



# Quo Vadis V-Modell

Qualitätssicherung und Qualitätsmanagement  
im V-Modell 97 und V-Modell XT

Thomas Binder  
Dietmar Winkler  
Stefan Biff

Vienna University of Technology  
Quality Software Engineering

Authors: **Thomas Binder**  
**Dietmar Winkler**  
**Stefan Biffl**

tomybinder@gmx.at  
{dietmar.winkler, stefan.biffl}@qse.ifs.tuwien.ac.at  
<http://qse.ifs.tuwien.ac.at>

Contact: **Vienna University of Technology**  
Institute of Software Technology & Interactive Systems (ISIS)

Favoritenstraße 9-11/188  
A-1040 Vienna  
Austria, Europe

Telephone: +43 1 58801 18810  
Telefax: +43 1 58801 18899  
Web <http://qse.ifs.tuwien.ac.at>

## Zusammenfassung

Komplexe Anforderungen an Softwareprojekte erfordern ein strukturiertes Vorgehen zur Planung, Kontrolle und Steuerung der Entwicklungsprozesse. Vorgehensmodelle unterstützen dieses strukturierte Vorgehen und ermöglichen einen methodischen Ansatz, um den Entwicklungsprozess transparenter, überschaubarer und nachvollziehbarer zu gestalten. Diese zunehmende Komplexität der Projekte erfordert eine intensive Zusammenarbeit zwischen unterschiedlichen Unternehmen und Institutionen. Die Verwendung einer abgestimmten gemeinsamen Vorgehensweise ist daher unumgänglich.

Das V-Modell<sup>®1</sup> – entwickelt im Auftrag von deutschen Bundesministerien – ist ein verpflichtendes Vorgehensmodell zur Abwicklung von Softwareprojekten im öffentlichen und militärischen Sektor in Deutschland und erfüllt somit diese Forderungen. Nach dem V-Modell 1992 und dem V-Modell 1997 steht nun das V-Modell XT in der aktuellen Version 1.2 zur Verfügung. Neben der Modularisierung stellt auch das Tailoring – das Anpassen der Projektdurchführung – einen wesentlichen Schritt zur Erhöhung der Flexibilität des Entwicklungsprozesses dar.

Diese Arbeit gibt einen kurzen Überblick der bisherigen Entwicklung des V-Modells unter besonderer Berücksichtigung der Themen Qualitätssicherung, Konfigurationsmanagement sowie Problem- und Änderungsmanagement.

**Schlagerworte** V-Modell, V-Modell XT, Qualitätssicherung, Qualitätsmanagement

---

<sup>1</sup>V-Modell<sup>®</sup> ist eine geschützte Marke der Bundesrepublik Deutschland.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Problemfeld Softwareentwicklung	1
1.2	Vorgehensmodelle in der Softwareentwicklung	2
1.3	Bedeutung des V-Modells	2
<b>2</b>	<b>Überblick über das V-Modell</b>	<b>3</b>
2.1	Ursprung des V-Modells	3
2.2	Ziele des V-Modells	3
2.3	Offene Nutzung	4
2.4	Normen	4
2.5	Tailoring	4
<b>3</b>	<b>Entwicklungsgeschichte</b>	<b>5</b>
3.1	V-Modell 92	5
3.2	V-Modell 97	5
3.2.1	Aufbau des V-Modell 97	6
3.2.2	Tailoring im V-Modell 97	6
3.3	V-Modell XT	7
3.3.1	Aufbau des V-Modell XT	7
3.3.2	Tailoring im V-Modell XT	9
<b>4</b>	<b>Qualitätsmanagement im V-Modell</b>	<b>9</b>
4.1	Qualitätssicherung – QS	9
4.1.1	QS im V-Modell 97	10
4.1.2	Vergleich zu QS im V-Modell XT	10
4.2	Konfigurationsmanagement – KM	11
4.2.1	KM im V-Modell 97	11
4.2.2	Vergleich zu KM im V-Modell XT	11
4.3	Problem- und Änderungsmanagement – PROB	12
4.3.1	PROB im V-Modell 97	12
4.3.2	Vergleich zu PROB im V-Modell XT	12
4.4	Verbesserungspotential Qualitätsmanagement	12
<b>5</b>	<b>Fazit</b>	<b>13</b>
	<b>Literaturverzeichnis</b>	<b>15</b>

# 1 Einleitung

Dieses Kapitel gibt einen Überblick über die Problematik der Softwareentwicklung, den Lösungsansatz durch Vorgehensmodelle allgemein und über die Bedeutung V-Modell.

## 1.1 Problemfeld Softwareentwicklung

Die Erstellung von Software ist ein hochkomplexer Vorgang, der leider nicht immer wie gewünscht durchgeführt werden kann, wie eine Studie der amerikanischen „Standish Group“ belegt. Im „Chaos-Report“ erfolgt alle zwei Jahre eine Befragung von IT Managern über deren Softwareprojekte<sup>2</sup>. Diese und die folgenden Zahlen stammen aus dem „Chaos Report 1994“ [18]:

- Nur 16,2 % der Projekte konnten wie geplant durchgeführt werden.
- 52,7 % sprengten entweder Zeit oder Kostenrahmen oder wurden nur in reduziertem Umfang zu Ende geführt.
- 31,3 % der Projekte mussten überhaupt abgebrochen werden.

Die häufigsten drei Gründe, warum Projekte scheitern sind dem Chaos-Report zufolge: Lückenhafte oder falsche Anforderungen (13.1%), mangelnde Einbindung des Auftraggebers (12.4%) oder mangelnde Ressourcen (10.6%). Diese und andere Faktoren tragen zu mangelnder Produktqualität bei. Qualität bedeutet hier, dem Kunden das zu liefern, was er haben möchte. Werden benötigte Funktionen einer Software fehlerhaft oder gar nicht umgesetzt, so wirkt sich das direkt auf die Softwarequalität aus. Dies wiederum nimmt Einfluss auf die Kundenzufriedenheit und somit im Weiteren auf Erfolg oder Misserfolg einer Firma. Um Qualität planen zu können, muss diese immer der kontrollierbar bleiben: Der Ist-Status muss immer dem definierten Soll-Status gegenüber gestellt werden können, sodass notwendige Schritte zur Erreichung der Qualitätsanforderungen rechtzeitig ergriffen werden können.

