Journal Vol. 26 No. 3 1987
- [9] Sowa, J.F., Zachman, J.A., Extending and formalizing the framework for information systems architecture, IBM Systems Journal Vol. 31 No. 3 1992
- [10] <http://www.zachmaninternational.com>
- [11] <http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf>
- [12] [http://www.omg.org/mda/mda\\_files/09-03-WP\\_Mapping\\_MDA\\_to\\_Zachman\\_Framework1.pdf](http://www.omg.org/mda/mda_files/09-03-WP_Mapping_MDA_to_Zachman_Framework1.pdf)
- [13] <http://www.opengroup.org/architecture/togaf8-doc/arch/chap39.html>
- [14] Fahney, R., Herrmann, A., Weissbach, R., Nutzen von systematischem Requirements Engineering & Management für das Projektmanagement der Entwicklung innovativer Produkte. In: Dorn, K.-H., Fitzsimons, C., Frick, A., Kerber, G., Marré, R., Wagenhals, K. (Hrsg.): Innovation durch Projektmanagement – oder?! Beiträge zur Konferenz „interPM“ Glashütten 2008, dpunkt verlag, Heidelberg, 2008
- [15] Bisher unveröffentlichte Ergebnisse der Fachgruppe „Requirements Management“ der GPM Deutschen Gesellschaft für Projektmanagement e.V. ([www.gpm-ipma.de](http://www.gpm-ipma.de)) und des Arbeitskreises „Requirements Engineering & Projektmanagement“ der Fachgruppe 2.1.6 „Requirements Engineering“ der (deutschen) Gesellschaft für Informatik e.V. ([www.repm.de](http://www.repm.de)).
- [16] [http://www.competence-site.de/downloads/c1/66/i\\_file\\_12206/GPM\\_Studie2004\\_summary\\_paconsult.pdf](http://www.competence-site.de/downloads/c1/66/i_file_12206/GPM_Studie2004_summary_paconsult.pdf)
- [17] [http://www.competence-site.de/downloads/f3/ae/i\\_file\\_12390/ProjektManagement\\_Kosten\\_Nutzen.pdf](http://www.competence-site.de/downloads/f3/ae/i_file_12390/ProjektManagement_Kosten_Nutzen.pdf)
- [18] Standish Group: The CHAOS Report, 1994, [http://www.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www.standishgroup.com/sample_research/chaos_1994_1.php) (nicht mehr online verfügbar, aber bei den Autoren dieser Best Practices einsehbar)



