

Planung und Realisierung von Software-Produktlinien

Dr. Peter Knauber

Fraunhofer Institut für Experimentelles Software Engineering (IESE)

Sauerwiesen 6, D-67561 Kaiserslautern

Tel.: 06301-707 242

Email: Peter.Knauber@iese.fhg.de

Kurzbiographie: Dr. Peter Knauber ist Abteilungsleiter für Software-Produktlinien am Fraunhofer IESE. Von 1992 bis 1995 hielt er unter anderem Vorlesungen zu Themen der Software-Entwicklung an der Berufsakademie und der Fachhochschule Mannheim sowie der Fachhochschule Trier; seit 1999 hält er Vorlesungen über Software-Produktlinien und Reengineering an der Universität Kaiserslautern.

Seit 1997 arbeitet er am Fraunhofer IESE. Dort ist er verantwortlich für öffentlich geförderte und Industrie-Projekte, in denen Methoden für Software-Produktlinien und Software-Architekturen entwickelt und in Unternehmen transferiert werden.

Abstract

Software Produktlinien werden allgemein als vielversprechender Ansatz gesehen, um bei der Entwicklung ähnlicher Software-Produkte von Wiederverwendung zu profitieren [10]. In der Theorie geht der Entwicklung des ersten Produkts eine ausführliche Planungs- und Modellierungsphase voraus. In der Praxis findet jedoch meist die Entwicklung des größten Teils der notwendigen Infrastruktur zusammen mit der Entwicklung der ersten Produkte statt. Die Herausforderung bei der Software-Entwicklung von und mit Produktlinien besteht darin, trotz geringen Planungsvorlaufs eine tragfähige Plattform für die späteren Mitglieder der Produktlinie zu erarbeiten.

Einleitung

Software-Produktlinien (auch: Produktfamilien, Systemfamilien) versprechen Wiederverwendung in großem Stil, indem gemeinsame oder ähnliche Funktionalität explizit für die Wiederverwendung in vielen (oder zumindest mehreren) Systemen entwickelt wird, wodurch sich Kosten und Aufwand auf diese Systeme verteilen [6]. Das Konzept ist aus der Produktion in traditionellen Industriebereichen, wie z.B. der Automobilindustrie oder der Unterhaltungselektronik, abgeleitet, wo Produktlinienentwicklung seit langem sehr erfolgreich eingesetzt wird: Es wird eine generische, also anpassbare Plattform (auch Produktlinien-Infrastruktur genannt) entwickelt, auf welche die späteren Mitglieder der Produktlinie aufbauen. Durch die Wiederverwendung der Funktionalität, die von der Plattform zur Verfügung gestellt wird, können – je nach Zielsetzung der Produktlinie – Vorteile wie geringere Entwicklungs- und Wartungskosten, kürzere time-to-market und höhere Qualität der Produkte erreicht werden.

Die gesamte in der Plattform realisierte Funktionalität, die in unterschiedlichem Ausmaß von allen Produkten genutzt wird, muss nur einmal entwickelt werden und kann dann zentral gewartet und weiterentwickelt werden. Die Vorteile aus dem Produktlinienansatz ergeben sich durch eben diese Synergieeffekte, wodurch Pro-

Produktlinien gegenüber anderen Wiederverwendungsverfahren deutlich besser abschneiden [10]. Mittlerweile sind bereits relativ viele Erfolge durch den Einsatz von Produktlinienverfahren dokumentiert [1, 4, 5].

Der Unterschied zwischen Theorie und Praxis

In der Theorie findet die Software-Entwicklung von und mit Produktlinien in einem zweiphasigen Entwicklungszyklus statt, in dem zunächst *für* und dann *mit* Wiederverwendung produziert wird (vgl. Abbildung 1). Die Entwicklung für Wiederverwendung wird als *Domain Engineering*, die Entwicklung mit Wiederverwendung als *Application Engineering* bezeichnet. Dieser zweiphasige Lebenszyklus liegt z.B. auch dem ESAPS-Projekt¹ zugrunde [11]. Bei ESAPS handelt es sich um das aktuell größte Forschungsprojekt zum Thema Software-Produktlinien in Europa mit über 20 Projektpartnern aus Industrie und Forschung.

