

Ein Modell zur Einflussanalyse von Änderungen in Entwicklungsprozessen

R. Koppe, S. Häusler, A. Hahn

OFFIS Institut für Informatik
{koppe, haeusler, hahn}@offis.de

Abstract: Wie ist der Einfluss einer neuen Methode auf die Produktivität eines Produktentwicklungsprozesses? Die Beantwortung dieser Frage ist eine der wichtigsten Erfolgsfaktoren für Unternehmen. Die quantitative Bewertung von neuen Methoden und Werkzeugen für existierende Entwicklungsprozesse unterstützt die Entscheidungsbasis von Managern für Prozessoptimierungen signifikant. Dieser Beitrag stellt ein Modell zur Einflussanalyse von Prozessänderungen vor, welches eine Bewertung neuer Methoden im Kontext von Entwicklungsprozessen gegen die Projektdimensionen von Zeit, Kosten und Produktqualität unterstützt.

1 Einleitung

Reifegradmodelle wie das Capability Maturity Model Integration (CMMI) [CMU06] definieren Ebenen der Reife einer Organisation. Ziel ist die Entwicklung von Prozessen zu solchen, die ständig optimiert werden. In diesem Zusammenhang versprechen neue Methoden und Werkzeuge große Potentiale für Prozess- und Produktverbesserungen. Doch ob sich eine neue Methode oder ein neues Werkzeug in einem Entwicklungsprozess wirklich lohnt, ist nicht leicht zu beantworten. Eine vom restlichen Entwicklungskontext isolierte Evaluierung neuer Methoden oder Werkzeuge ist oft nicht ausreichend. Als Alternative erlauben Pilotprojekte einen Einblick in das Prozessverhalten. Allerdings gestatten Pilotprojekte nur eine begrenzte Kontrolle, liefern keine statistisch signifikanten Ergebnisse [MPR05] und sind oftmals mit hohem (monetärerem) Aufwand verbunden. Zudem fehlt auch für die Bewertung von Pilotprojekten ein methodisches Gerüst.

Dieser Beitrag beschreibt ein Modell zur Einflussanalyse von Änderungen in Produktentwicklungsprozessen komplexer Systeme. Eingebunden ist das Modell in eine Methodik [Ko10], die es Prozessplanern und Entscheidern erlaubt, auf Basis identifizierter Wirkzusammenhänge (z.B. Expertenwissen), bereits vor Investitionen in neue Methoden (vgl. Investitionsrechnung z.B. [Go06]) Abschätzungen über deren Mehrwert durch Prozess-Simulation vorzunehmen. Methodik und Modell eignen sich dabei für beliebige Entwicklungsprozesse von Produkten (Software, Hardware, Computer Aided Design) und werden in aktuellen Fallstudien angewandt und evaluiert.

Im Folgenden gibt Abschnitt 2 begrenzt durch die Länge des Beitrags einen kurzen Überblick über verwandte Arbeiten. Abschnitt 3 beschreibt das Modell zur Einflussanalyse, welches in Abschnitt 4 exemplarisch angewendet wird. Schließlich liefert Abschnitt 5 einen Ausblick auf weitere Arbeiten.