Qualität zu erreichen ist dennoch immer mit einem gewissen Trade-Off verbunden: Jedes Projekt ist ein Spannungsdreieck aus Zeit, Kosten und Qualität. Das bedeutet, dass ein Optimieren eines Faktors immer einen Einfluss auf die anderen Faktoren hat. Es ist nicht möglich alle drei Faktoren gleichzeitig zu verbessern. Soll beispielsweise ein Projekt schneller fertig werden, also die Zeit optimiert werden, so leiden darunter entweder Kosten oder Qualität oder beide. Liegt dagegen das Augenmerk auf Zeit und Qualität so steigen die Kosten stark. Der Einsatz von Prozessen und Methoden kann helfen, dieses Dreieck besser zu kontrollieren. [13]

---

<sup>2</sup>Befragt wurden Manager von großen Firmen mit einem durchschnittlichen IT Projektvolumen von 2.322.000 US-Dollar bzw. 1.975.000 Euro und mittleren Unternehmen (1.331.000 US-Dollar bzw. 1.130.000 Euro) als auch kleinen Unternehmen (434.000 US-Dollar bzw. 370.000 Euro). Kurs vom 06.12.2005 laut Yahoo Currency Converter (<http://finance.yahoo.com/currency>): 1 US-Dollar entspricht 0,85 EUR.

## 1 Einleitung

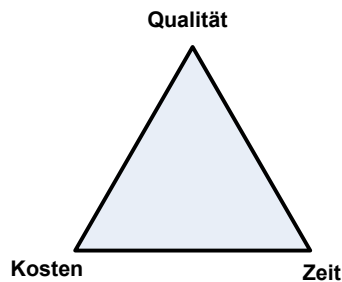


Abbildung 1: Softwareprojekt als Spannungsdreieck Zeit-Qualität-Kosten [13]

### 1.2 Vorgehensmodelle in der Softwareentwicklung

Um diesen Problemen entgegenzuwirken, werden bei der Durchführung von Softwareprojekten so genannte Vorgehensmodelle eingesetzt, um Planung und Ablauf des Projekts besser kontrollieren zu können. In einem solchen Vorgehensmodell wird der Lebenszyklus eines Softwareprojektes abgebildet: Alle Teilschritte, die im Laufe eines Projektes durchgeführt werden, haben einen definierten Platz, sodass sie in einer strukturierten Weise abgearbeitet werden können. Der sogenannte Software-Life-Cycle beinhalten Phasen wie „Anforderungen“, „Spezifikation“, „Implementierung“, „Integration“ und „Wartung“. [15]

Ein Vorgehensmodell legt dabei fest, welche Inhalte eine Phase hat, wann sie beginnen kann und welche Bedingungen erfüllt sein müssen, damit sie abgeschlossen werden kann. Auf diese Weise soll sichergestellt werden, dass keine Phase voreilig beendet wird und alle nötigen Schritte durchgeführt werden, um das Projekt erfolgreich durchführen zu können. Ein weiterer Vorteil eines Vorgehensmodells ist die Nachvollziehbarkeit: Da alle Schritte nach einem gewissen Plan durchgeführt werden, erfolgt der Ablauf sehr transparent, wodurch es ermöglicht wird, Erfahrungswerte zu sammeln und Teile eines Projektes wieder zu verwerten. [2]

### 1.3 Bedeutung des V-Modells

Das V-Modell ist nun ein solches Vorgehensmodell, das mit den beschriebenen Problemstellungen umgehen können soll. Es nimmt als Vorgehensmodell in Europa und vor allem in Deutschland eine besondere Stellung ein, da es in Deutschland im öffentlichen und militärischen Sektor zwingend vorgeschrieben ist. Möchte sich ein Unternehmen an einer Ausschreibung des öffentlichen oder militärischen Sektors in Deutschland teilnehmen, so muss dieses Unternehmen in der Lage sein, seinen Softwareentwicklungsprozess mit dem V-Modell konform durchzuführen.

Weiters ist eine Besonderheit des V-Modells, dass es durch seinen modularen Aufbau und seine starken Anpassungsmöglichkeiten für jede Projektgröße einsetzbar sein soll. Das V-Modell gilt als sehr ausgereift, da es sich bereits seit 1992 im praktischen Einsatz befindet und immer wieder in neuen Versionen aktuellen Rahmenbedingungen angepasst wird. Außerdem gibt es eine Reihe von Werkzeugen, welche die Arbeit und Organisation mit dem Modell vereinfachen. [10]

## 2 Überblick über das V-Modell

Das V-Modell ist ein Vorgehensmodell zur Erstellung von Software, das den Anspruch erhebt, mit den komplexen Problemen eines Software-Projektes umgehen zu können. Dieses Kapitel erläutert die Geschichte und Grundsätze des V-Modells.

### 2.1 Ursprung des V-Modells

Das V-Modell wurde im Auftrag des deutschen Bundesministeriums für Verteidigung (BMVg) und des deutschen Bundesamt für Wehrtechnik und Beschaffung (BWB) für militärische Softwareanwendungen entwickelt. Die Entwicklung begann 1986, im Jahr 1991 wurde der Einsatz des V-Modells für den gesamten wehrtechnischen Bereich in Deutschland verpflichtend. Das deutsche Bundesministerium des Inneren übernahm das Modell 1992 als „einheitlicher Standard für den gesamten öffentlichen Bereich“ [8], da im zivilen Sektor die Problemfelder bei der Softwareentwicklung ähnlich sind. Ziel war nicht nur ein Entwicklungsmodell für den öffentlichen Sektor zu erstellen, sondern man war auch immer bemüht, das V-Modell offen zugänglich zu machen, um so auch eine breite Akzeptanz in der Privatwirtschaft abseits von staatlichen Stellen zu schaffen. [2][8][9]

Das V-Modell hat seinen Namensursprung einerseits in der Kurzform für „Vorgehensmodell“, andererseits bildet die grafische Repräsentation des V-Modells den Buchstaben „V“. Abbildung 2 zeigt einen Ausschnitt der Systementwicklung aus dem V-Modell, der für die Namensgebung mitverantwortlich ist.

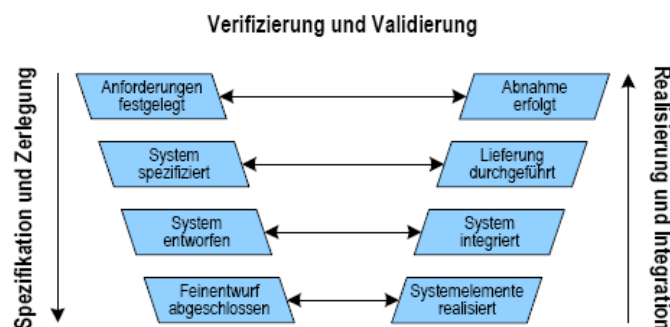


Abbildung 2: Die Form des Modells gibt dem V-Modell seinen Namen [4]

### 2.2 Ziele des V-Modells

Das V-Modell regelt das „Wer“, „Was“ und „Wann“ in einem Projekt. Die definierten Ziele des V-Modells sind [10]:

- Verbesserung der Kommunikation zwischen allen Beteiligten  
Durch Standardisierung von Begriffen und Bestandteilen des Projekts werden Reibungsverluste in der Kommunikation zwischen Projektbeteiligten gering gehalten.