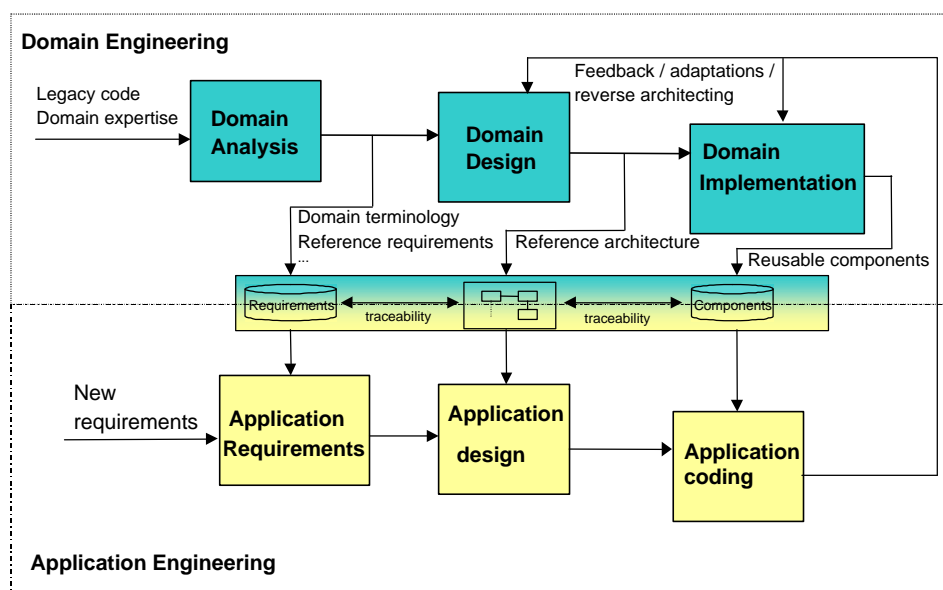


Abbildung 1: Das zweiphasige Prozessmodell im ESAPS-Projekt

In der Praxis zeigt sich jedoch, dass es oft nicht sinnvoll (oder sogar nicht möglich) ist, die beiden genannten Phasen voneinander zu trennen. Im Regelfall muss – schon aus Zeitgründen – der größte Teil der Plattform zusammen mit der Entwicklung der (ersten) Produkte realisiert werden.

Allerdings entsteht bei der Produktentwicklung praktisch nie eine Plattform “nebenbei”, außer es handelt sich um ein Produkt, das die gesamte Funktionalität der späteren Produktlinie in sich vereint, also eine Art Obermenge der Produktlinienfunktionalität umfasst. Um eine wirklich tragfähige Plattform für eine Produktlinie zu entwickeln, ist es nötig, zunächst zu bestimmen, welche Funktionalität tatsächlich in identischer oder ähnlicher Form in zukünftigen Produkten benötigt wird und mit diesem Wissen die Plattform zu entwerfen. Nur so können später teure Änderungen und Erweiterungen, wie sie auch bei der Einzelsystementwicklung anfallen, für zukünftige Produktvarianten vermieden werden.

1. ESAPS (Engineering Software Architectures, Processes and Platforms for System Families) ist das Projekt mit der Nummer 99005 im Eureka Σ!2023-Programm innerhalb der ITEA-Initiative.

Eine Lösung

Die Herausforderung besteht also darin, die Plattform so zu planen und unter Benutzung geeigneter Implementierungsmechanismen zu entwickeln, dass Plattformteile bereits in ersten Produkten eingesetzt werden können, ohne dass dazu die gesamte Plattform realisiert sein muss. Zu diesem Zweck wurde am Fraunhofer IESE PuLSETM (für **P**roduct **L**ine **S**oftware **E**ngineering) als Entwicklungsmethode für Produktlinien entworfen [3]. Im Sinne eines inkrementellen Plattformaufbaus können die beiden Phasen des Produktlinienzyklus nach einer initialen Planungsphase verwoben ablaufen. Bei der Planung wie auch der Entwicklung selbst können bereits existierende Architektur-, Design- und Code-Teile aus eventuellen Legacy-Systemen berücksichtigt werden [2].

PuLSE enthält eine Komponente, die explizit für die Planung von Produktlinien, das sogenannte Scoping, zuständig ist [8]. So wird es möglich, systematisch diejenigen Teildomänen zu identifizieren, für die sich die Investition in Wiederverwendung am wahrscheinlichsten auszahlt. Kriterien dafür sind unter anderem die Reife (und damit Stabilität) der Teildomäne, die Varianz verschiedener Produkte innerhalb der Teildomäne sowie die Kopplung mit anderen Teildomänen. Basierend auf den Ergebnissen des Scoping lassen sich auch erste Empfehlungen aussprechen, welche Implementierungsmechanismen für ihre Umsetzung in zukünftigen Produkten geeignet sind [9]. Die Möglichkeiten dabei reichen von der Definition gemeinsamer Schnittstellen bis hin zur Entwicklung von Generatoren.

