

# Konsistente Evolution von lebenszyklusübergreifenden Variabilitätsmodellen

Klaus Schmid

Institute of Computer Science  
University of Hildesheim  
Marienburger Platz 2  
D-31141 Hildesheim, Germany  
schmid@sse.uni-hildesheim.de

**Abstract:** Variabilitätsmanagement gilt als konzeptionelle Grundlage einer systematischen Entwicklung von Produktlinien. Dabei wird oft die Notwendigkeit eines lebenszyklusübergreifenden Ansatzes betont. Unter dem Aspekt der Evolution ergeben sich jedoch besondere Herausforderungen dabei. In diesem Beitrag geben wir eine kurze Übersicht einiger wesentlicher Herausforderungen.

## 1 Motivation

Produktlinienentwicklung hat sich in den letzten Jahren als ein effektiver Ansatz zur Entwicklung von Mengen ähnlicher Systeme in der Industrie etabliert. Dabei profitieren von einer Produktlinienentwicklung sowohl die Kostenseite als auch die Qualität der entwickelten Systeme. Die Details der Durchführung einer Produktlinienentwicklung unterscheiden sich dabei von Unternehmen zu Unternehmen sehr stark [4].

Als eine der konzeptionellen Grundlagen einer Produktlinienentwicklung wird immer wieder das Variabilitätsmanagement genannt. Betrachtet man aber die unterschiedlichen Ansätze [3, 1, 7, 5], die für ein Variabilitätsmanagement vorgeschlagen werden, so unterscheiden sich diese deutlich. Die Unterschiede liegen dabei nicht nur in den zugrundeliegenden Paradigma, sondern auch in der Lebenszyklusüberdeckung. So sind viele Ansätze aus den frühen Phasen (Modellierung / Verständnis der Produktlinie) motiviert [3] oder sind sogar darauf beschränkt [2].

## 2 Konsistentes Variabilitätsmanagement

Der zentrale Unterschied zwischen einer Einzelsystementwicklung und einer Produktlinienentwicklung liegt im Übergang von der Perspektive eines einzelnen Systems auf die Betrachtung einer Menge von Systemen. Die Variabilitätsmodellierung dient dazu die Unterschiede zwischen den Systemen herauszustellen. Der vielleicht bekannteste Ansatz dazu ist die Featuremodellierung [3]. Eine weitere Gruppe von Ansätzen ist die Entscheidungsmodellierung [8, 7]. Das Variabilitätsmodell beschreibt die Variation von Systemartefakten (z. B. Anforderungen, Architekturelemente, Implementierung, usw.). Dies geschieht bei der Featuremodellierung durch Bezugnahme auf Eigenschaften eines Systems, während dies bei der Entscheidungsmodellierung durch die Beschreibung der notwendigen Entscheidungen beschrieben wird.

Idealerweise wird Variabilitätsmodellierung lebenszyklusübergreifend angewendet. Das heißt, ein Variabilitätsmodell liegt vor, das es erlaubt alle Artefakte innerhalb des Lebenszyklus integriert zu konfigurieren. Abbildung 1 veranschaulicht diese Situation. Dadurch wird ein Zusammenhang zwischen den verschiedenen Stufen einer Entwicklung (Anforderungen, Architektur, Implementierung, Test) hergestellt. Dies hat einerseits positive Auswirkungen, wie beispielsweise die Unterstützung der Verfolgbarkeit von Variabilitäten über die verschiedenen Lebenszyklusstufen hinweg. Andererseits führt es durch die Verzahnung der verschiedenen Stufen auch zusätzliche Komplexität ein.

Ein zentrales Variabilitätsmodell kann jedoch auch entscheidende Herausforderungen in Bezug auf eine konsistente Modellierung von Variabilität und die Evolution von Variabilitätsinformationen stellen. Dies wollen wir in zwei Schritten diskutieren:

- grundsätzliche Fragen der Konsistenz von Variabilitätsmodellen;
- Fragen der Evolution von Variabilitätsmodellen.