2 Verwandte Arbeiten

Verwandte Arbeiten motivieren einen klaren Handlungsbedarf zur Analyse von Prozessänderungen durch neue Methoden, auch wenn ihr Fokus allgemeiner [Ba04] oder detaillierter [Kr08] als hier gesetzt wird. Es existieren zahlreiche Arbeiten zur Einflussanalyse und Einflussbewertung im Bereich des Software Engineering (SE) [Ke99], deren Vorgehen und Ansätze auch auf andere Bereiche der Produktentwicklungen anwendbar sind. Software Process Simulation (SPS) ist ein verbreiteter Ansatz, der durch Abdel-Hamid und Madnick [AM91] geprägt wurde. SPS wird unter anderem dazu verwendet Performanz-Analysen und „was-wäre-wenn“-Analysen durchzuführen und den Einfluss von Prozessänderungen vorherzusagen. Für einen Überblick verweisen wir auf [MP08] und [ZKP08]. Die grundlegenden Modelle werden in der Regel durch empirisch gewonnenen Daten kalibriert, um das Verhalten der realen Welt abzubilden [Ra99]. Dieses führt zum Empirical SE, mit dem Ziel Wirkzusammenhänge zwischen verschiedenen Prozesselementen zu beschreiben. Der Detaillierungsgrad modellierter Wirkzusammenhänge ist proportional abhängig von ihrem Einfluss auf kritische Faktoren wie Zeit, Kosten und Qualität. Andere Ansätze simulieren stochastische Prozesse [LK91] und damit die Folge von Aktivitäten, die durch Transitionen verbunden sind. Eine Transition besitzt dabei eine Übergangswahrscheinlichkeit, welche die Folge von Aktivitäten und damit auch den Prozessaufwand bestimmt. Existierende Arbeiten bieten kein standardkonformes generisches Modell zur Beschreibung von Prozessen und inhärenten Wirkzusammenhängen. Sie abstrahieren entweder sehr von diesen ([CE01], [DP07]), oder beschreiben Produkte und deren Charakteristika, welche den Prozessfortschritt wesentlich bestimmen, nicht adäquat ([Ma96], vgl. [Ru02]).

3 Beschreibung des Modells

Wir beschreiben zunächst die Grundidee zur Modellierung von Entwicklungsprozessen und Wirkzusammenhängen. Dann erläutern wir die Kernkonzepte des Modells und zeigen die Anwendung des Modells an einem Beispiel aus der Hardware Entwicklung.

Das Modell basiert auf der Idee, ein probabilistisches Modell zur Analyse des ökonomischen Mehrwerts von Prozessalternativen zu verwenden [DP07]. Ein solches Modell abstrahiert von der realen Welt und deren inhärentem Verhalten. Die Berechtigung hierfür liegt im immensen Aufwand, der notwendig wäre, um „vollständige“ Pilotprojekte oder empirische Studien durchzuführen.



Abbildung 1: Einfacher Prozess mit Wahrscheinlichkeiten an Transitionen

Abbildung 1 zeigt einen vereinfachten Prozess zur Entwicklung eines Chips [Ko10] (vgl. Abbildung 3). Aufgaben (Blöcke) sind durch Transitionen verbunden. Die Analyse-Aufgabe überprüft die Ergebnisse der vorherigen Schritte. Durch Expertenbefragung wird die Notwendigkeit eines Rücksprungs auf 20% festgelegt, um die Ergebnisse des

Register Transfer Level (RTL) Designs zu überarbeiten. Analysen erlauben nun die Abschätzung des Aufwands eines solchen Prozesses, indem jede Aufgabe einen bestimmten Aufwand besitzt. Diese Modellierung abstrahiert allerdings sehr weit von der Wirklichkeit, in der unterschiedliche Einflüsse (Komplexität des Produkts, Genauigkeit des Analyse-Werkzeugs, Erfahrung des Entwicklers, ...) etwa auf den Aufwand und den Erfolg einer Aufgabe wirken. Um diesem Problem Rechnung zu tragen, beschreiben wir relevante Wirkzusammenhänge explizit durch sogenannte Verhaltensmodelle in einem detaillierteren probabilistischen Modell.

