

Unterstützung der Koexistenz von agilen und traditionellen Anforderungsartefakten

Olga Boruszewski¹

Abstract: Die hier vorgestellte Dissertation [Bo16] befasst sich mit dem Zusammenspiel verschiedener Repräsentationen von Anforderungen in Softwareprojekten. Anforderungen werden auf Anforderungsartefakten, wie Spezifikationen, Use Cases oder Geschäftsprozessmodellen, festgehalten und dann für viele Aktivitäten innerhalb der Softwareentwicklung eingesetzt. Häufig werden mehrere Arten von Anforderungsartefakten zusammen eingesetzt, um verschiedene Blickwinkel abzudecken. Dabei treten Probleme auf, weil die Anforderungen aufgrund von inhaltlichen Überlappungen Abhängigkeiten besitzen. Die Dissertation untersucht den Umgang mit Anforderungsartefakten in Projekten und die dabei entstehenden Probleme und erarbeitet dann Lösungskonzepte. Basierend auf einer Praxisstudie stellt die Arbeit Methoden bereit, mit denen eine Anforderungslandschaft modelliert und optimiert werden kann und mit denen Verknüpfungen automatisiert genutzt werden. Die Arbeit ermöglicht damit, verschiedene Arten von Anforderungen effektiv zu kombinieren und damit auch hybride - also teils agile, teils traditionelle - Softwareentwicklungsansätze zu unterstützen.

Keywords: Software Engineering, Requirements Engineering, Anforderungsartefakte, Artefaktmodelle, Tracing, Anforderungskommunikation, hybride Entwicklungsansätze.

1. Einleitung

Der Erfolg von Softwareprojekten hängt in großem Umfang davon ab, wie gut die Anforderungen an die Software verstanden, ausgearbeitet und umgesetzt werden. Fehler in einer dieser Aktivitäten können dazu führen, dass fertiggestellte Software die Probleme der Kunden nicht löst oder nicht deren Wünschen entspricht. Auch qualitativ hochwertige Software kann so dennoch zu Unzufriedenheit bei den Kunden führen. Um Anforderungen effektiv zu erarbeiten, spielen Artefakte eine wichtige Rolle. Diese helfen, durch Externalisierung von Wissen ein gemeinsames Verständnis herzustellen [GF15], Missverständnisse aufzudecken und Informationen weiter zu konkretisieren [Pa14].

Je nach Art der Artefakte werden unterschiedliche Aspekte von Anforderungen besonders in den Vordergrund gerückt oder weggelassen. Story Cards stellen Anforderungen als einzelne nutzerorientierte Interaktionen dar und fördern so die Priorisierung, Schätzung und Detail-Diskussion einzelner Anforderungen. Abhängigkeiten zwischen Anforderungen sind so jedoch häufig nicht gut sichtbar. Geschäftsprozessmodelle hingegen bringen Aktivitäten der Softwarenutzer in eine Reihenfolge und weisen so auf Abhängigkeiten hin. Die einzelnen Modellelemente lassen sich jedoch wiederum schlecht für sich priorisieren oder beispielsweise zu Komponenten zuordnen.

¹ Gottfried Wilhelm Leibniz Universität Hannover, Fachgebiet Software Engineering, olga.liskin@gmail.com

Oft werden in einem Projekt mehrere Arten von Artefakten gemeinsam verwendet, um für die unterschiedlichen Aktivitäten jeweils eine passende Repräsentation parat zu haben. Die Informationen in den unterschiedlichen Artefakten überlappen sich dabei teilweise, was zu Abhängigkeiten zwischen Artefaktarten führt. Wird am Prozessmodell eine Aktivität verändert, so muss diese Änderung möglicherweise auch in den User Stories durchgeführt werden, um Inkonsistenzen zu vermeiden.

Die meisten Vorgehensmodelle, wie SCRUM [SB02], V-Modell [RB08] oder RUP [Kr04], folgen einem agilen oder einem traditionellen, das bedeutet plangetriebenen, Entwicklungsansatz. Sie unterscheiden sich unter anderem darin, welche Artefaktarten sie verwenden. In agilen Ansätzen wird die Dokumentation so leichtgewichtig wie möglich gestaltet. Es werden einige wenige Artefaktarten, wie Epics, User Stories und Abnahmetests verwendet und die einzelnen Artefakte so simpel wie möglich gehalten [BA04]. In traditionellen Methoden wird Wert darauf gelegt, dass die Dokumentation möglichst vollständig und möglichst detailliert ist [BT03]. Neben rein agilen oder traditionellen Ansätzen werden auch immer häufiger hybride Entwicklungsansätze eingesetzt, die agile und plangetriebene Methoden vereinen [BT03], [Th15]. Da die typischen Artefakte der einzelnen Ansätze grundsätzlich verschieden sind, können hier besondere Herausforderungen durch deren Abhängigkeiten entstehen. Während die Artefakte, die agil verwendet werden, sich häufig und leicht ändern, sollen Artefakte im plangetriebenen Vorgehen eher einen festen Planungs-Rahmen darstellen und gleich bleiben.

Wie ein gutes Zusammenspiel zwischen verschiedenen Anforderungsartefakten – vor allem auch in hybriden Ansätzen – erreicht werden kann, bildet daher die Forschungsfrage für diese Arbeit.

