

# Metamodellbasierte Integration von Projekt Controlling Mechanismen in das V-Modell XT – Positionspapier –

Marco Kuhrmann<sup>1</sup>, Jürgen Münch<sup>2</sup>, Andreas Rausch<sup>3</sup>

<sup>1</sup>Technische Universität München, Institut für Informatik, I4  
Boltzmannstr. 3, 85748 Garching, Germany, kuhrmann@in.tum.de

<sup>2</sup>Fraunhofer IESE, Fraunhofer-Platz 1,  
67663 Kaiserslautern, Germany, muench@iese.fraunhofer.de

<sup>3</sup>Technische Universität Kaiserslautern,  
FB Informatik, AG Softwarearchitektur, Erwin-Schrödinger-Straße,  
rausch@informatik.uni-kl.de

**Abstract:** Die Messung von Kennzahlen zur Bestimmung des Projektstatus ist ein essenzielles Werkzeug zur Kontrolle von Softwareentwicklungsprojekten. Deshalb wurden durch viele Organisationen Methoden, Metriken usw. zur Messung eingeführt. Trotzdem sind immer noch Diskrepanzen hinsichtlich des Verständnisses zwischen verschiedenen Ebenen einer Organisation, z.B. zwischen ausführenden Abteilungen und dem Management, zu verzeichnen. Diese Diskrepanzen sind nicht nur vertikal (zwischen Hierarchien), sondern auch horizontal (zwischen verschiedenen Projekten oder Organisationseinheiten) festzustellen – Projekte sind trotz normierter Prozesse oftmals nicht vergleichbar. Dies erschwert angemessene Reaktionen, insbesondere wenn es zu kritischen oder unvorhergesehenen Situationen kommt. Dieser Beitrag analysiert Potenziale für eine Integration von Mess- und Kontrollverfahren in das V-Modell<sup>®</sup> XT.

## 1 Einleitung

Die Erstellung von Softwaresystemen ist immer noch voller Risiken und Unvorhersehbarkeiten. Studien zufolge scheitern etwa ein viertel aller Projekte, bzw. leiden unter Budgetüberschreitungen, Qualitätsmängeln oder Verzögerungen [Sta99, BB+04]. Die Konsequenzen sind dramatisch, wie z.B. durch fehlerhafte Software verschuldete Waffenfehlfunktionen oder der immense monetäre Schaden (ca. 6 Milliarden €), der durch die Verzögerungen im Rahmen des Toll Collect Projekts entstanden ist. Diese Zustände sind auf Dauer nicht haltbar. Die Anwendung standardisierter Prozesse für die Softwareentwicklung, wie z.B. dem Rational Unified Process (RUP, [Kru00]) oder dem V-Modell XT [RB06, VMXT] versprechen hier Abhilfe. Standardisierte und bewährte Vorgehensweisen halten Best Practices ebenso vor, wie sie die Möglichkeit zur Ausbildung und Wissensaufbereitung bieten. Verschiedene empirische Studien zeigen die Vorteile standardisierter Prozesse, die einmal eingeführt, Qualität, Produktivität und Projekterfolg verbessern können [Wa02]. Hierfür sind Schlüsselfaktoren, wie z.B. Werkzeugunterstützung oder ausgebildete (auch zertifizierte) Mitarbeiter notwendig. Trotzdem sind

verschiedene Aspekte bei der Einführung und Anwendung eines standardisierten Prozesses zu beachten:

- Eine organisationsweit akzeptierte Prozessbeschreibung, inkl. normiert beschriebener Vorgehensweisen und eines einheitlichen Vokabulars.
- Angemessene Methoden für die Messung und das Controlling. Entsprechende Instrumentarien müssen die Beobachtung von Projekten unterstützen und Kennzahlen für Auswertungen liefern.
- Ein auf Erfahrungen basierender Katalog von Deeskalations- und Verfahrensanweisungen für Krisenszenarios.