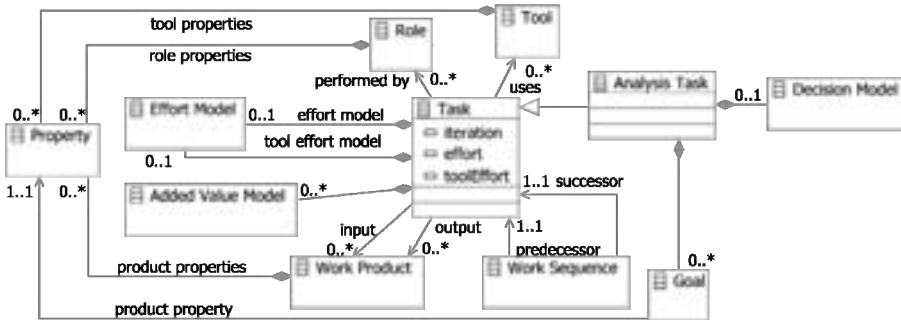


Abbildung 2: Basis des konzeptionellen Modells

Abbildung 2 zeigt einen Ausschnitt des konzeptionellen Modells, als Erweiterung des Software & Systems Process Engineering Meta-modell (SPEM 2.0), einem Standard der OMG [OMG08] zur methodischen Beschreibung von Entwicklungsprozessen. Methodische Beschreibungen können für konkrete Prozesse wiederverwendet werden und dienen somit dem Aufbau einer Art Wissensdatenbank, deren praktische Vorteile wir auch bei unserem Ansatz nutzen. Im Gegensatz zu anderen Modellen (z.B. [DP07]) werden explizit Produkte (Work Product) als Ein- und Ausgaben von Aufgaben (Task) modelliert. Eigenschaften (Property) charakterisieren Produkte sowie auch Rollen (Role) und Werkzeuge (Tool), die einen nicht zu vernachlässigenden Einfluss auf das Prozessverhalten besitzen. Analyse-Aufgaben (Analysis Task) verfügen über Ziele (Goal), welche die Validierung von Produkteigenschaften (z.B. Qualität) beschreiben. Dynamische Verhaltensmodelle beschreiben das Prozessverhalten auf Basis der genannten Konzepte und detaillieren einfache Übergangswahrscheinlichkeiten (etwa von Markov-Ketten) in parametrisierbare Wahrscheinlichkeitsverteilungen. Das Modell verbindet also Methoden- / Prozessmodellierung mit analysierbaren Prozessverhalten. Wir unterscheiden drei Arten von solchen dynamischen probabilistischen Verhaltensmodellen: Added Value Model, Effort Model und Decision Model, die wir im Folgenden erläutern.

Added Value Model beschreibt den Wertbeitrag einer Aufgabe zu Eigenschaften von Produkten (z.B. Qualität und Umfang) oder Rollen (z.B. Lerneffekte). Solche Veränderungen werden als lineare, nicht-lineare oder durch parametrisierte Wahrscheinlichkeitsverteilungen beschrieben. Formal ist *Added Value* eine Funktion $i \times R \times T \times P_{in} \rightarrow p_{out}$ mit i als Zähler der aktuellen Iteration (Fortschritt, Lernen) der Aufgabe, R als Teil der Eigenschaften beteiligter Rollen, T als Teil der Eigenschaften verwendeter Werkzeuge und P_{in} als Teil der eingehenden Produkteigenschaften. Das Ergebnis p_{out} ist z.B. eine berechnete Produkteigenschaft (funktionale Korrektheit) eines ausgehenden Artefakts und beeinflusst somit alle Verhaltensmodelle nachfolgender Entwicklungsschritte.

Effort Model dient zur Berechnung des Aufwands einer Aufgabe. Der Aufwand unterteilt sich in menschliche Arbeit und Werkzeuglaufzeit, um so Kosten für Mensch und Werkzeug genau zu unterscheiden. Ein *Effort Model* ist eine Funktion $i \times R \times T \times P_{in} \rightarrow \text{effort} \in \{\text{effort}, \text{toolEffort}\}$ und analog zu Added Value definiert. Das Ergebnis ist ein konkreter Wert in Stunden für den menschlichen Aufwand bzw. die Werkzeuglaufzeit.