Beispielsweise ist genau festgelegt, welche Aufgaben und Befugnisse jede Rolle im Projekt hat. So können Fehler aufgrund unterschiedlicher Interpretation von Begriffen durch verschiedene Projektmitarbeiter ausgeschlossen werden.

- Eindämmung der Gesamtkosten über den gesamten Projekt- und Systemlebenszyklus

Der Einsatz eines standardisierten Vorgehensmodells verspricht transparente Projekte, die sich leichter kalkulieren und steuern lassen.

- Verbesserung und Gewährleistung der Qualität

Das V-Modell definiert zahlreiche Zwischenergebnisse, die frühzeitig eine Qualitätskontrolle ermöglichen. Ergebnisse werden standardisiert, um so leichter überprüfbar zu sein.

- Minimierung der Projektrisiken

Durch eine standardisierte Vorgehensweise und Definition von Ergebnissen und Rollen ermöglicht das V-Modell hohe Transparenz und somit bessere Planbarkeit von Projekten. Auf diese Weise lassen sich Abweichungen vom Projektplan früher erkennen und Risiken minimieren.

### 2.3 Offene Nutzung

Das V-Modell darf von Ausbildnern und Anwendern lizenzfrei genutzt werden. Für die Verbreitung des V-Modells sowie die Anpassung ist der Erwerb von Lizenzen nötig, die jedoch kostenfrei sind. So kann das V-Modell weitgehend offen genutzt werden, wobei die Lizenzvergabe garantiert, dass auch angepasste Modelle den Qualitätsansprüchen des V-Modells genügen. [5]

Durch eine englische Übersetzung wird angestrebt, dass das V-Modell auch international, vor allem als europäisches Softwareentwicklungsmodell zu etablieren. [8]

### 2.4 Normen

Das V-Modell XT ist kompatibel zu Standards wie ISO 9001:2000 [16], für Qualitätsmanagementsysteme, ISO/IEC 15288 [17], das den Lebenszyklus von Prozessen abbildet und zu CMMI [6], das es ermöglichen soll, Softwareprojekte schnell und günstig abzuwickeln und gleichzeitig die Qualität hoch zuhalten. Ebenso wird der NATO Standard APAQ 150 [12] unterstützt, der bei militärischen Projekten oft vorgeschrieben ist. Das V-Modell XT beinhaltet Konventionsabbildungen, die vorgeben, wie diese Standards auf das V-Modell abgebildet werden können. [10]

### 2.5 Tailoring

Ein wichtiger Aspekt des V-Modells ist das so genannte Tailoring. Das V-Modell ist generisch aufgebaut, sodass es grundsätzlich für jede Art von Projekt geeignet ist. Um es



nun für ein konkretes Projekt zielgerichtet einsetzen zu können, besteht die Möglichkeit, irrelevante Teile des Modells wegzulassen, wobei das Fehlen von wichtigen Elementen verhindert werden soll. Das V-Modell 97 beschreibt sich selbst und das Tailoring wie folgt:

„Das V-Modell kann als Checkliste angesehen werden, die eine große Menge erprobter und sinnvoller Regelungen enthält. Für jedes Einzelprojekt wird aus diesem Angebot eine Teilmenge ausgewählt. Das V-Modell kann daher auch als ein ‚Baukasten‘ angesehen werden, aus dem im Rahmen des Tailoringvorgangs die für ein Projekt geeigneten Bausteine (Aktivitäten/Produkte) herausgenommen werden.“ *V-Modell 97: Teil 3: Handbuchsammlung [9]*

Typischerweise ist diese projektspezifische Anpassung einer der ersten und auch kritischsten Schritte des Anwenders in einem Projekt. Für alle Komponenten des V-Modells ist definiert, ob diese optional oder verpflichtend sind. Dies soll garantieren, dass keine Bestandteile, die für die Durchführung eines Projekts unverzichtbar sind, weggekürzt werden. Das V-Modell XT enthält mit dem „Projektassistenten“ zudem ein Tool, welches das Tailoring stark vereinfachen soll. [8][10]

## 3 Entwicklungsgeschichte

Dieser Abschnitt erklärt die Versionen des V-Modells sowie die wichtigsten Weiterentwicklungen im Vergleich zur jeweiligen Vorversion.

### 3.1 V-Modell 92

Mit dem V-Modell 92 wurde 1992 die erste zivile Version des V-Modells veröffentlicht. Als Vorteile nennt die offizielle Dokumentation [8] vor allem die Vorgehensstandardisierung und die damit verbundenen positiven Effekte auf Projekte: Bessere Kommunikation, einheitliches Vorgehen, höhere Produktqualität sowie höhere Produktivität durch reduzierten Aufwand bei Schulung und Einarbeitung. Des Weiteren sollte der Aufwand für Wartung deutlich reduziert werden.

Der Aufbau und das Tailoring des V-Modell 92 entsprechen im wesentlichen dem der Version 97. Da sich diese Arbeit auf die Unterschiede zwischen den Versionen 97 und XT konzentriert, wird hier nicht weiter auf den Aufbau des V-Modell 92 eingegangen.

### 3.2 V-Modell 97

Das V-Modell 97 aus dem Jahr 1997 ist die Weiterentwicklung des V-Modells 92. Die im Umgang mit dem Vorgehensmodell gesammelte Erfahrung wurde genutzt, um das V-Modell zu verbessern. Auf diese Weise sollte es nicht nur den aktuellen Anforderungen gerecht werden, es sollte auch die allgemeine Akzeptanz erhöht werden.

Neben vielen Änderungen im Detail nahm man in die Version von 1997 auch Regelungen für die Hardwareentwicklung auf, um in der Systementwicklung einem ganzheitlichen

Entwicklungsansatz entsprechen zu können. Die Dokumentation wurde im Volumen reduziert und neu organisiert. Neue Technologien, wie inkrementelle Softwareentwicklung und objektorientierte Vorgehensweise wurden integriert. Die Terminologie wurde an den ISO Standard 8402 angeglichen. [9]

### 3.2.1 Aufbau des V-Modell 97

Das V-Modell ist in der Version 97 grundsätzlich in vier Unterbereiche aufgeteilt. Diese Unterbereiche werden als Submodelle bezeichnet und umfassen „Softwareentwicklung“, „Qualitätssicherung“, „Konfigurationsmanagement“ und „Projektmanagement“. Jedes Projekt, das gemäß V-Modell 97 durchgeführt wird, enthält alle diese Submodelle (siehe Abbildung 3). Jedes Submodell enthält wiederum ein Vielzahl an Aktivitäten, in denen selbst die Abhängigkeiten untereinander definiert sind. Somit ergibt sich auch ein Fluss, in welcher Reihenfolge die Aktivitäten aufeinander folgen. Die dabei entstehenden Ergebnisse heißen Produkte. Das Submodell „Qualitätssicherung“ legt beispielsweise fest, dass die Aktivitäten „QS-Plan erstellen“ und „Prüfplan erstellen“ am Beginn der Qualitätssicherung erfolgen müssen, und die Produkte „QS-Plan“ und „Prüfplan“ als Ergebnisse liefern. Weiterführende Aktivitäten, welche die Prüfungsvorbereitung oder die Prüfung selbst betreffen, können erst im Anschluß durchgeführt werden. [9]

