

Unterstützung evolutionärer Softwareentwicklung durch Merkmalmodelle und Traceability-Links

Matthias Riebisch
Technische Universität Ilmenau,
matthias.riebisch@tu-ilmenau.de

Änderungen und Nachvollziehbarkeit. Anforderungen an Softwaresysteme unterliegen vielfältigen Änderungen in schneller Folge. Kann eine Software nicht mehr entsprechend verändert werden, verliert sie ihre Nutzbarkeit. In der Praxis führen Änderungen häufig zu strukturellen Mängeln, was zu abnehmender Änderbarkeit und zu steigender Entropie der Software führt. Als Konsequenz dieser Effekts des sog. Architectural Decay steigt der Aufwand für Änderungen, bis ein Zustand erreicht wird, in dem Änderungen nicht mehr stabilisiert werden können. Wirtschaftliche Schäden sind die Folge, weil auch eine Neuentwicklung der Software wegen Kosten und Risiken keinen Ausweg darstellt.

Zur Vermeidung dieses Effekts muss die Wartung von Software in Bezug auf Veränderungen und Verständlichkeit unterstützt werden. Zur Verständlichkeit tragen z.B. Dokumentation, Modelle und aussagekräftige Bezeichner bei. Entwurfsmuster fördern zusätzlich die Änderbarkeit. Softwaretechnische Prinzipien wie Kapselung, Information Hiding und Abstraktionen verringern Abhängigkeiten. Die Angabe von Entwurfsentscheidungen erleichtert Änderungen. Alle diese Maßnahmen haben jedoch bisher zu keiner ausreichenden Verbesserung der Situation geführt. Eine bessere Nachvollziehbarkeit der Auswirkungen von geänderten Anforderungen kann – neben den genannten Maßnahmen – einen wesentlichen Beitrag zur Verständlichkeit und zur Prüfbarkeit leisten.

Traceability-Links. Traceability-Links wurden zur Verknüpfung von Anforderungen unterschiedlicher Ebenen entwickelt [RJ01]. Sie unterstützen Änderungen z.B. bei Konsistenzprüfungen. Aufgrund des bei Anforderungen sehr unterschiedlichen Grads von Abstraktion, Formalisierung, Unschärfe und Vollständigkeit erfordern Erstellung und Pflege der Traceability-Links hohen manuellen Aufwand. Das Konzept der Traceability-Links eignet sich über das ursprünglichen Ziel hinaus gut zur Verknüpfung der Anforderungen mit Entwurfsmodellen und Implementierung. Ein Traceability-Link des Typs <<implementedBy>> kann beispielsweise eine Anforderung aus dem Pflichtenheft mit der Entwurfsbeschreibung einer Komponente sowie mit deren Quellcode verbinden. Die Vorteile dieses Konzepts liegen in der expliziten Modellierung von Bezügen und Abhängigkeiten. Die Implementierung der Traceability-Links ist auf der Basis existierender Methoden, Technologien und Werkzeuge möglich [SR02].

Bei komplexen Softwaresystemen ist wegen der großen Anzahl solcher Links die Werkzeugunterstützung eine Voraussetzung für den praktischen Einsatz. Für Traceability-Links zwischen Entwurfs- und Implementierungselementen sind Erstellung und Pflege weitgehend automatisierbar [PR04]. Entwurfswerkzeuge können Traceability-Links aufzeichnen, Refactoring-Werkzeuge Veränderungen durchführen. Links zwischen Anforderungen und Lösungselementen müssen dagegen weitgehend manuell erstellt und geprüft werden, weil Entscheidungen über inhaltliche Zusammenhänge wegen der durch Begriffe beschriebenen Anforderungen das Hintergrundwissen der Entwickler erfordern. Dieser hohe manuelle Aufwand gefährdet das Ziel des Konzepts. Werkzeugunterstüt-

zung erfordert es, Anforderungen stärker zu strukturieren und zu formalisieren, wozu eine Vielzahl von aktuellen Ansätzen des Requirements Engineering beitragen können. Außerdem ist die große Differenz in Abstraktionsgrad und Unschärfe zwischen Anforderungen und Lösung zu überbrücken, wozu hier Merkmalmodelle eingesetzt werden.