Auch auf der Ebene der Messdatenerfassung wäre eine weitgehend standardisierte Vorgehensweise wünschenswert. Langjährige Erfahrungen zeigen allerdings, dass es keine allgemeingültigen Metriken zur Messung von Entwicklungsprozessen und -produkten im Software-Bereich gibt. Solche Metriken sind immer abhängig von dem Messzweck, dem Messobjekt, der Messperspektive, dem Qualitätsfokus und dem Kontext. Es gilt also zu unterscheiden zwischen Messmechanismen, die über Projekt- und Organisationsgrenzen hinweg bereitgestellt werden können und individuell vorzunehmenden Anpassungen. Auf der Ebene von Prozessstandards bietet sich an, bereits im Metamodell Mechanismen für die Instrumentierung mit Metriken bereitzustellen. Solche Mechanismen (z.B. Attributierungen von Prozessen oder direkte Modellierung entsprechender Aktivitäten, wie in [GHY97]) können einheitlich für das vorzunehmende Tailoring von Messprogrammen genutzt werden. Des Weiteren bietet sich an, auf Basiskataloge für Metriken zurückzugreifen, die weitgehend kontextunabhängig sind. Beispiele sind Metriken aus dem Bereich des Projektmanagements (z.B. Termintreue oder weitere, wie z.B. in [Bur02] definiert). Solche Basiskataloge können vielfach bereits auf der Ebene von Organisationen definiert werden, damit in konkreten Projekten bekannt ist, welche Daten zu ermitteln sind. Darüber hinaus lassen sich hier bereits auch die Messmethoden definieren. Dies liefert im Rahmen einer organisationsspezifischen Implementierung eines Vorgehensmodells ein Methodenset, welches weite Teile des Projektcontrollings erfasst.

Aktuelle Ansätze aus dem Bereich der *Softwareleitstände* oder der sog. *Project Cockpits* [Mue06] bieten bereits Möglichkeiten der integrierten Projektdurchführung mit Kontrolldatenerfassung. Hier sind im Wesentlichen zwei Ansätze zu finden: Erstens *Management-, Cockpits*. Diese dienen der „Management-kompatiblen“ Zusammenstellung und Präsentation von Daten und Statistiken für das gehobene Management. Hier sind in der Regel mehrere Projekte zu beachten, die aufbereitet und auch verglichen werden müssen. *Einheitliche, vergleichbare* Datensätze sind hierfür unbedingt erforderlich. Die zweite Variante sind die *projektbasierten Cockpits*, die sich vornehmlich an Projektleiter richten. Diese benötigen weitaus fein granularere Informationen über den Stand eines Projekts, um den aktuellen Ist-Status mit dem dokumentierten Vorgehen abzugleichen.

Mit diesem Beitrag wollen wir Optionen und Potenziale aufzeigen, die Metamodellbasierte Vorgehensmodelle wie das V-Modell XT [RB06, VMXT] bieten, um organisationsweite Qualitätsmaßstäbe auf der Ebene der Systementwicklungsmodelle zu integrie-

ren. Hierzu diskutieren wir verschiedene Realisierungsvarianten für integrierte Kontrollverfahren. Im Anschluss betrachten wir das V-Modell XT genauer und zeigen potenzielle Verbindungs- und Integrationspunkte für die Implementierung von Controlling-Mechanismen auf der Metamodellebene.

## 2 Stand der Technik und Varianten

Im Bereich der Messung und Steuerung von IT-Projekten sind zurzeit viele neuere Ansätze zu finden, die Ähnlichkeiten zur Flugsicherung und zu Fertigungsleitständen der Produktionstechnik aufweisen [Mue06]. Jedoch auch Ansätze, die eine direkte Prozessintegration suchen, sind zu finden [Riz06]. In diesem Abschnitt wollen wir vornehmlich die Möglichkeiten der Integration derartiger Verfahren in Vorgehensmodelle betrachten.