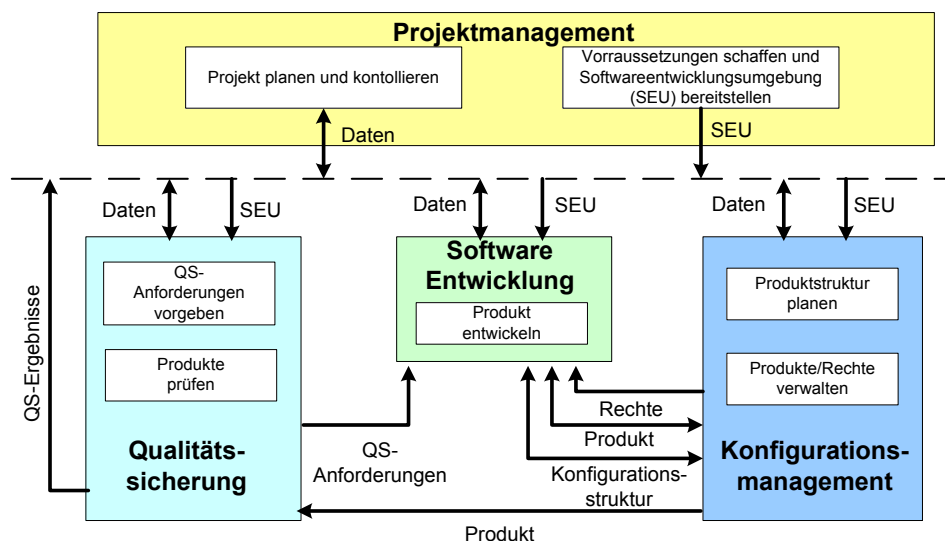


Abbildung 3: Zusammenspiel der Submodell im V-Modell 97 [9]

### 3.2.2 Tailoring im V-Modell 97

Die Anpassung des V-Modells erfolgt im Tailoring. Im V-Modell 97 ist es nicht möglich, ganze Submodelle wegzulassen, stattdessen werden einzelne Aktivitäten innerhalb eines Submodells gestrichen. Um diesen sehr komplexen Vorgang etwas zu vereinfachen, existie-

ren für sechs Projekttypen<sup>3</sup> standardisierte „Vortailorings“, die einen Teil der benötigten Anpassungen vorweg nehmen. Beim Tailoring an sich sind zahlreiche Streichbedingungen zu beachten, die festlegen, welche Teile des V-Modells weggelassen werden dürfen ohne dabei die Durchführbarkeit eines Projekts zu gefährden. Das V-Modell schreibt vor, alle Schritte des Tailorings zu dokumentieren, um auch diesen Vorgang transparent und nachvollziehbar zu halten.

## 3.3 V-Modell XT

Das V-Modell XT wurde Anfang 2005 veröffentlicht, um erneut auf geänderten Anforderungen reagieren zu können. In erster Linie sollte jungen Entwicklungen der Softwarebranche – wie Agile Softwareentwicklung oder Extreme Programming – Rechnung getragen werden. Außerdem sollte die leider immer noch relativ unübersichtliche Dokumentation weiter vereinfacht werden. Auch dem Tailoring wurde mehr Bedeutung zugemessen – soviel, dass es sogar im Namen der neuen Version festgeschrieben wurde: XT steht für eXtreme Tailoring bzw. eXtendable Tailoring. Die Version XT ist außerdem stark dahingehend erweitert worden, dass auch die Ausschreibung eines Projektes möglichst breit abgedeckt werden soll – dies ist vor allem in Hinblick auf öffentliche Ausschreibungen interessant, die vom V-Modell als Teil des Projekts berücksichtigt werden. Vor allem für nicht IT-Personen wird es so möglich, Projekte und Projektausschreibungen zu leiten. Für die aktuelle Version gibt es darüber hinaus – viel mehr als für die vorhergehenden Versionen – Sammlungen von Vorlagen, Schulungsunterlagen, wie beispielsweise interaktive Lerntouren und auch ein eigenes Customizing Tool für das Tailoring. Diese Unterlagen sollen den Einstieg in das V-Modell erleichtern und ständig erweitert werden. Des Weiteren wurde auch festgeschrieben, dass das V-Modell laufend einem Erneuerungsprozess unterliegen soll, um auf aktuellem Stand zu bleiben. [7][10][14]

### 3.3.1 Aufbau des V-Modell XT

Anders als die Version 97 unterscheidet das V-Modell XT zwischen den Projekttypen der Systementwicklung für Auftragnehmer<sup>4</sup>, der Systementwicklung für Auftraggeber<sup>5</sup>, der Systementwicklung in Eigenentwicklung<sup>6</sup> sowie der „Einführung und Pflege eines organisationsspezifischen Vorgehensmodells“. Dies ist eine wichtige Neuerung, da so Auftraggeber- und Auftragnehmerseite eines Projekts erstmals eigens abgedeckt werden. Die Einteilung in unterschiedliche Projekttypen nimmt bereits einen großen Teil der projektspezifischen Anpassung vorweg. [10]

Auch der Bereich der Submodelle wurde grundsätzlich überarbeitet. Es gibt nun 21 Vorgehensbausteine, welche die Funktion der Submodelle in der Vorgängerversion erfüllen.

---

<sup>3</sup>Diese Projekttypen sind „kleine administrative IT-Vorhaben“, „mittlere administrative IT-Vorhaben“, „große administrative IT-Vorhaben“, „kleine/mittlere technisch-wissenschaftliche IT-Vorhaben“, „große technisch-wissenschaftliche IT-Vorhaben“ sowie „Auswahl, Beschaffung und Anpassung von Fertigprodukten“.