Das Scoping von Produktlinien sollte unbedingt auch dann erfolgen, wenn aus Zeit- oder Kostengründen ein inkrementeller Produktlinienansatz gewählt werden muss: nur so lässt sich sicherstellen, dass bei der Entwicklung der ersten Produkte nicht implizite Entscheidungen getroffen werden, welche die spätere Nutzung der so entstandenen Assets im Rahmen der Plattform erschweren oder sogar unmöglich machen. Eben wegen dieser Bedeutung der Produktlinienplanung wird Scoping auch explizit als eigene Disziplin im Software Product Line Practice Framework [7] des Software Engineering Instituts (SEI) aufgeführt.

Zusammenfassung

Software-Produktlinien werden mittlerweile als der erfolgreichste Mechanismus für strategische Wiederverwendung angesehen da sie gleichzeitig domänenspezifisch und produktorientiert sind. Um mit Produktlinien erfolgreich zu sein, sind jedoch zwei Dinge Voraussetzung: die Möglichkeit, Anforderungen an zukünftige Produkte zumindest ungefähr abschätzen zu können, um eine halbwegs verlässliche Planung zu ermöglichen, sowie die Bereitschaft, von Standard-, also zweiphasigen, Produktlinienverfahren abzuweichen *ohne* dabei die Zielrichtung der Produktlinie gegenüber dem jeweils aktuellen Produkt aus den Augen zu verlieren und Einzelsystementwicklung zu betreiben.

Nur dann zahlt sich die Investition in Produktlinien aus.

Literatur

- [1] T. Bardo et al., *CORE: A Product Line Success Story*, CrossTalk: J. Defense Software Engineering, Vol. 9, No. 3, 1996
- [2] Bayer et al., *Transitioning Legacy Assets to a Product Line Architecture*, Seventh European Software Engineering Conference/Seventh ACM SigSoft

- Symposium Foundations of Software Engineering, Springer Verlag, Berlin, 1999, pp. 446-463
- [3] Bayer et al., *PuLSE: A Methodology to Develop Software Product Lines*, Symposium on Software Reusability '99, ACM Press, New York, 1999, pp. 122-131
 - [4] L. Brownsword und P. Clemens, *A Case Study in Successful Product Line Development*, Technical Report CMU/SEI-96-TR-016, Carnegie Mellon University, Pittsburgh, Pa., 1996
URL: <http://www.sei.cmu.edu/publications/documents/96.reports/96.tr.016.html>
 - [5] C. Gacek et al., *Successful Software Product Line Development in a Small Organization. A Case Study*, IESE Public Technical Report No. 013.01/E, 2001
URL: http://www.iese.fhg.de/pdf_files/iese-013_01.pdf
 - [6] P. Knauber et al., *Applying Product Line Concepts in Small and Medium-Sized Companies*, IEEE Software , 2000, pp. 88-95
 - [7] L. Northrop, *A Framework for Software Product Line Practice – Version 3.0*, URL: <http://www.sei.cmu.edu/plp/framework.html>
 - [8] K. Schmid, Multi-Stage Scoping for Software Product Lines, in Proc. of Software Product Lines: Economics, Architectures, and Implications, P. Knauber, G. Succi (Eds.), Workshop at International Conference on Software Engineering, Limerick, Ireland, 2000,
URL: http://www.iese.fhg.de/pdf_files/iese-070_00.pdf
 - [9] K. Schmid und C. Gacek, *Implementation Issues in Product Line Scoping*, Software Reuse: Advances in Software Reusability- Proceedings of the 6th International Conference, ICSR'6, Vienna, Austria, 2000, Lecture Notes in Computer Science, 1844 (Ed.) William B. Frakes, Springer, 2000, pp. 170 - 189
 - [10] D. Rine und R. Sonnemann, *Investment in reusable Software. A study of software reuse investment success factors*, The Journal of Systems and Software, Vol. 41, 1998, pp. 17-32
 - [11] F. van der Linden, *ESAPS – Engineering Software Architectures, Processes and Platform for System Families*, in Proc. of Software Product Lines: Economics, Architectures, and Implications, P. Knauber, G. Succi (Eds.), Workshop at International Conference on Software Engineering, Limerick, Ireland, 2000,
URL: http://www.iese.fhg.de/pdf_files/iese-070_00.pdf