### 2.1 Softwareleitstände und -cockpits

Softwareleitstände und Software-Cockpits versuchen, Daten und Informationen zielgerichtet zu sammeln und als Berichte, bzw. Entscheidungsgrundlagen aufzubereiten und zu präsentieren [MH04, Mue06]. Sie unterscheiden sich von „klassischen“ Dashboards dadurch, dass Metriken individuell wählbar sind. Software-Cockpits können somit *organisationsweit* eingeführt werden, bieten den ausführenden Projektmitarbeitern aber dennoch die Möglichkeit weit gehender Anpassbarkeit.

### 2.2 Metrikdefinition, Controlling und Reporting

Hierzu muss jedoch die Frage beantwortet werden, **w**o Metriken definiert werden? Es gibt hier mindestens zwei prinzipielle Ansatzpunkte: Entweder in einem Werkzeug oder direkt im implementierten Vorgehensmodell. Die erste Variante entspricht dabei am ehesten der Vorstellung eines „klassischen“ Software-Cockpits nach [MH04]. Eine separate Anwendung unterstützt das Management auf der einen und die Projektleitung auf der anderen Seite. In ihr werden die auszuwerten Controllingdaten erfasst. Dieses Projektcockpit unterstützt die Anwender bei der Festlegung von Messzielen, Maßen und Messmethoden. Es ist z.B. denkbar, dass Metriken mit der *Goal-Question-Metric-Methode* (GQM, [BCR94]) abgeleitet und dann werkzeuggestützt implementiert werden. Auch nicht automatisch erfassbare Metriken wie z.B. *Function Points* (FP, [PB05]) können in Cockpits integriert werden. Messungen können beispielsweise im Rahmen eines automatisierten Reportings, wie z.B. im Microsoft Solution Frameworks (MSF) in Verbindung mit dem Team Foundation Server [VSTS] umgesetzt wird, erfolgen.

Ein wesentliches Problem wird dadurch jedoch nicht adressiert – die Integration der Messung in den Systementwicklungsprozess. Neuere Prozessansätze, wie das *Live Paradigma* [Riz06] versuchen hier, eine Brücke zu schlagen. Auch wenn es hier nicht das Primärziel ist, Projektcontrolling zu implementieren, sondern agilen Methoden zu noch mehr Salonfähigkeit zu verhelfen, ist die Fragestellung durchaus aus verwandt: *Was muss wann wie gemessen werden, um Entscheidungen treffen zu können?* [Riz06] kom-

biniert im Live Ansatz Elemente formaler Prozesse mit Agilen Methoden und definiert Anforderungen und Fähigkeiten in diesem Konzept, die auch für die Implementierung von allgemeinen Controllingmechanismen geeignet sind.

Mit dem V-Modell XT liegt ein Vorgehensmodell vor, das explizit auf die Erweiterung und Ausgestaltung angewiesen ist. Die Möglichkeiten einer direkten Integration mit dem V-Modell sind viel versprechend, da so ein Instrumentarium geschaffen werden kann, das neben Qualitäts- und Risikomanagementfähigkeiten gleichzeitig ein Benchmarking [VBen] gestattet. Einzelne Projekte werden somit aufgrund genormter Messdaten und Messpunkte vergleichbar.

### **3 Erweiterungspotenziale für das V-Modell XT**

Prinzipiell ist es also ein viel versprechender Ansatz, Metriken und Messmethoden bereits auf der Ebene eines Vorgehensmodells zu implementieren. Ausgehend von einem Standardmodell, wie dem V-Modell XT, lassen sich hier verschiedene Aufsetzpunkte finden. Bei der Implementierung eines organisationspezifischen Vorgehensmodells [VBen, KT06] auf der Basis des V-Modells stehen viele Möglichkeiten, u.a. auch die Anpassung des Metamodells selbst zur Verfügung. Verfolgt man im Rahmen einer Anpassung das Ziel, fortgeschrittene Kontrollmechanismen zu integrieren, führt kein Weg an einer entsprechenden Anpassung des Metamodells vorbei.