Decision Model beschreibt die Entscheidung für eine folgende Aufgabe auf Grundlage bestimmter Ziele, die an Produkteigenschaften gestellt werden. Ein *Decision Model* ist eine Funktion $i \times R \times T \times P_{in} \times G \rightarrow ws$ mit den oben beschriebenen Parametern. G ist eine Menge von Zielen die für die Analyse-Aufgabe definiert wurden. Ein Entscheidungsmodell wird damit durch den einfachen Vergleich von Soll und Ist oder als komplexer (boolescher) Ausdruck definiert, der durch R , T und P_{in} beeinflusst wird. Das Ergebnis ws ist eine ausgehende Transition, welche auf die folgende Aufgabe verweist.

4 Anwendung des Modells zur Einflussanalyse

Wir zeigen nun die exemplarische Anwendung des Modells an einem Entwicklungsprozess aus der Hardware Entwicklung [Ko10]. Ziel der Prozessänderung ist die Reduktion des notwendigen Aufwands. Es sind zwei Prozessalternativen zu vergleichen:

- im bisherigen Entwicklungsprozess wird aus einem SystemC Modell durch einen Entwickler manuell ein Very High Speed Integrated Circuit Hardware Description Language (VHDL) Modell abgeleitet,
- der Prozess wird um ein neues Werkzeug zur automatischen Transformation eines SystemC Modells in ein entsprechendes VHDL Modell erweitert.

Es sei angemerkt, dass der grundlegende Ansatz auch auf andere Entwicklungsprozesse (System-, oder Softwareentwicklung) anwendbar ist. Lediglich die Wirkzusammenhänge sind andere. Außerdem ist der Ansatz nicht auf Bewertungen von Werkzeugänderungen oder auf Neuentwicklungen beschränkt. So erlaubt das Modell auch die Abbildung anderer Entwicklungsmethoden und -vorgehen oder auch Mitarbeiterkonstellationen.

Wir führen Einflussanalysen von Prozessänderungen durch, indem wir zunächst den Ist-Prozess und die Prozessänderung mit ihren Wirkzusammenhängen, die einen Einfluss auf den Prozessfortschritt besitzen (z.B. die aktuelle Qualität des Produkts) modellieren. Abbildung 3 zeigt dazu den Entwicklungsprozess mit Produkten, Eigenschaften, einem Ziel und einem Entscheidungsmodell. Die Prozessänderung ist umrahmt. Das Behavioral Design erzeugt ein *SystemC* Modell mit der Eigenschaft *functional correctness*. Diese Eigenschaft wird durch ein Added Value Model bestimmt, welches *Behavioral Design* zugeordnet ist. Das *SystemC* stellt eine Eingabe sowohl für die neue Aufgabe *Design Transformation* als auch für das *RTL Design* dar. Beide Aufgaben überführen das *SystemC* in ein *VHDL* Artefakt, welches auch eine Eigenschaft *functional correctness* besitzt. Das modellierte Verhalten der beiden Prozesse unterscheidet sich also in den Added Value Modellen der beiden Aufgaben, welche den Wertbeitrag der verschiedenen Aktivitäten zur funktionalen Korrektheit der Artefakte beschreiben. Erfüllt die *VHDL functional correctness* nicht das definierte Ziel (z.B. ist die verifizierte funktionale Korrektheit kleiner 100%), so ist ein Rückschritt zu *RTL Design* notwendig.

