

# Requirements-Engineering und -Management in Produktmanagement und Produktlinien-Entwicklung

Andreas Birk, Software.Process.Management  
Gutenbergstraße 99, 70197 Stuttgart, E-Mail: andreas.birk@swpm.de

## Abstract

Die Rolle des Produktmanagements (PM) in der industriellen Software-Entwicklung ist im Requirements-Engineering und -Management (REM) lange Zeit nur wenig beachtet worden. Mit dem Trend zu Software-Produktlinien (SPL) anstelle einzeln entwickelter Produkte nimmt die Bedeutung des PM zu. Dieser Artikel umreißt die besonderen Herausforderungen, die PM und SPL an den Umgang mit Requirements stellen.

## Die Annahmen des Requirements-Engineering und -Management

Die Methoden des REM sind weithin durch Annahmen geprägt, die von der Entwicklung von Einzelprodukten ausgehen: Projektumfang und Adressatenkreis („Stakeholder“) stehen weitgehend fest, bevor das RE beginnt. Die Gesamtmenge der relevanten Requirements ist im Wesentlichen klar umgrenzbar; Definition und Umsetzung der Requirements liegen zeitlich (relativ) nah beieinander; Requirements werden nach der Umsetzung kaum mehr benötigt, allenfalls bei späterer Wartung.—All diese Annahmen treffen bei der industriellen Produktentwicklung und insbesondere bei der Produktlinien-Entwicklung nur selten zu. Es ist also angebracht, die Herausforderungen an das REM in diesem Umfeld näher zu betrachten.

## REM im Produktmanagement

Das Produktmanagement befasst sich mit der Planung und dem Marketing von Produkten über alle Phasen des Produkt-Lebenszyklus hinweg. Einen guten Überblick über die Aufgaben bietet das Pragmatic Marketing Framework [9]. Es definiert die drei Rollen Director Product Strategy, Technical Product Manager und Product Marketing Manager und ihre Aufgaben. Einen engen REM-Bezug weisen nur die beiden ersten Rollen auf. Der Director Product Strategy schafft mit Produktportfolio, Marktanforderungen und Produkt-Roadmap vor allem Vorgaben und Rahmenbedingungen für das REM. Die technischen REM-Aufgaben nimmt der Technical Product Manager wahr. Sie umfassen: Innovation, User Personas, Use Scenarios und Release Milestones.

Eine detaillierte Binnensicht auf das Produktmanagement von Software-Produkten bietet das Reference Framework for Product Management von van der Weert, Binkkemper et al. [10]. Es gliedert Produktmanagement in Portfolio Management, Product Roadmapping, Requirements Management und Release Planning. RM ist hier also ein Teilbereich des PM. Es steht in engen Wechselbeziehungen zu Portfolio- und Release-Management, die im herkömmlichen Verständnis von REM kaum beachtet werden.

Abbildung 1 gibt einen Überblick über die Rollen und den Informationsfluss, die dem REM im Produktmanagement zugrunde liegen. Die Produkt-Anforderungen werden von verschiedenen Rollen schrittweise aus den Kunden- und Marktanforderungen abgeleitet.

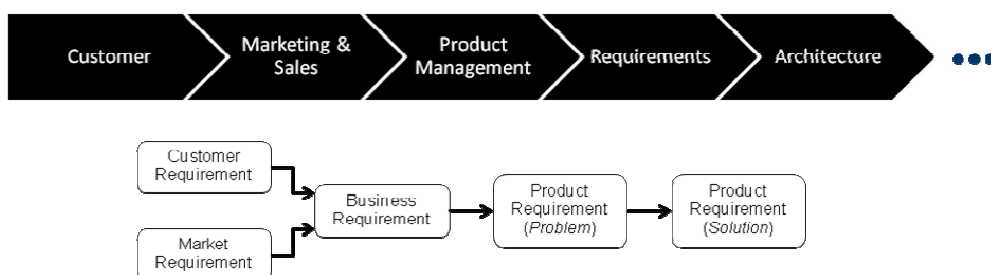


Abbildung 1: Beteiligte Rollen und Informationsfluss beim RE im Produktmanagement-Umfeld.

In der Praxis fällt auf, dass die Rolle des Requirements Engineers im Umfeld der Produktentwicklung fast nie etabliert ist. Vielmehr teilen Produktmanager und Architekt die REM-Aufgaben unter sich auf. Im Detail geschieht dies auf recht unterschiedliche Weise. Werner beschreibt [11] die schrittweise Transformation von

Anforderungen entlang definierter Zustände und Quality Goals, während Fricker [5] die iterative Abstimmung zwischen Requirements-Definition (PM) und Implementation Proposal (Architektur) schildert.

