Identifikation von Anforderungen aus Benutzerdokumentation

Isabel John

Fraunhofer-Institut für Experimentelles Software Engineering, Abteilung „Information Systems Development“,
67663 Kaiserslautern, Isabel.John@iese.fraunhofer.de

1 Einleitung

Typischerweise werden Systeme nicht auf der grünen Wiese entwickelt, im Allgemeinen entwickelt man existierende Systeme weiter oder migriert Systeme auf Basis von mehreren Altsystemen.

Ein Entwicklungsansatz der systematischen Wiederverwendung genutzt werden kann ist die Entwicklung in Produktlinien[1][2]. Hier wird die Entwicklung in zwei Lebensyzklen unterteilt (Entwicklung für Wiederverwendung und Entwicklung mit Wiederverwendung) und Produktzentriert wiederverwendbare Komponenten entwickelt. Die Einführung von Produktlinien ist jedoch eine Investition die geplant werden muss. Bei diesen frühen Planungs- und Anforderungsaktivitäten (Produktlinien Scoping) ist die Einbeziehung des Wissens von Domänenexperten essentiell.

In existierenden Scoping Ansätzen wird die Information die benötigt wird interaktiv von den Domänenexperten erfragt[3]. Dies ist ein aufwands- und kommunikationsintensiver Task durch die große Anzahl an Domänenexperten die potentiell involviert werden müssen und der Zeit die für Workshops und Interviews benötigt wird[4] und scheitert oder verzögert sich oft wegen der der mangelnden Verfügbarkeit und starken Einbindung der Domänenexperten in Entwicklungs- und Planungsaktivitäten.

Der hier präsentierte CaVE Ansatz (Commonality and Variability Extraction) [5][6] bietet Lösung für dieses Verfügbarkeitsproblem. Der Ansatz unterstützt Scoping und Anforderungsanalyse mit einer patternbasierten Methode die Benutzerdokumentation existierender Systeme systematisch analysiert. Es können Features (für den Benutzer relevante Eigenschaften des Systems), Domänen und Subdomänen (Konzeptionelle Einheiten innerhalb der Systembeschreibung) Use Cases und Use Case Elemente, Relationen und Gemeinsamkeiten und Variabilitäten aus Benutzerdokumentation identifiziert werden. Diese Methode kann von nicht spezialisierten Kräften (z.B. Studenten) nach kurzem Einarbeitungsaufwand angewandt werden und unterstützt somit die Einführung von Produktlinien durch die Entlastung der Domänenexperten. Durch die flexibel einsetzbaren Pattern ist der Ansatz sowohl für Produktlinien als auch für Einzelsysteme anwendbar.

2 Elemente des Ansatz

Der Ansatz besteht aus folgenden Elementen:

1. Analyse-Pattern die beschreiben, wie man typischerweise Anforderungsinformation in Benutzerdokumentation identifiziert (z.B. „Use case Beschreibungen kann man in nummerierten Listen finden“). Die Pattern (siehe Tabelle 1) werden je nach Art der Dokumentation und gewünschten Artefakten ausgewählt und bei Analyse und Vergleich der Dokumentationen manuell angewendet. Es existieren Pattern für verschiedene Eingabeartefakte (Phrasen, Überschriften, Listen etc) und verschiedene Ausgabeartefakte (Features, Use Cases, Variabilitäten etc). Die jeweils passenden Artefakttypen und damit die passenden Pattern können systematisch ausgewählt werden.
2. Eine Analyse Methode (siehe Abb. 1) die die Schritte bei der manuellen Analyse beschreibt. In der Vorbereitungsphase (Preparation) werden Patterns ausgewählt und die Dokumentation in handhabbare und vergleichbare Teile unterteilt. In der Analysephase werden die Dokumentteile mit Hilfe der Pattern miteinander verglichen. Die extrahierten Phrasen und Sätze werden markiert oder in vorläufigen Produktlinien Artefakten wie variablen Use Cases oder einer Produkt-Feature-Matrix gesammelt. Im dritten Schritt werden die vorläufigen Artefakte mit dem Domänen-Experten validiert und können für die Produktlinienentwicklung genutzt werden.

Output	Pattern
Features	Headings of sections or subsections typically contain features
Features	Features can be found in highlighted phrases (bold or italic font) or in extra paragraphs
Use Cases	Phrases like “normally” “with the exception”, “except” can mark Use Case extensions
Use Cases	Numbered lists or bulleted lists are markers for an ordered processing of sequential steps and describe Use Case descriptions
Variability	Arbitrary elements occurring only in one user manual are optional elements
Variability	Phrases or sentences that differ in only one or a few words can be evidence for alternatives