<sup>4</sup>offizielle Bezeichnung: „Systementwicklungsprojekt (AN)“

<sup>5</sup>offizielle Bezeichnung: „Systementwicklungsprojekt (AG)“

<sup>6</sup>offizielle Bezeichnung: „Systementwicklungsprojekt (AN/AG)“

### 3 Entwicklungsgeschichte

Vier dieser Bausteine müssen in jedem Projekt enthalten sein und werden daher auch als „V-Modell-Kern“ bezeichnet. Dieser Kern soll garantieren, dass ein Mindestmaß an Projektdurchführungsqualität vorhanden ist und umfasst folgende Vorgehensbausteine:

- Problem- und Änderungsmanagement
- Qualitätssicherung
- Konfigurationsmanagement
- Projektmanagement

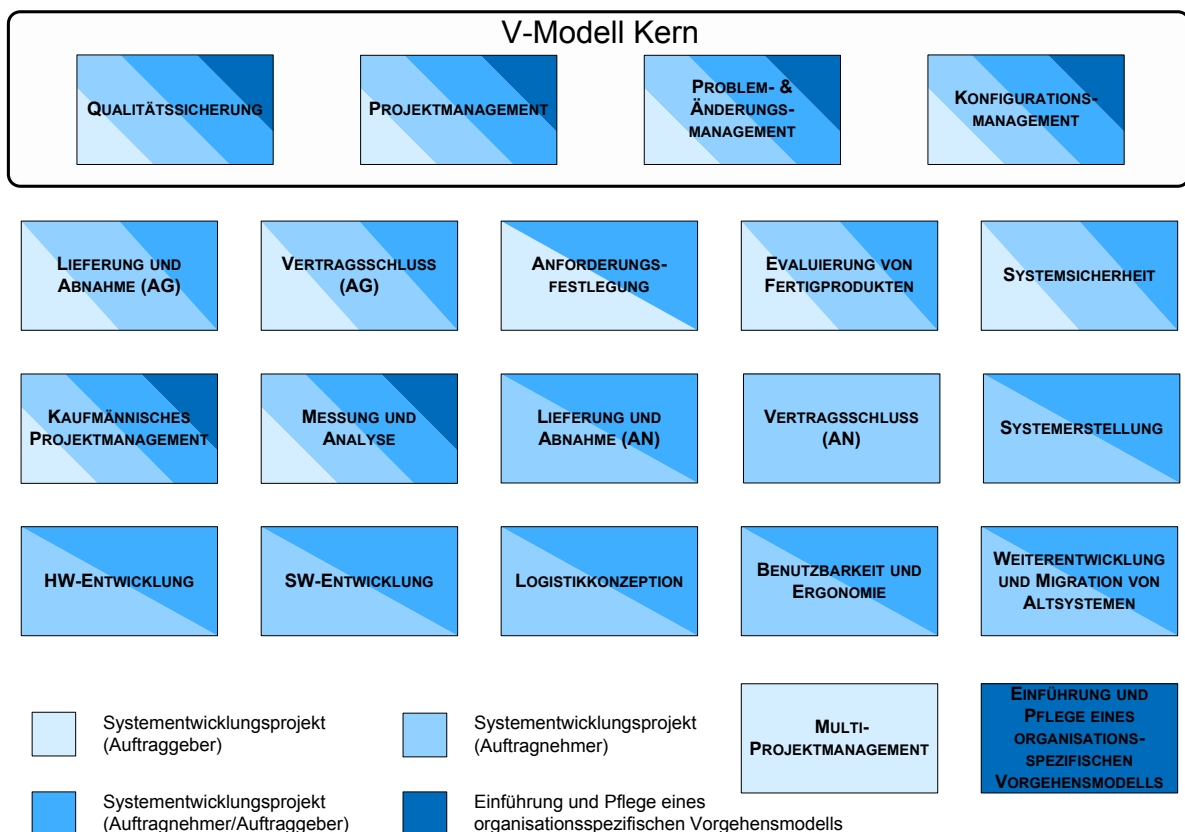


Abbildung 4: Die Vorgehensbausteine des V-Modell XT, Erweiterung zu [11]

Neben diesem Kern existieren noch 17 weitere Vorgehensbausteine, die in Abbildung 4 zu sehen sind. Die Farbe der Bausteine zeigt die Zugehörigkeit der einzelnen Vorgehensbausteinen zu den verschiedenen Projekttypen.

Die in den Vorgehensbausteinen enthaltenen Aktivitäten und Produkte wurden ebenfalls einem Redesign unterzogen. So wurden sie teilweise anders auf die Bausteine verteilt und inhaltlich überarbeitet. Die vergleichsweise kurzen Beschreibungen in der Vorversion sind längeren Texten gewichen, die nicht nur verständlicher sondern auch technisch auf dem aktuellen Stand sind. [10]

### 3.3.2 Tailoring im V-Modell XT

Zur Anpassung des Modells an die speziellen Anforderungen eines Projekts wird auch hier Tailoring eingesetzt. Dadurch, dass viele der Vorgehensbausteine optional bzw. mit einem der vier definierten Projekttypen verknüpft sind, kann das Tailoring bereits auf einer allgemeineren Ebene stattfinden, bevor man innerhalb von verwendeten Vorgehensbausteinen einzelne Aktivitäten entfernt. Zusätzlich sind die Abhängigkeiten von Vorgehensbausteinen und Aktivitäten untereinander definiert, sodass sicher gestellt werden kann, dass keine zwingend benötigten Elemente entfernt werden. Um den Vorgang des Tailorings weiter zu erleichtern, enthält die Dokumentation des V-Modells XT mit dem „V-Modell XT Projektassistent“ ein Werkzeug, das die Anpassung überwacht und für das jeweilige Projekt eine maßgeschneiderte Dokumentation erzeugt. [10]

## 4 Qualitätsmanagement im V-Modell

Qualitätsmanagement soll sicherstellen, dass das Ergebnis eines Projekts den Qualitätsanforderungen genügt. Zu diesem Zweck müssen Methoden zur Verfügung stehen, die es ermöglichen die Qualität zu messen und zu kontrollieren und gegebenenfalls Maßnahmen setzen, falls die Qualität unter den Anforderung liegt. Im V-Modell findet das Qualitätsmanagement hauptsächlich in den Submodellen (Version 97) bzw. den Vorgehensbausteinen (Version XT) „Qualitätssicherung – QS“ und „Konfigurationsmanagement – KM“ sowie dem nur in der Version XT enthaltenem Vorgehensbaustein „Problem- und Änderungsmanagement – PROB“ statt.

Dieses Kapitel stellt diese Bausteine des V-Modells vor, wobei zuerst der Ansatz an sich beschrieben wird, dann die Implementierung im V-Modell 97 und anschließend auf die Änderungen durch das V-Modell XT eingegangen wird. [9][10]

Eine detaillierte Darstellung der Übertragung des V-Modells 97 auf die Version XT befindet sich in der „Konventionsabbildung“ in der offiziellen Dokumentation [10].