### **3 Konsistenz von Variabilitätsmodellen**

Selbst ohne Betrachtung von Evolutionsaspekten ist es nicht einfach festzustellen, was es für ein Variabilitätsmodell bedeutet konsistent zu sein. Das Variabilitätsmodell beschreibt die Konfigurationsmöglichkeiten der Produktlinie. Doch genau genommen kann man dieses nicht in Isolation betrachten, sondern man muss es in Kombination mit den verschiedenen Artefakten des Lebenszyklus betrachten, wie dies in Abbildung 1 ausgedrückt wird. Dies steht im Gegensatz zu den meisten aktuellen Arbeiten, die ausschließlich das Variabilitätsmodell (z. B. Feature-Baum) betrachten.

Mit diesem erweiterten Konsistenzbegriff ergibt sich aber die Frage: welche Konfigurationen sind jeweils gerade gemeint? Handelt es sich um die Möglichkeiten, die im Moment als Anforderungen vorhanden sind, oder um diejenigen, die in der Architektur dargestellt werden, oder die die tatsächlich implementiert werden? Hinzu kommt, dass bestimmte Variabilitäten nur für bestimmte Abschnitte des Lebenszyklus relevant sein könnten (beispielsweise bestimmte Realisierungsversionen). Was also ist in diesem erweiterten Sinne ein „konsistentes“ Variabilitätsmodell?

Aus formaler Sicht ist dies ein Modell, das die Variationen, die durch die Produktlinie unterstützt werden, vollständig und widerspruchsfrei repräsentiert. Aus einer pragmatischen Sicht heraus, hat diese Definition aber einige Probleme:

- *Anwendbarkeit im Lebenszyklus*: Eine einzelne Variabilität muss nicht notwendigerweise in allen Phasen der Entwicklung sichtbar sein. So kann eine Variabilität beispielsweise erst in der Architektur relevant werden, beispielsweise weil es nicht sicher ist welche von zwei Möglichkeiten vorzuziehen ist. Idealerweise würde dieses Problem so gelöst, dass jede Architekturvariabilität auch einer Anforderungsvariation entspricht. Dies ist jedoch nicht immer sinnvoll.

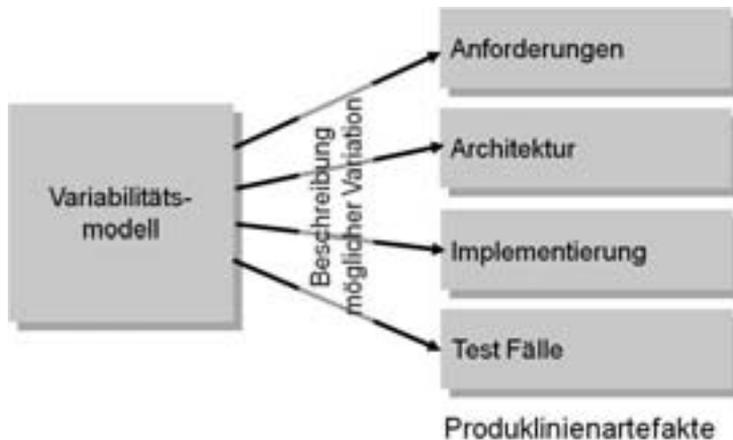


Abbildung 1. Darstellung der Beziehungen zwischen Variabilitätsmodell und Lebenszyklusartefakten.

- *Umsetzungsgrad*: Die Realisierung einer Variabilität kann sehr unterschiedlich auf den verschiedenen Ebenen des Lebenszyklus sein. Dies kann selbst mehrere Gründe haben:
  - *Fortschritt der Umsetzung*: Gewisse Variationen wurden bereits auf der Anforderungsebene beschrieben. Die Umsetzung ist jedoch (auf Grund des Projektablaufs) noch nicht erfolgt.
  - *Planungsunterschiede*: Der Umfang der Funktionalität, der auf einer Ebene geplant wird, entspricht nicht dem, der auf anderen Ebenen umgesetzt wird. Beispielsweise geschieht es, dass Anforderungen erfasst werden, jedoch wird dann entschieden, dass verschiedene Variationen (vorerst) nicht weiter umgesetzt werden.