### **Herausforderungen an das REM bei Software-Produktlinien**

Die Entwicklung von Software-Produktlinien [4][7][8] ist wesentlich komplexer als die Entwicklung von Einzelprodukten. So ist das Requirements-Engineering und -Management für SPL sehr anspruchsvoll [1][2][3]: Gleichzeitig müssen mehrerer Produktvarianten betrachtet werden, langfristig bindende Architekturalternativen müssen abgewogen werden, Variabilität muss angemessen beschrieben werden, und Entscheidungen müssen auch für spätere Zeiten dokumentiert sein. Zugleich aber sind reife REM-Praktiken auch ein wichtiges Erfolgskriterium für die Entwicklung von Software-Produktlinien.

Good Practices für das REM in der SPL-Entwicklung umfassen unter anderem [1][3]: Klare Kommunikations- und Abstimmungsstrukturen (z.B. Change Control Board, Architecture Review Board), Programm-Management-Office als Ergänzung des Projektmanagements, explizite und zugleich pragmatische Definition von Variabilität sowie ein zentrales, werkzeuggestütztes Requirements- und Dokumenten-Management.

Die SPL-Prozesse müssen in besonderem Maße auf die Anwendungsdomäne und das Umfeld der Software-Entwicklung angepasst sein. Zum Beispiel wirkt sich die Anwendungsdomäne mit ihrer spezifischen Marktstruktur wesentlich auf die Kommunikationsbeziehungen und die Organisation des REM aus (z.B. Endverbrauchermarkt vs. wenige Einzelkunden). Ähnlich bedeutsam ist die Frage, ob das Produkt ein reines Software-Produkt ist oder ob es sich um ein Hardware/Software-System handelt.

### **Zusammenfassung**

Die Praxis des Produktmanagements und der SPL-Entwicklung stellen Herausforderungen an den Umgang mit Requirements, die von den aktuellen REM-Methoden noch nicht zufriedenstellend erfüllt werden. Dazu zählen Unterstützung und Dokumentation der vielfältigen Abstimmungsprozesse im PM sowie die Definition komplexer Variabilitätsbeziehungen bei SPL und deren Integration mit herkömmlichen Verfahren der Requirements-Definition. Existierende REM-Methoden müssen weiter entwickelt werden, um hier geeignete Lösungen zu bieten. Neben fokussierten Forschungsvorhaben ist dazu auch gerade der enge Dialog zwischen Forschung und Praxis erforderlich, wie er beispielsweise in Netzwerken der GI-FG Requirements-Engineering statt findet (vgl. [6]).

### **Literatur**

- [1] Birk, A.: Requirements-Engineering und -Management in Produktmanagement und Produktlinien-Entwicklung. Präs. beim Jahrestreffen der GI-FG RE, 2007. (<http://www.swpm.de/publications.html>)
- [2] Birk, A., Heller, G., John, I., von der Maßen, T., Müller, K., Schmid, K.: Product line engineering: The state of the practice. IEEE Software, 20 (6), 2003.
- [3] Birk, A.; Heller, G.: Challenges for Requirements Engineering and Management in Software Product Line Development. Proc. REFSQ 2007, Springer-Verlag, Berlin, 2007.
- [4] Clements, P., Northrop, L. M.: Software Product Lines: Practices and Patterns. Addison Wesley, 2002.
- [5] Fricker, S., Gorschek, T., and Myllyperkiö, P.: Handshaking between Software Projects and Stakeholders Using Implementation Proposals. Proc. REFSQ 2007, Springer-Verlag, Berlin, 2007.
- [6] GI-AK Requirements-Engineering-Frameworks und Produktlinien (REFPL). <http://refpl.gi-ev.de>
- [7] van der Linden, F.; Schmid, K.; Rommes, E.: Software Product Lines in Action. Springer, Berlin, 2007.
- [8] Pohl, K., Böckle, G., van der Linden, F.: Software product line engineering. Springer, Berlin, 2005.
- [9] Pragmatic Marketing: Pragmatic Marketing Framework. <http://www.pragmaticmarketing.com/pragmatic-marketing-framework>
- [10] Weerd, I. van de, Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L.: On the creation of a reference framework for software product management: Validation and tool support. Proc. 1st Int. WS Product Management, Minneapolis, MN, 2006.
- [11] Werner, M.P.: The Value of the Quality Gateway. Journal of Universal Knowledge Management (J.UKM), 1(2), 2006.