#### **3.1 Das V-Modell XT Metamodell**

Das Metamodell des V-Modell XT (Abbildung 1) stellt alle wesentlichen Bestandteile zur Verfügung, die für verschiedene Messungen in Betracht kommen. Es definiert sowohl Artefakte als auch Prozesselemente. Da es ein formales Modell ist, das die Elemente und deren Beziehungen zueinander definiert, können Metriken eindeutig Elementen des Vorgehensmodells zugeordnet werden. Da das V-Modell mit den Projektdurchführungsstrategien auch über dynamische Elemente verfügt, die darüber hinaus auch die Erstellungszeitpunkte aller weiteren Elemente (Produkte) regeln, können über die Kopplung auf der Metamodellebene nicht nur die Messobjekte, sondern auch die Zeitpunkte für Messungen festgelegt werden. Im Gegensatz zu rein informellen Prozessbeschreibungen wird die Vergleichbarkeit von Messungen damit erheblich unterstützt und gesteigert.

Die Berücksichtigung der Messinstrumentierung im Metamodell zeigt vielfältige Potenziale für Messung, Analyse und Steuerung auf. So kann bspw. anhand der Produktzustände einzelner für eine Projektetappe notwendiger Produkte der Projektfortschritt ermittelt werden (z.B. anhand von Soll/Ist-Vergleichen, Termin- und Qualitätskontrollen, abgerechneten Aufwänden, Restkalkulation, Trend etc.).

Jedes der gezeigten Metamodellelemente ist anpassbar. Der Zeitpunkt der Anpassung ist dabei entscheidend. Das V-Modell sieht mindestens drei Zeitpunkte für eine Anpassung [VMXT, VBen] vor:

- 1.) Anpassung des Prozesses auf Organisationsebene,
- 2.) Anpassung des Prozesses auf Projektebene - Tailoring und
- 3.) Anpassung nach dem Tailoring.

Eine Anpassung im Rahmen der organisationspezifischen Einführung hat dabei die größte Tragweite, da sich diese auf alle Projekte auswirkt.

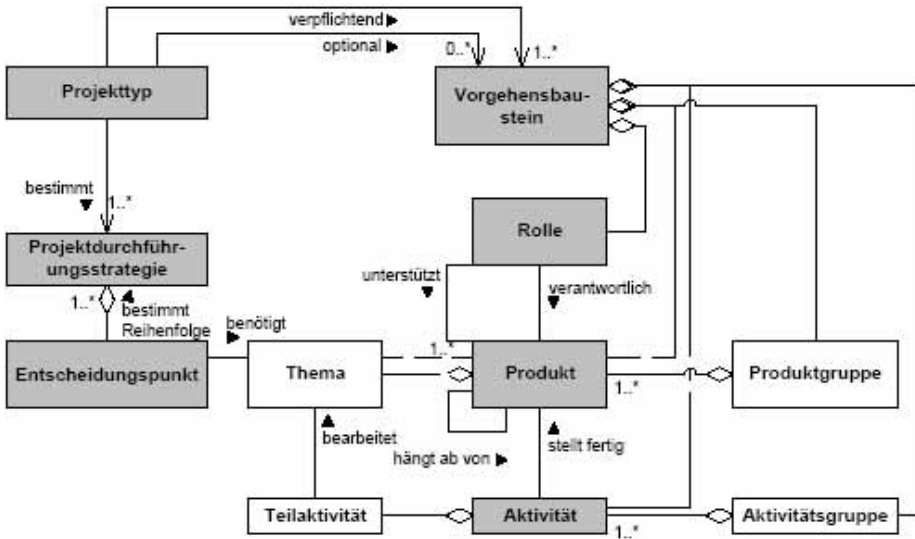


Abbildung 1 Metamodell des V-Modell XT (vereinfacht)