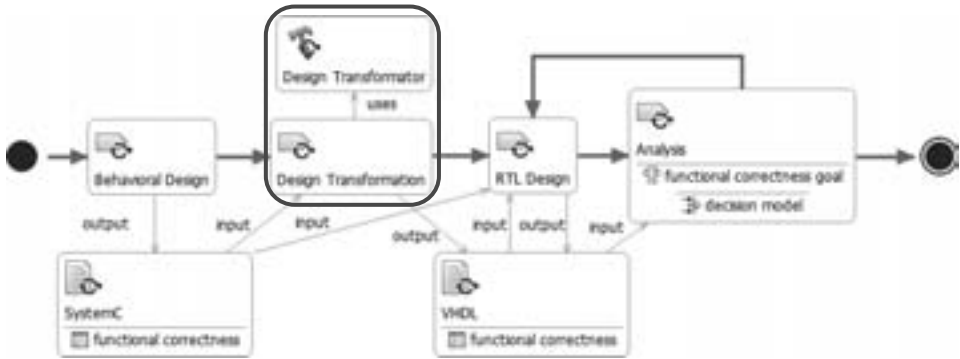


Abbildung 3: Erweiterung des Prozesses um Produkte, Ziel und Entscheidungsmodell

Die Added Value Modelle von *Design Transformation* und *RTL Design* werden als parametrisierte Wahrscheinlichkeitsverteilungen modelliert. Die Prozessänderung soll nun diese Verteilung der *functional correctness* deutlich nach rechts verschieben (z.B. sichert der Hersteller des Werkzeugs zu, aus einem korrekten SystemC immer korrektes VHDL zu generieren), um den notwendigen Aufwand durch Iterationen deutlich zu reduzieren, wie in Abbildung 4 gezeigt.

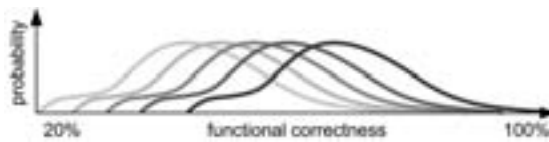


Abbildung 4: Wertbeitrag zur funktionalen Korrektheit des VHDL

Für die Analyse interessiert uns nun jeweils der notwendige Prozessaufwand der beiden Prozessalternativen. Monte Carlo Simulation wird für die Exploration der Verhaltensmodelle und Wahrscheinlichkeitsverteilungen verwendet. Die Ergebnisse werden dann statistisch aufbereitet (z.B. Minimum, Maximum, Mittelwert, Median und Quantile).

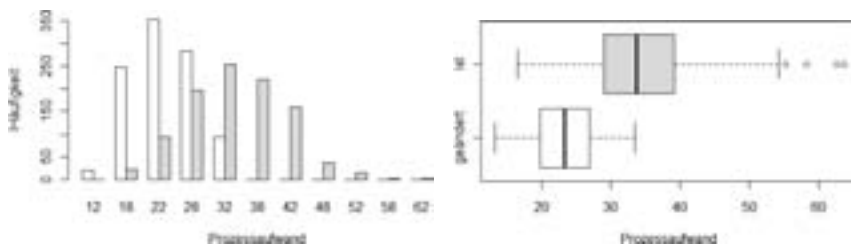


Abbildung 5: Ergebnisse des Prozessaufwands bei 1000-facher Simulation des Ist-Prozesses (dunkel) gegen den geänderten Prozesses (hell), links Verteilung des Aufwands als Histogramm, rechts aggregierte Lagemaße des Prozessaufwands als Box Plot

Abbildung 5 zeigt die Ergebnisse der beiden Alternativen grafisch als Histogramm und Box Plot. Es ist leicht zu erkennen, dass der geänderte Prozess (hell) vor dem aktuellen Prozess (dunkel) liegt. Der Wertbeitrag der neuen Methode wird als Differenz der jeweiligen Prozessaufwände etwa durch Mittelwert- oder Quantil-Vergleich beschrieben und eignet sich dann für strategische Break-even Point bzw. Pay-off Analysen.