Tabelle 1: Beispiel Pattern

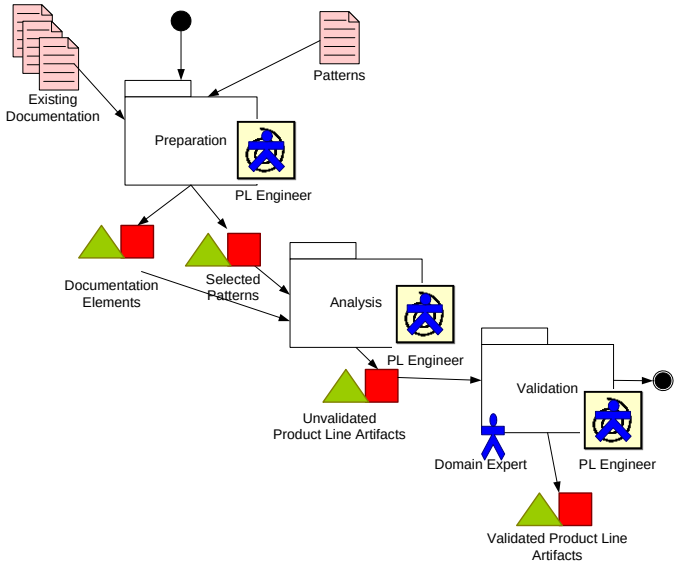


Abbildung 1: Übersicht des CaVE Ansatz

3. Ein Metamodell das das Mapping von Benutzerdokumentation zu Scoping- und Anforderungsdokumenten beschreibt und zusätzlich den Variabilitätsaspekt von Produktlinien integriert.

3 Validierung

Der Ansatz wurde empirisch in mehreren industriellen Fallstudien und einem Experiment validiert. Die Fallstudien kamen zu folgenden Ergebnissen:

Die Korrektheit der Ergebnisse lag zwischen 79 und 96%, d.H. es wurde mehr als ¾ der Elemente, die durch Anwendung von CaVE gefunden wurden von den Experten als korrekt bewertet

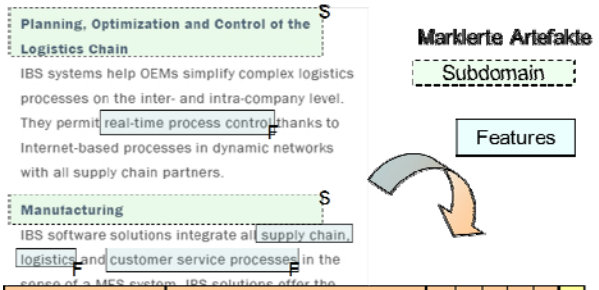
Die Vollständigkeit der Ergebnisse lag zwischen 70 und 89%, d.H. es wurden mehr als 70% der von den Experten gefundenen Elemente auch mit CaVE gefunden

Die Expertenlast wurde in jeder Fallstudie um mehrere Stunden (zwischen 3 und 12 Stunden) reduziert.

Das Experiment zeigte die Anwendbarkeit der Methode und gute Ergebnisse bei einigen Elementtypen wie Features und Variabilities. Bei Use Case Elementen und anderen Anforderungen zeigte das Experiment, dass der Ansatz noch schwächen hat, hier sollten neue Pattern entwickelt werden oder der Ansatz methodisch verbessert.

4 Ausblick

Der Ansatz wird am Fraunhofer IESE in Industrieprojekten verwendet und ist inzwischen auch außerhalb des IESE im Einsatz (siehe z.B. [7]). Momentan handelt es sich um einen manuellen Ansatz der auf kleineren Dokumentmengen anwendbar ist.



Subdomain	Nr	Feature	Automotive	LIMS	PRISMA	Professional	Webmodule	future Pr.
Logistics Chain	76	real-time process control	X	X	X	X		
	77	Order Management	X	X	X	X		
	78	Order Generation	X	X	X	X		
Manufacturing	79	supply chain logistics		X		X	X	
	80	customer service processes		X	X			
	81	integral quality mangement	X	X	X	X		
	82	manufacturing planning	X	X		X		
	83	manufacturing control	X	X	X	X		
	84	selective data creation		X		X		

Abbildung 2 Beispielartefakte

Eine Anwendung in einem Tool ist durch eine Formalisierung der Pattern leicht möglich und wurde in Teilbereichen auch schon angegangen. Zusätzlich wären weitere Fallstudien sinnvoll, um die Anwendbarkeit der Methode in verschiedenen Kontexten zu zeigen. Weitere Beispiele und Einzelheiten finden sich in [6].

4 Referenzen

[1] K. Pohl, G. Böckle, F. van der Linden. Software Product Line Engineering. Foundations, Principles, and Techniques, Springer, 2005

[2] Clements, P.C., Northrop L.: Software Product Lines: Practices and Patterns. Addison-Wesley 2001

[3] Marcela Balbino Santos de Moraes, Eduardo Santana de Almeida, Silvio Romero de Lemos Meira Systematic Review on Software Product Lines Scoping, Proceedings of ESELAW'09, 2009

[4] Isabel John, Michael Eisenbarth: A decade of scoping: a survey. SPLC 2009: 31-40

[5] Isabel John: Using Documentation for Product Line Scoping. IEEE Software 27(3): 42-47 (2010)

[6] Isabel John Pattern-based Documentation for Software Product Lines, Dissertation, Fraunhofer IRB, 2010

[7] Danilo Beuche Variantenmanagement und Wiederverwendung von Anforderungen. Tutorial/Workshop Reconf 2010