### 3.2 Zeitpunkt der Metrikdefinition

Diese Möglichkeiten gilt es zu nutzen und bereits auf der Metamodellebene die Erfassung objektiver Kennzahlen zu ermöglichen. Dabei muss auf der Ebene des Metamodells (also auf der Ebene des organisationspezifischen Vorgehensmodells) nur die grundlegende Fähigkeit der Datenerfassung und die Möglichkeit der Assoziation gegeben werden. Analog wie bei Software-Cockpits ermöglicht dieses Prinzip die nachlaufende projektspezifische Auswahl von Metriken und Messmethoden. Im Weiteren ist dann im Rahmen der organisationspezifischen Anpassung zu entscheiden, ob die Auswahl konkreter Methoden schon zum Zeitpunkt des Tailorings stattfinden muss (z.B. wenn schon wie gerade skizziert ein gewisses Methodenset vordefiniert wurde). Alternative Modelle können auch eine Auswahl während der Projektdurchführung ermöglichen.

### 3.3 Erweiterungspotenziale und Messpunkte

Gerade aufgrund des Metamodells des V-Modells lassen sich viele Koppelstellen und Messpunkte finden. Typischer Weise bieten sich wie weiter oben schon erläutert die Projektergebnisse, also die *Produkte* an, um Fortschritte, bzw. den Status zu messen.

Software-Cockpits, bzw. Live Ansätze, wie wir sie bisher betrachtet haben bieten hier bereits ausreichende Implementierungen [Riz06, VSTS]. Ein Potenzial, das jedoch auf der Ebene des V-Modell XT vorhanden ist, ist die Möglichkeit der Integration beliebiger Inhalte im Rahmen einer *Metamodellerweiterung*, bzw. -anpassung. So ist es z.B. denkbar, dass neben der Integration von Messpunkten oder Methoden gleichzeitig auch *Maßnahmenkataloge* definiert und integriert werden können. Solche Maßnahmen spiegeln die Erfahrungen einer Organisation im Projektgeschäft wider und können als *Risikominimierende Maßnahmen* im Rahmen des Risikomanagements sowohl im organisations- als auch im projektspezifischen Vorgehensmodell implementiert werden. Maßnahmenkataloge können im Rahmen einer Knowledge Base realisiert werden. Anhand der Daten, die aus abgeschlossenen Projekten abgeleitet werden können, welche Maßnahmen unter welchen Bedingungen welche Effekte hatten, wächst das Wissen und die Präzision der Daten. Stetig aktualisiert reifen derartige Maßnahmenkataloge mit der Zeit zu Assistenzsystemen, die z.B. in einem organisationsweiten Wiki oder Projektportal abgelegt werden können, eine Verfahrensweise, die z.B. im MSF [VSTS] gelebt wird. Standardsituationen können auf diese Weise hinterlegt und dokumentiert werden.

## 4 Zusammenfassung

Mit diesem Beitrag haben wir verschiedene Realisierungsvarianten für Softwareleitstände/Software-Cockpits betrachtet. Zurzeit entstehen einige Lösungen in diesem Arbeitsbereich (z.B. [Riz06] oder [VSTS]), die sich jedoch eher an der unterstützenden Anwendungslandschaft orientieren, d.h. die Funktionalität in separate Tools mit unterschiedlichem Integrationspotenzial auslagern.

Dieser Beitrag orientiert sich an einem Ansatz, der wesentlich früher greift, nämlich bereits zum Zeitpunkt der Einführung eines Systementwicklungsprozesses in einer Organisation. Hier bestehen die Möglichkeiten, Verfahren, Produkte und Maßstäbe zu definieren und im einzuführenden Prozess zu hinterlegen. Eine Definition muss an dieser Stelle allerdings noch nicht vollständig erfolgen, um den einzelnen Projekten noch Gestaltungsräume zu lassen. Jedoch schlagen wir vor, frühzeitig Koppelstellen zu identifizieren und entsprechende Messpunkte einheitlich und verbindlich vorzugeben.