- [19] Standish Group: Unfinished Voyages - A Follow-Up to The CHAOS Report, 1995;  
[http://www.standishgroup.com/sample\\_research/unfinished\\_voyages\\_1.php](http://www.standishgroup.com/sample_research/unfinished_voyages_1.php) (nicht mehr online verfügbar, aber bei den Autoren dieser Best Practices einsehbar)
- [20] Standish Group: CHAOS: A Recipe for Success, 1999,  
[http://www.standishgroup.com/sample\\_research/PDFpages/chaos1999.pdf](http://www.standishgroup.com/sample_research/PDFpages/chaos1999.pdf) (nicht mehr online verfügbar, aber bei den Autoren dieser Best Practices einsehbar)
- [21] Standish Group: Extreme CHAOS, 2001,  
[http://www.standishgroup.com/sample\\_research/PDFpages/extreme\\_chaos.pdf](http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf) (nicht mehr online verfügbar, aber bei den Autoren dieser Best Practices einsehbar)
- [22] [http://www.certified-re.de/fileadmin/IREB/Services/ireb\\_cpre\\_syllabus\\_fl\\_de\\_v20.pdf](http://www.certified-re.de/fileadmin/IREB/Services/ireb_cpre_syllabus_fl_de_v20.pdf)
- [23] <http://www.sei.cmu.edu/reports/06tr008.pdf>
- [24] <http://ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/Releases/1.3/V-Modell-XT-Gesamt.pdf>
- [25] Hood, C., Wiebel, R.: Optimieren von Requirements Management & Engineering, Springer Verlag, Berlin, Heidelberg, New York, 2005
- [26] IEEE Std 610.12™-1990 IEEE Standard Glossary of Software Engineering Terminology
- [27] Kerkow, D., Dörr, J., Paech, B., Olsson, T., Koenig, T., "Elicitation and Documentation of Non-functional Requirements for Sociotechnical Systems". In: José Luis Maté, Andrés Silva, "Requirements Engineering for Sociotechnical Systems", Idea Group, Inc., 2004
- [28] Basili, V., Caldiera, G., Rombach, D., "The Goal Question Metric Approach", Wiley, 1994
- [29] IEEE: Std. 830-1998: IEEE Recommended Practice for Software Requirements Specification, IEEE, Washington, 1998
- [30] [http://repm.de/docspublic/SE2007workshop/Wieviel\\_RE\\_Stecktlm\\_SE\\_onlineversion.pdf](http://repm.de/docspublic/SE2007workshop/Wieviel_RE_Stecktlm_SE_onlineversion.pdf)
- [31] <http://www.industrialized-software.org/kiss-initiative>
- [32] <http://certified-re.de/>
- [33] Ebert, C.: Systematisches Requirements Engineering und Management: Anforderungen ermitteln, spezifizieren, analysieren und verwalten. dpunkt.verlag, Heidelberg, 2., aktualisierte und erweiterte Auflage 2008
- [34] Forsberg, K., Mooz, H.: System Engineering Overview. In: Software Requirements Engineering, eds.: Dorfman, M., Thayer, R., IEEE Comp. Soc. Press, Los Alamitos, USA, 1997
- [35] Forsberg, K., Mooz, H., The Relationship of System Engineering to the Project Cycle, presented at the joint conference sponsored by: National Council On Systems Engineering (NCOSE) and American Society for Engineering Management (ASEM), Chattanooga, TN, 21–23 October 1991,  
<http://www.csm.com/repository/model/rep/o/pdf/Relationship%20of%20SE%20to%20Proj%20Cycle.pdf>

[36] IABG Industrieanlagen-Betriebsgesellschaft mbH, V-Modell 97, [http://v-modell.iabg.de/index.php?option=com\\_docman&task=cat\\_view&gid=16&Itemid=30](http://v-modell.iabg.de/index.php?option=com_docman&task=cat_view&gid=16&Itemid=30)

[37] <http://en.wikipedia.org/wiki/V-Modell>

[38] Fahney, R., Anforderungs-orientierte Sicht auf IT-Architekturentwurf – eingeführt am Beispiel der Machbarkeitsanalyse. In: Hegering, H.-G., Lehman, A., Ohlbach, H.-J., Scheideler, C. (Hrsg.), Informatik 2008 Beherrschbare Systeme – dank Informatik, Beiträge der 38. Jahrestagung der Gesellschaft für Informatik e.V. (GI), Band 2, GI-Edition - Lecture Notes in Informatics (LNI), P-134, Bonner Köllen Verlag, Bonn, 2008

[39] Ehram, H., Fahney, R., McKeown, K., Requirements Engineering und Geschäftsprozessmodellierung - zwei Seiten der gleichen Medaille. In: Münch, J., Liggesmeyer, P. (Hrsg.), Software Engineering 2009 – Workshopband, GI-Edition - Lecture Notes in Informatics (LNI), P-150, Bonner Köllen Verlag, Bonn, 2009

## **Anhang B – Mitarbeit & Überprüfung**

Arbeitsgruppe REQMOD der eCH Fachgruppe HERMES  
Mitglieder von REQMOD (alphabetisch)

Autoren  
Philippe de Vallière  
Ralf Fahney  
Frank H. Ritz

## **Anhang C – Abkürzungen**

Abkürzungen sind erläutert zum Zeitpunkt des ersten Auftretens im Text

## **Anhang D – Glossar**

Begriffe sind definiert zum Zeitpunkt des ersten Auftretens im Text