5 Zusammenfassung und Ausblick

Dieser Beitrag präsentiert ein Modell zur Einflussanalyse von Methodenänderungen in Entwicklungsprozessen. Das Modell erweitert den OMG Standard SPEM 2.0 [OMG08] zur Beschreibung von Prozessen um Konzepte zur Modellierung von Wirkzusammenhängen zwischen Produkt- und Prozesseigenschaften. Eingebunden ist das Modell in eine Methodik [Ko10], welche die Modellierung von Ist- und Soll-Prozessen beschreibt und mittels Simulation einen Vergleich der Entwicklungsdauer und -kosten der Prozesse ermöglicht. Wir entwickeln derzeit Werkzeuge zur Modellierung und Analyse von Entwicklungsprozessen auf Basis des hier vorgestellten Modells. Weitere Werkzeuge unterstützen die Sammlung und Analyse von operativen Daten zur Modellierung von Wirkzusammenhängen. Bei unserer weiteren Arbeit werden wir unseren Ansatz und Werkzeuge evaluieren und deren Ergebnisse zur weiteren Verbesserung dieser nutzen.

Literaturverzeichnis

- [AM91] Abdel-Hamid, T.K.; Madnick, S.E.: Software project dynamics: an integrated approach. Englewood Cliffs, Prentice Hall, 1991.
- [Ba04] Balazova, M.: Methode zur Leistungsbewertung und Leistungssteigerung der Mechatronikentwicklung. Dissertation, Universität Paderborn, 2004.
- [CE01] Cho, S.; Eppinger, S.: Product Development Process Modeling Using Advanced Simulation. In: Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Pittsburgh, 2001. ASME 2001.
- [CMU06] Carnegie Mellon University: CMMI for Development, Version 1.2. 2006.
- [DP07] Deissenboeck, F; Pizka, M.: The Economic Impact of Software Process Variations. In: Software Process Dynamics and Agility. Springer, Berlin, 2007; S. 259-271.
- [Go06] Götze, U.: Investitionsrechnung: Modelle und Analysen zur Beurteilung von Investitionsvorhaben. Springer, Berlin, 2006.
- [HE08] Hassine, A.; Barke, E.: On Modeling and Simulating Chip Design Processes: The RS Model. In: Int. Engineering Management Conference. IEEE, Estoril, 2008; S. 81-85.
- [Ke99] Kellner, M. et.al.: Software Process Simulation Modeling: Why? What? How?. In: Journal of Systems and Software, Band 46, Nr 2-3. Elsevier, Amsterdam. S. 91-105
- [Ko10] Koppe, R. et.al.: Process Model Based Methodology for Impact Analysis of New Design Methods. In: Proceedings of the 1st International Conference on Modelling and Management of Engineering Processes, Cambridge 2010. Springer, London, 2010.
- [Kr08] Kröll, M.: Methode zur Technologiebewertung für eine ergebnisorientierte Produktentwicklung. Dissertation, Universität Stuttgart, 2008.
- [Ma96] Madachy, R.: System dynamics modeling of an inspection-based process. In: 18th Int. Conference on Software Engineering. IEEE, Berlin, 1996. S. 376-386.
- [MP08] Müller, M; Pfahl, D.: Simulation Methods. In: Guide to Advanced Empirical Software Engineering. Springer, London, 2008.
- [MPR05] Münch, J.; Pfahl, D.; Rus, I.: Virtual Software Engineering Laboratories in Support of Trade-off Analyses. In: Software Quality Journal, Band 13, Nr 4. Springer, Netherlands, 2005; S. 407-428
- [OMG08] OMG: Software & Systems Process Engineering Meta-model Specification, 2008.
- [Ra99] Raffo, D. et.al.: Empirical Studies Applied to Software Process Models. In: Empirical Software Engineering, Band 4, Nr 4. Springer, Netherlands, 1999; S.353-369.
- [Ru02] Rus et.al.: Systematically combining process simulation and empirical data in support of decision analysis in software development. In: 14th Int. Conference on Software Engineering and Knowledge Engineering. ACM, Ischia, 2002. S. 827-833.
- [ZKP08] Zhang, H.; Kitchenham, B.; Pfahl, D.: Reflections on 10 Years of Software Process Simulation Modeling: A Systematic Review. In: Making Globally Distributed Development a Success Story. Springer, Berlin, 2008; S. 345-356.