Will man dies auch unterstützen, so erfordert eine systematische, lebenszyklusübergreifende Unterstützung des Variabilitätsmanagements, dass es beispielsweise möglich ist zu annotieren, dass einige Variabilitäten nur lokale Gültigkeit haben. Entsprechend sind zwei Einschränkungen des zuvor skizzierten, übergreifenden Variabilitätsmanagements notwendig:

1. Bestimmte Variationen sind evtl. nur für einen Ausschnitt der Artefakte relevant. Diese Zuordnung sollte nicht nur indirekt durch die Artefaktabhängigkeiten, sondern auch explizit beschreibbar sein.
2. Der Umsetzungsgrad für Variationen ist über den Lebenszyklus hinweg nicht homogen. Dies erfordert eine Erfassung der Umsetzung einer Variabilität über den Lebenszyklus hinweg.

Beide Eigenschaften zusammen machen es erforderlich, die Gültigkeit eines Variabilitätsmodells in Bezug auf die Zuordnung zu bestimmten Lebenszyklusphasen (z. B. repräsentiert durch Artefakte) zu definieren.

## 4 Evolution von Variabilitätsmodellen

Bereits in [6] haben wir das Thema der Evolution von Variabilitätsmodellen betrachtet. Dort leiteten wir (aus Sicht der Anforderungen) eine Taxonomie für Produktlinien-evolutionsschritte ab. Dabei unterschieden wir in die drei Ebenen: produktbezogene Änderungen, Änderungen im Hinblick auf Varianten, sowie Änderungen von Gemeinsamkeiten. Auf jeder dieser Ebenen kann es für sich betrachtet zu Evolutionsschritten kommen. Darüber hinaus sind auch Übergänge zwischen diesen Ebenen möglich. Eine Übersicht gibt Abbildung 2.

Zusätzlich kann man Evolutionsschritte noch bezüglich ihres Umfangs kategorisieren. So gibt es Änderungen die ganze Produktlinien verändern (z. B. die Aufspaltung von einer Produktlinie in zwei), solche die die Produkte einer Produktlinie verändern (z. B. Hinzufügen eines Produkts) und schließlich solche, die zur Änderung einzelner Anforderungen führen.

Die Details der Taxonomie würden den Rahmen dieser Ausführungen sprengen, jedoch konnten 20 verschiedene Evolutionsoperationen identifiziert werden. Diese Kombinationen ergaben sich insbesondere durch die getrennte Betrachtung des Hinzufügens, Wegnehmens und Ändern von Elementen. Dabei wurde die Analyse jedoch ausschließlich aus der Anforderungsperspektive durchgeführt. Die oben angeführten Überlegungen zu einer den Lebenszyklus umfassenden Perspektive wurden dabei noch nicht berücksichtigt.

Die Notwendigkeit der Unterstützung der verschiedenen Variabilitätsoperationen führt neben dem Bedarf zur direkten Unterstützung der verschiedenen Operationen auch zur Notwendigkeit die entsprechenden Informationen zu erfassen und zu verwalten. Auf Basis der Analyse der Variabilitätsoperationen konnten wir die Notwendigkeit zur Verwaltung (und Manipulation) verschiedenster Abhängigkeiten identifizieren [6].

## 5 Zusammenfassung

In den beiden vorhergehenden Abschnitten, diskutierten wir einige Herausforderungen zur Produktlinienervolution. Dies waren einerseits Anforderungen, die sich aus der unterschiedlichen Abdeckung der verschiedenen Lebenszyklusabschnitte ergaben, zum anderen solche, die sich aus der Evolution der Produktlinie als Ganzes ergeben.