### 4.1 Qualitätssicherung – QS

Bei der Qualitätssicherung im V-Modell wird grundsätzlich zwischen konstruktiven und analytischen Maßnahmen unterschieden. Bei den konstruktiven Maßnahmen geht es um die Vermeidung von qualitätsrelevanten Risiken, um das Entgegenwirken von Qualitätsmängeln, sowie darum, sicherzustellen, dass alle Prüfgegenstände auch wirklich prüfbar sind. Dies erfolgt beispielsweise durch die Anwendung von Standards oder Checklisten. Bei den analytischen Maßnahmen werden im Nachhinein Prüfungen und Bewertungen durchgeführt, um somit die Qualität nachweisen zu können und werden zum Beispiel durch Prüftechniken wie Reviews oder Tests von Systemteilen durchgeführt. Analytische Methoden kommen meist dort zum Einsatz, wo es besonders wichtig ist, gewisse Qualitätsanforderungen zu erfüllen, beispielsweise in sicherheits- oder zeitkritischen Anwendungen. Die Qualitätskontrollen erfolgen dabei immer in Selbstprüfung, d.h. derjenige der ein Produkt bearbeitet, überprüft dieses. Sofern es im Qualitätssicherungsplan vorgesehen ist, erfolgt anschließend eine formale Überprüfung. Diese formale

Prüfung erfolgt nie durch Ersteller eines Produkts selbst, sodass das Vier-Augen-Prinzip zur Anwendung kommt: mindestens zwei Personen haben das Produkt überprüft. [9][10]

### 4.1.1 QS im V-Modell 97

Im V-Modell 97 werden alle Maßnahmen zur Sicherung der Qualität im Submodell „Qualitätssicherung“ festgelegt. Die Hauptaktivitäten sind dabei „QS-Initialisierung“, in der ein QS-Plan erstellt wird, „Prüfungsvorbereitung“, bei der Prüfplan, Prüfspezifikation und Prüfprozedur definiert werden, „Prozessprüfung von Aktivitäten“ und „Produktprüfung“, bei der jeweils Prüfungsprotokolle erstellt werden, sowie „QS-Berichtswesen“, welches Berichtsdokumente erzeugt. Aktivitäten und Produkte, die für einzelne Projekte nicht benötigt werden, werden innerhalb dieses Submodells per Tailoring entfernt. [9]

Während alle analytischen Maßnahmen auch innerhalb dieses Submodells durchgeführt werden, erfolgt die Durchführung der konstruktiven Maßnahmen im Submodell „Softwareentwicklung“. Das heißt, dass die Kontrolle durch die Qualitätssicherung nach Fertigstellung<sup>7</sup> erfolgt, im Gegensatz zur konstruktiven Qualitätssicherung, die während der Softwareentwicklung selbst angewandt wird. [9]

### 4.1.2 Vergleich zu QS im V-Modell XT

Durch die Neuorganisierung der vier Submodelle in 21 Vorgehensbausteine, hat der Vorgehensbaustein „Qualitätssicherung“ im Umfang deutlich abgenommen, ohne jedoch generell an Bedeutung zu verlieren. In ihm sind nur noch jene Aktivitäten enthalten, die in jedem Projekt benötigt werden. Während die neue Aktivität „Projekt planen“ – bisher ein Teil der umfangreicheren Aktivität „QS-Initialisierung“ – in den Vorgehensbaustein „Projektmanagement“ des V-Modell Kerns verschoben wurde, wurden weitere Qualität sichernde Aktivitäten, die für spezielle Projekte relevant sein können, in folgende Vorgehensbausteine übertragen: „Systemerstellung“, zuständig für die Erstellung eines Systems und der zugehörigen Unterstützungssysteme, „Benutzbarkeit und Ergonomie“, mit Fokus auf die Mensch-Maschine Schnittstelle sowie „Lieferung und Abnahme (AG)“, das die Lieferung und Abnahme von Auftraggeberseite durch Prüfspezifikationen definiert. [10]

Es wurde jedoch nicht nur die Einordnung der Aktivitäten verändert, sondern es wurden auch die Inhalte neu ausgerichtet. Als Beispiel sei hier die Aktivität „Produktprüfung“ genannt: Während in der Version 97 allgemein die Prüfbarkeit festgestellt wurde und wie das Produkt inhaltlich geprüft werden sollte, erfolgt im V-Modell XT eine Aufteilung nach den zu prüfenden Objekten. So lauten die Aktivitäten nun „Dokument prüfen“, „Systemelement prüfen“, „Benutzbarkeit prüfen“ und „Lieferung prüfen“. Auch die Produkte wurden neu aufgeteilt und ausgerichtet. Ein Beispiel dafür ist das „Prüfprotokoll“, welches in insgesamt fünf neue Produkte aufgeteilt wurde: „Prüfprotokoll Systemelement“, „Prüfprotokoll Benutzbarkeit“, „Prüfprotokoll Prozess“, „Prüfprotokoll Lieferung“ sowie „Prüfprotokoll Dokument“. [10]

---

<sup>7</sup>Hierbei handelt es sich nur um eine vorläufige Fertigstellung. Erst wenn das Qualitätsmanagement seine Prüfungen positiv durchgeführt hat, kann ein Produkt tatsächlich als fertiggestellt angesehen werden.

## 4.2 Konfigurationsmanagement – KM

Ziel eines Konfigurationsmanagement ist es, einen Überblick darüber zu haben, in welche Produktversion welche Unterprodukte eingeflossen sind. In Programmen ist zum Beispiel wichtig zu wissen, welche Version einer Programmbibliothek mit welcher Version eines Programmteils erstellt wurde, sodass immer nachvollziehbar bleibt, aus welchen Bausteinen – und vor allem mit welcher Version der Bausteine – ein Programm getestet oder gar ausgeliefert wurde. Auf diese Weise können Unterschiede und Zusammenhänge verschiedener Versionen und Konfigurationen sichtbar werden und auch nur so können Produktänderungen kontrolliert erfolgen. Konfigurationsmanagement kümmert sich jedoch nicht nur um Versionen von Programmen, auch die Versionen von Dokumenten werden mit eingebunden, sodass transparent bleibt, welche Entscheidung im Projekt auf welche Version eines Dokumentes zurückzuführen ist. [9][10]

Um Konfigurationsmanagement im V-Modell zu ermöglichen, haben alle Produkte einen von vier Zuständen: Der Anfangszustand von jedem Produkt ist „geplant“, d.h. das Produkt ist für das jeweilige Projekt vorgesehen, wurde aber noch nicht begonnen. Sobald die Arbeit an einem Produkt startet, nimmt es den Zustand „in Bearbeitung“ an. Ist die Bearbeitung an einem Produkt abgeschlossen, so kommt es zur Qualitätssicherung im Status „vorgelegt“. Anschließend wird von der Qualitätssicherung einerseits die Prüfbarkeit getestet, andererseits die inhaltliche Qualität. Je nach Ausgang der Überprüfung wird der Status nun wieder auf „in Bearbeitung“ zurückgesetzt oder der Status „akzeptiert“ wird vergeben und das Produkt ist abgeschlossen. Sollte es zu einem späteren Zeitpunkt notwendig sein, eine neue Version eines Produktes zu erstellen – beispielsweise auf Grund neuer Anforderungen – so kommt wieder das Konfigurationsmanagement ins Spiel. Es muss sowohl eine neue Versionsnummer vergeben, als auch die notwendige Dokumentation erstellt werden. [9][10]