Die hieraus entstehenden Potenziale sind vielfältig. Primär zu nennen sind die zu erwartenden positiven Auswirkungen auf die Qualität und das Risikomanagement. Die Qualität wird messbar und vergleichbar. Risikominimierende Maßnahmen sind somit auch sofort verfügbar, da eine objektive Entscheidungsbasis geschaffen wird. Bewährte Verfahren können hinterlegt und zur Anwendung empfohlen werden. Somit werden darüber hinaus auch die Projekte und deren Ergebnisse vergleichbar, was sich z.B. auch positiv auf die Reife eines Prozesses, z.B. hinsichtlich der Einstufung im CMMI äußern kann.

Der durchgängige Metamodell-basierte Ansatz, den wir empfehlen bedarf noch weiterer Erforschung, verspricht jedoch erheblichen Einfluss auf Kontrollen, Qualität und Risikominimierung in beliebigen Prozessen. Insbesondere für die wachsende Anzahl von Projekten, die es im Bereich des *Global Development* gibt, sehen wir hier ein enormes Potenzial, nicht nur um die Projekte zu vergleichen und zu überwachen, sondern weiter

gehend Entscheidungen darüber treffen zu können, ob sich derartige Projekte überhaupt „rechnen“.

## Literaturverzeichnis

- [Stan99] Standish Group International, Inc. *CHAOS: A Recipe for Success*. 1999.
- [BB+04] K. Bergner, M. Broy, K. Moll, M. Pizka, A. Rausch, T. Seifert. *Erfolgreiches Management von Softwareprojekten*, Informatik Spektrum 27:5, Springer-Verlag, 419-432, 2004.
- [BCR94] V. R. Basili, G. Caldiera, H. D. Rombaci. *Goal Question Metric Paradigm*. Encyclopedia of Software Engineering, Volume 1, John Wiley & Sons, pages 528-532, 1994.
- [Bur02] M. Burghardt. *Einführung in Projektmanagement*, Publicis Corporate Publishing, 2002.
- [GHY97] I. Graham, B. Henderson-Sellers, H. Younessi. *The Open Process Specification*, Addison-Wesley, 1997.
- [Kru00] P. Kruchten. *The Rational Unified Process, An Introduction*. Addison-Wesley, 2000.
- [KT06] M. Kuhrmann, T. Ternité. *Including the Microsoft Solution Framework as an agile method into the V-Modell XT*, Evaluation of Novel Approaches to Software Engineering (ENASE '06), 2006.
- [MH04] J. Münch, J. Heidrich. *Software project control centers: concepts and approaches*. The Journal of Systems and Software, Volume 70, pages 3-19, 2004.
- [Mue06] J. Münch. *Software-Cockpits bündeln Informationen*, Computerzeitung 37:21, Konradin IT-Verlag, 2006, S 18.
- [PB05] B. Poensgen, B. Bock. *Function-Point Analyse*. dPunkt, ISBN: 3-89864-332-8, 2005.
- [RB06] A. Rausch, M. Broy. *Das V-Modell XT*. dPunkt, 2006.
- [Riz06] S. Rizzo. *Agiles Projekt- und Anforderungsmanagement: Der Live-Ansatz*. Objekt Spektrum, 38-44, SIGS-DATACOM, 2006.
- [VBen] *V-Bench*. Webseite des Projekts im Web, <http://www.v-bench.de>.
- [VMXT] *V-Modell XT*. Definition und Dokumentation im Web, <http://www.v-modell-xt.de>.
- [VSTS] *Visual Studio Team System Developer Center*. <http://www.microsoft.com/germany/msdn/vstudio/teamsystem/expand/default.aspx>.
- [Wa02] H. Watts. *Three Process Perspectives: Organization, Teams, and People*. Annals of Software Engineering, Volume 14, pages 39-72. Kluwer Academic Publishers, 2002.