Die Kombination beider Formen der Evolution, wie sie täglich in jeder realen Produktlinie geschieht, erfordert wenn sie erfolgreich sein soll, eine umfangreiche Evolutionsunterstützung. Dabei muss sich beispielsweise auch der Projektfortschritt im Sinne einer unterschiedlichen Umsetzung von Produktlinienvariabilität im Variabilitätsmodell widerspiegeln.

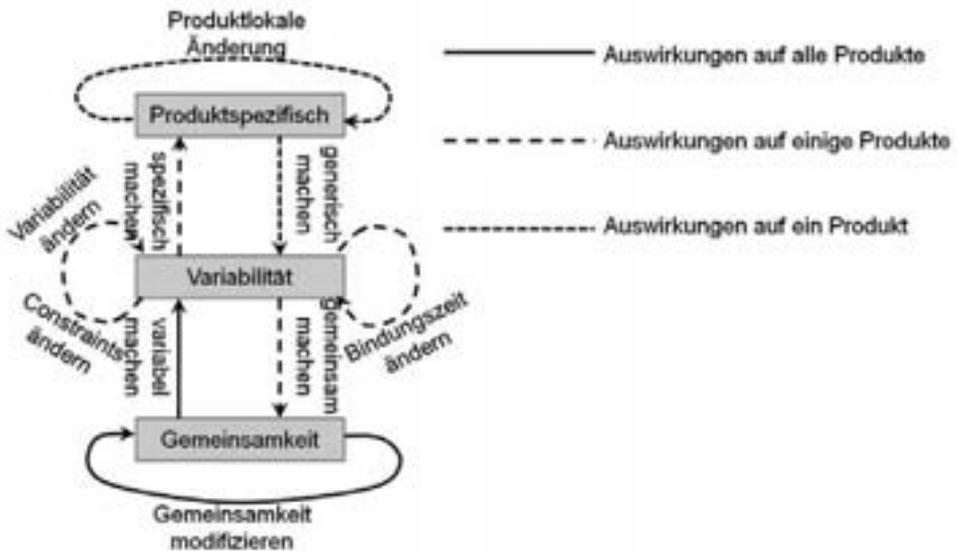


Abbildung 2. Übergänge in der Evolution einer Produktlinie

Entsprechend unserer Diskussion scheinen beide Evolutionsdimensionen weitestgehend orthogonal zu sein. Während dies den Vorteil hat, dass beide getrennt betrachtet werden können, bleibt jedoch die Herausforderung beide Aspekte explizit zu betrachten. Uns ist aktuell noch keine adäquate Unterstützung dieser Evolutionsherausforderungen bekannt.

## References

- [1] K. Czarnecki and U. W. Eisenecker. *Generative Programming: Methods, Techniques, and Applications*. Addison-Wesley, 1999.
- [2] M. Eriksson, J. Börstler, and K. Borg. The PLUSS approach — domain modeling with features, use cases and use case realizations. In *Proceedings of the 9th International Conference on Software Product Lines*, 2005.
- [3] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21 ESD-90-TR-222, Software Engineering Institute Carnegie Mellon University, 1990.
- [4] F. van der Linden, K. Schmid, and E. Rommes. *Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering*. Springer, 2007.
- [5] K. Pohl, G. Böckle, and F. van der Linden. *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer, August 2005.
- [6] K. Schmid and H. Eichelberger. A requirements-based taxonomy of software product line evolution. *Electronic Communications of the EASST*, 8, 2008.
- [7] K. Schmid and I. John. A Customizable Approach to Full-Life Cycle Variability Management. *Science of Computer Programming*, 53(3):259–284, 2004.
- [8] Software Productivity Consortium Services Corporation, Technical Report SPC-92019-CMC. *Reuse-Driven Software Processes Guidebook, Version 02.00.03*, November 1993.