### 4.2.1 KM im V-Modell 97

Das Submodell „Konfigurationsmanagement“ enthält vier Hauptaktivitäten: „KM-Planung“, welche die KM-Tätigkeiten eines Projektes plant und einen KM-Plan erstellt, „Produkt- und Konfigurationsverwaltung“, die so genannte „Konfigurations-Identifikationsdokumente“ erstellt, das „Änderungsmanagement (Konfigurationssteuerung)“, das Änderungsvorschläge bewertet und über Änderungen entscheidet, sowie „KM-Dienste“, die Infrastruktur- sowie Dokumentationsaufgaben erfüllen. [9]

### 4.2.2 Vergleich zu KM im V-Modell XT

Wie schon beim QS ist auch beim KM das Submodell auf mehrere Vorgehensbausteine aufgeteilt worden. Die Aufteilung erfolgte jedoch ausschließlich auf Vorgehensbausteine des V-Modell-Kerns: „Konfigurationsmanagement“, „Problem- und Änderungsmanagement“ und „Projektmanagement“. Auf Seite der Produkte und Aktivitäten wurden nur die wenigsten umbenannt oder neu aufgeteilt, die Struktur ist im Großen und Ganzen die selbe geblieben. [10]



Auf Grund der technischen Weiterentwicklung und der breiten Unterstützung durch verschiedenen Tools benötigt das Konfigurationsmanagement heutzutage generell weniger Aufwand in Projekten. Daher ist es auch im V-Modell XT weit weniger umfangreich als noch im V-Modell 97. [10]

### 4.3 Problem- und Änderungsmanagement – PROB

Im Problem- und Änderungsmanagement geht es darum, wie mit Problemen und Änderungswünschen umgegangen wird. Oftmals sind Änderungen notwendig, da sich das Umfeld oder die Bedürfnisse der Benutzer verändern können oder auch Fehler in einem Produkt gefunden werden. Der leider meist üblichen Situation, dass Änderungswünsche an den Entwickler von unterschiedlicher Seite herangetragen werden und diese dann Änderungen nach eigenem Ermessen durchführen, wird ein strukturierter Prozess entgegengesetzt, in dem jeder Änderungswunsch dokumentiert und geprüft wird. Erst wenn sich nach einer eingehenden Prüfung ein Änderungswunsch als sinnvoll – auch im wirtschaftlichen Sinn des Projekts – erwiesen hat, werden Änderungen zur Durchführung freigegeben. [9][10]

#### 4.3.1 PROB im V-Modell 97

Problem- und Änderungsmanagement ist im V-Modell 97 nicht in einem eigenen Submodell enthalten. Es existieren jedoch zwei Aktivitäten, die diese Funktion erfüllen: „Änderungsmanagement (Konfigurationssteuerung)“, das sich im Submodell KM befindet, sowie „Software-Pflege und -Änderung (SWPÄ)“ im Submodell „Softwareentwicklung“. [9]

#### 4.3.2 Vergleich zu PROB im V-Modell XT

Im V-Modell XT ist ein eigener Vorgehensbaustein für Problem- und Änderungsmanagement eingeführt worden, der auch zum V-Modell-Kern gehört. In diesem haben die damit verbundenen Aktivitäten des V-Modell 97 einen Platz gefunden und sind auf den aktuellen Stand der heutigen Softwareentwicklungs-Methodik gebracht worden. [10]

### 4.4 Verbesserungspotential Qualitätsmanagement

Während der Erstellung dieser Arbeit konnten weder in der Dokumentation zum V-Modell selbst, noch in der Literatur, fehlende oder unterentwickelte Aspekte im Qualitätsmanagement gefunden werden. Durch den Einsatz und die Erprobung über 13 Jahre in der Praxis, hat das V-Modell einen großen Reifegrad erreicht, der das Spektrum des Qualitätsmanagement denkbar gut abdeckt.

Natürlich darf man sich hier nicht auf dem Erreichten ausruhen, sondern muss die Änderungen der Anforderungen an Qualität und Entwicklungsmodelle weiter beobachten und auch die Erkenntnisse dieser Beobachtungen weiter in das V-Modell einfließen lassen.



## 5 Fazit

Das V-Modell hat ohne Frage einen weiteren großen Schritt nach vorne gemacht. Die größte Kritik am V-Modell 97 betraf hauptsächlich den große Overhead an Managementfunktionen und die teilweise riesigen Mengen an anfallenden Papieren und Dokumenten, bedingt durch mangelhafte Kenntnisse am V-Modell und suboptimalem Tailoring. [13] Das V-Modell XT versteht es hier durch die Definition von vier unterschiedlichen Projekttypen und dem Reorganisieren der vier Submodelle in 21 Vorgehensbausteine einen viel modulareren Ansatz zu schaffen, der das Tailoring deutlich vereinfacht – nicht zuletzt auch durch das Bereitstellen von Tools. Die explizite Berücksichtigung der Auftragnehmerseite und das Miteinbeziehen der Ausschreibungsphase in das V-Modell ermöglichen es außerdem, bereits wirklich von Anfang an einen strukturierten und nachvollziehbaren Prozess einsetzen zu können.

Auf Seite der Qualitätssicherung muss gesagt werden, dass auch hier die Neuaufteilung sehr sinnvoll erscheint, vor allem die höhere Granularität verspricht das Tailoring zu erleichtern und zu verbessern. Die Neuschaffung des verbindlichen Vorgehensbausteins „Problem- und Änderungsmanagement“ ist daher ebenfalls sehr positiv zu bewerten und zeigt, dass erkannt worden ist, wie wichtig dieser Aspekt in Projekten sein kann.

Die übrigen Änderungen beziehen sich größtenteils auf Entwicklungen, die in der Softwareindustrie seit Verabschiedung des V-Modell 97 passiert sind: Vor allem neue Techniken sowie neue Methoden in der Entwicklung von Software. Dieses „Update“ war sehr wichtig, da nur so gewährleistet werden kann, dass das V-Modell für aktuelle Projekte einsetzbar bleibt.

Alles in allem ist das V-Modell XT ein sehr viel versprechendes Modell. Dennoch muss an dieser Stelle gesagt werden: Ein guter Prozess garantiert noch kein gutes Produkt, aber er erhöht die Wahrscheinlichkeit dafür enorm. Wichtig sind vor allem auch die Menschen, die mit dem V-Modell arbeiten: Gerade bei kritischen Tätigkeiten wie dem Tailoring sind gut ausgebildete Mitarbeiter entscheidend.