Merkmalmodelle als Bindeglied zwischen Anforderungen und Lösung. Für die Anforderungsdefinition von Software-Produktlinien haben sich Merkmalmodelle (engl.: Feature Models) etabliert [Ka90]. Diese wurden als zentrale Notation für Merkmale, Anforderungen und Entwurfsentscheidungen in Produktlinien-Entwicklungsmethoden weiterentwickelt. Jedes Merkmal kennzeichnet eine Produkteigenschaft aus Sicht des Kunden, wird durch einen Begriff beschrieben und fasst eine Menge von Anforderungen zusammen. Merkmalmodelle setzen Merkmalen in Beziehung zueinander. Es wurden verschiedene Merkmal- und Beziehungstypen definiert [RSP03]. Die Beziehungen werden mittels Ausdrücken der Object Constraint Language OCL beschrieben [Ri03].

Da Merkmalmodelle einerseits Anforderungen strukturieren, und andererseits zur Strukturierung von Lösungen genutzt werden, werden sie hier als Bindeglied zwischen Anforderungen und Lösung eingesetzt, um durch bessere Strukturierung der Traceability Links mehr Werkzeugunterstützung zu ermöglichen. Traceability-Links verbinden Anforderungen (z.B. in Use-Case-, Klassen- und Interface-Modellen) mit Merkmalen sowie diese mit Modell- und Implementierungs-Elementen. Das Merkmalmodell stellt hier vom Abstraktionsgrad her eine Zwischenebene bereit. Dies erweist sich auch beim Reverse Engineering als vorteilhaft [PR04]. Außerdem unterstützen Merkmalmodelle die Erstellung geeigneter Architekturen, indem sie für Modularisierung und Separation of Concerns ausgewertet werden.

Werkzeugunterstützung für diesen Ansatz wurde durch Verbinden von Merkmalmodell-Werkzeugen mit den Repositories üblicher CASE-Werkzeuge mittels XML erreicht. Durch Auswertung der OCL-Ausdrücke der Merkmalbeziehungen lassen sich Konsistenzprüfungen bei der Weiterentwicklung von Produktlinien und Produkten durchführen. Der Nutzen des Ansatzes konnte bei der Weiterentwicklung eines umfangreichen Systems zur industriellen Bildverarbeitung gezeigt werden. Weitere Informationen sind [PR04] und [AI04] zu entnehmen.

- [AI04] Alexandria - Produktlinien-Entwicklungsmethodik. TU Ilmenau, 2004. Online verfügbar unter <http://www.theoinf.tu-ilmenau.de/pld>
- [Ka90] Kang, K., Cohen, S., Hess, J., Novak, W., Peterson, A., Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-021, SEI, 1990.
- [PR04] Pashov, I., Riebisch, M.: Using Feature Modeling for Program Comprehension and Software Architecture Recovery. In: Proc. 11th IEEE Conf. on Engineering of Computer-Based Systems (ECBS'04). IEEE, 2004 (in Druck)
- [Ri03] Riebisch, M.: Evolution und Komposition von Softwaresystemen. Habilitation. TU Ilmenau, 2003 (eingereicht).
- [RJ01] Ramesh, B.; Jarke, M.: Toward reference models for requirements traceability. IEEE Transactions on Software Engineering, Vol. 27, No. 1, pp. 58-93, 2001
- [RSP03] Riebisch, M.; Streitferdt, D.; Pashov, I.: Modeling Variability in Object-Oriented Product Lines. In: Buchmann, A.; Buschmann, F. [Eds.]: ECOOP Workshops 2003. LNCS, Springer, 2003. (in Druck)
- [SR02] Sametinger, J.; Riebisch, M.: Evolution Support by Homogeneously Documenting Patterns, Aspects and Traces. 6th Europ. Conf. on Software Maintenance and Reengineering. Budapest, 2002 (CSMR 2002) . Computer Society Press, 2002. S. 134-140.