Ob sich das V-Modell XT in Zukunft bewähren und behaupten kann, wird sich erst zeigen, wenn ausreichend Erfahrung über damit durchgeführte Projekte vorliegt. Da die Version XT noch relativ jung ist, wird man auf aussagekräftige Studien noch einige Zeit warten müssen.

Spannend ist auch die Frage, des internationalen Erfolges: Wird es dem V-Modell gelingen, international Akzeptanz zu finden? Je weiter sich die angekündigte englische Version noch verzögert<sup>8</sup> desto länger muss man auch auf internationale Anerkennung warten.

Ein anderer interessanter Faktor ist die laufende Fortschreibung: Bereits für das V-Modell 92 und das V-Modell 97 war geplant, die Modelle ständig fortzuschreiben und somit immer aktuell zu halten. Da dies bisher nicht gelungen ist, scheint es auch fraglich, ob es für das V-Modell XT gelingen wird. Nicht zuletzt auch deswegen, da man im Zeitplan – wie auch bei der englischen Version – deutlich hinterherhinkt: Die für

---

<sup>8</sup>Geplant war die englische Version ursprünglich für August 2005, ist inzwischen nur das V-Modell XT Grundlagenkapitel im Februar 2006 erschienen. Wann eine vollständige Übersetzung erscheinen soll, ist nicht bekannt. [1][19]

August 2005 angekündigte Version 1.2 ist nun erst im Februar 2006 erschienen. Eine Verschiebung, die – angesichts eines geplanten halbjährlichen Veröffentlichungszykluses – doch sehr groß erscheint. [1][19]

Ideen wie man das V-Modell weiterentwickeln könnte gibt es einige: [11] sieht einen immer noch zu großen Dokumentations- und Organisationsaufwand bei sehr kleinen Projekten. Andere Ideen umfassen Feedbackzyklen im Tailoring anstelle des einmaligen Tailorings nur am Beginn eines Projekts, verstärkte Methodenunterstützung oder einen Produktlinienansatz, wie er unter [3] vorgeschlagen wird. Auch hier muss sich erst zeigen, ob zukünftige Versionen des V-Modells diese Ideen aufgreifen und umsetzen.

## Literatur

- [1] BARTELT, CHRISTIAN: *KBSt-Foren*, 2005. <http://foren.bund.de/showthread.php?threadid=221#post1204>, Posting in einem Online Forum, Online Ressource, Abruf: 12. 12. 2005.
- [2] BENAD, JOHN: *Projektmanagement und Qualitaetssicherung unter dem V-Modell-XT*. Diplomarbeit, Technische Universität Dresden, Institut für Software- und Multimediatechnik, Dresden, Dezember 2004.
- [3] BIFFL, STEFAN, DIETMAR WINKLER, REINHARD HÖHN und HERBERT WETZEL: *Software Process Improvement in Europe: Potential of the new V-Model XT and Research Issues*. In: *European Software Process Improvement and Innovation Conference, Experience Track*. Wiley, November 2005.
- [4] BROY, MANFRED und ANDREAS RAUSCH: *Das neue V-Modell XT – Ein anpassbares Vorgehensmodell für Software und System Engineering*. Informatik Spektrum, 28(3):220–229, Juni 2005.
- [5] BUNDESVERWALTUNG, KOORDINIERUNGS UND BERATUNGSSTELLE DER BUNDESREGIERUNG FÜR INFORMATIONSTECHNIK IN DER: *Verwendungsmodell des V-Modell XT*, 2005. <http://www.kbst.bund.de/V-Modell/Nutzung-und-Lizenzen-494/Verwendungsmodell.htm>, Online Ressource, Abruf: 18. 02. 2006.
- [6] CARNEGIE MELLON SOFTWARE ENGINEERING INSTITUTE: *CMMI Main Page*, 2006. <http://www.sei.cmu.edu/cmmi/>, Online Ressource, Abruf: 18. 02. 2006.
- [7] HÖHN, REINHARD: *Das V-Modell: Fortsetzung einer Erfolgsstory*. OCG Journal, 5/2004:19–20, Dezember 2004.
- [8] IAGB: *Vorgehensmodell: Kurzbeschreibung*, 1995. [http://www.v-modell.iabg.de/kurzb/vm/k\\_vm\\_d.doc](http://www.v-modell.iabg.de/kurzb/vm/k_vm_d.doc), Online Ressource, Abruf: 09. 12. 2005.
- [9] IAGB: *V-Modell 97*, 1997. <http://www.v-modell.iabg.de/>, Online Ressource, Abruf: 09. 12. 2005.
- [10] KOORDINIERUNGS- UND BERATUNGSSTELLE DER BUNDESREGIERUNG FÜR INFORMATIONSTECHNIK IN DER BUNDESVERWALTUNG: *V-Modell XT CD Release 1.2*, 2006. CD-ROM, Download auf <http://www.kbst.bund.de/V-Modell/-469/V-Modell-XT.htm>.
- [11] MESZAR, WOLFGANG: *V-Modell XT als methodischer Ansatz in der Entwicklung einer Webtools – ein Erfahrungsbericht*. Praktikumsarbeit, TU Wien, Institut für Softwaretechnik und interaktive Systeme, 2006.
- [12] NATO: *APAQ-150 – NATO Quality Assurance Requirements for Software Development*, 1997. <http://www.nato.int/docu/stanag/aqap150/aqap150e.htm>, Online Ressource, Abruf: 20. 02. 2006.

## Literatur

- [13] PETRASCH, ROLAND: *Einführung in das Software-Qualitätsmanagement*. Logos Verlag Berlin, 1998.
- [14] RAUSCH, ANDREAS und DIRK NIEBUHR: *Erfolgreiche IT-Projekte mit dem V-Modell XT*. OBJEKTSpektrum, 3/2005:42–49, Mai 2005.
- [15] SOMMERVILLE, IAN: *Software Engineering*. Addison Wesley, 6th Edition, 2001.
- [16] STANDARDIZATION, INTERNATIONAL ORGANIZATION FOR: *ISO 9001:2000: Quality management systems – Requirements*, 2000. <http://www.iso.org/>, Online Ressource, Abruf: 18.02.2006.
- [17] STANDARDIZATION, INTERNATIONAL ORGANIZATION FOR: *ISO/IEC 15288:2002: Systems engineering – System life cycle processes*, 2002. <http://www.iso.org/>, Online Ressource, Abruf: 18.02.2006.
- [18] STANDISH GROUP: *The Chaos Report*. The Standish Group International Inc., 1994. [http://www.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www.standishgroup.com/sample_research/chaos_1994_1.php), Online Ressource, Abruf: 03.12.2005.
- [19] WAGNER, DIETRICH: *Das neue V-Modell XT: Maßgeschneidertes Projektvorgehen für Auftraggeber und Auftragnehmer*. PENTASYS BLICKPUNKTE, 07/05:1–3, Juni 2005.