1.1. Beiträge der Dissertation

Zunächst fehlt in der Wissenschaft bisher ein Überblick darüber, wie in der Praxis Anforderungsartefakte verwendet werden. Dazu gehören Fragestellungen, wie, welche Artefakte Projektteilnehmer aus welchen Gründen einsetzen und welche Herausforderungen sie dabei sehen. Mit einer Studie über den Stand der Praxis leistet diese Dissertation einen ersten Beitrag, indem sie diesen Überblick herstellt. Wie die Studie zeigt, ist ein gutes Zusammenspiel zwischen Anforderungsartefakten ein relevanter Faktor, ohne den die Anforderungskommunikation schnell ins Stocken kommt.

Bisherige Ansätze in der Wissenschaft beschäftigen sich damit, wie die in einem Projekt verwendeten Artefakttypen und deren Verknüpfungen modelliert werden können. Dazu werden sogenannte Traceability-Modelle aufgestellt [Ad13], [MGP09]. Ein Defizit dieser Modelle ist, dass sie nur explizit definierte Verknüpfungen betrachten, jedoch keine weiteren Aspekte modellieren, die das Zusammenspiel von Artefakten beeinflussen. Weitere Aspekte wären zum Beispiel Abhängigkeiten zwischen Artefaktarten oder die Anpassbarkeit eines Artefakts. Außerdem fehlt eine Hilfestellung zur Erstellung inhaltlich guter Modelle. Es bleibt offen, wann nun zwischen zwei bestimmten Artefakttypen ein bestimmter Verknüpfungstyp eingesetzt werden sollte und wann nicht. Genau das ist aber wichtig für die Erstellung hybrider Entwicklungsansätze oder die Individualisierung bestehender Vorgehensmodelle, wie RUP, SCRUM oder V-Modell XT.

Die vorliegende Dissertation schließt diese beiden Lücken, indem sie eine Erweiterung des Traceability-Modell-Konzeptes erstellt. Mithilfe von Anforderungslandschaften werden neben den Artefakttypen und ihren Verknüpfungen weitere Eigenschaften abgebildet, die für die Koexistenz von Artefakten von Bedeutung sind. Aufbauend darauf stellt die Arbeit ein Verfahren zur inhaltlichen Planung und Optimierung von Anforderungslandschaften in Hinblick auf Koexistenz vor. Ein risikobasierter Ansatz gibt Hilfestellung dabei, ob beispielsweise bestimmte Artefakttypen eingeführt und wann diese durch Traceability-Links verknüpft werden sollen. Der Ansatz basiert auf der Praxisstudie und greift damit genau die Probleme aus der Praxis auf.

Um weiterhin mit den Abhängigkeiten gut umgehen zu können, sind Tracing-Methoden notwendig. Diese dokumentieren Abhängigkeiten durch explizite Verknüpfungen und erlauben, schnell zu passenden Informationen zu gelangen. Die automatisierte Verwendung von Verknüpfungen zur Identifikation von relevanten Anforderungsartefakten wird durch das Konzept der Trace Queries [MC13] ermöglicht. Damit werden Suchanfragen auf Projektartefakten formuliert, die dann automatisiert durchgeführt werden. Allerdings müssen Trace Queries bisher manuell und auch selbstmotiviert angestoßen werden.

Um diese Lücke zu schließen, stellt diese Dissertation das Konzept der Trace-Operationen vor. Trace-Operationen bauen auf Trace Queries auf, erweitern diese aber um Auslöser und Überprüfungen, welche es erlauben, Trace Queries automatisiert anzustoßen und die Ergebnisse automatisiert auszuwerten. Durch die Erweiterung wird es möglich, Tracing-Aktivitäten in die eigentlichen anforderungsbezogenen Aktivitäten, wie die Ausarbeitung oder Priorisierung von Anforderungen, zu integrieren.

2. Stand der Praxis: Interviewstudie

Um einen Überblick über den Stand der Praxis zu bekommen, wird im Zuge der Dissertation zunächst eine Interviewstudie durchgeführt. Es wurden 21 Praktiker aus 6 Unternehmen in semi-strukturierten Interviews mit einer durchschnittlichen Dauer von ca. 75 Minuten befragt. Die Interviews wurden aufgezeichnet, transkribiert und die Aussagen im Anschluss kodiert und kategorisiert im Sinne der Aktivitäten aus der Grounded Theory (*Open Coding*, *Selective Coding* und *Categorization*) [GS67].

Die Praxiserfahrungen bestätigen, dass für verschiedene Aktivitäten unterschiedliche Arten von Artefakten gewählt werden. Zum Sammeln der Anforderungen werden häufig generische Dokumente (wie Word-Dokumente) verwendet. Diese erlauben, Information auf beliebige Art darzustellen, während beispielsweise Einzelelemente (wie User Stories) meist striktere Vorgaben haben, wie dass dort nur Nutzerinteraktionen festgehalten werden. Viele Praktiker haben berichtet, dass jedoch die Informationen in generischen Spezifikationsdokumenten oft zu vage und gleichzeitig an anderen Stellen zu spezifisch sind. Hier werden gern Einzelelemente, wie Use Cases oder User Stories verwendet, die helfen, das große Problem in kleine Teile zu teilen und auf diese dann nach und nach im angemessenen Detailgrad einzugehen. Doch auch hier können sich Probleme ergeben, wenn die Elemente zu technisch sind. Dann können Anforderungsgeber sie nicht mehr selbstständig verstehen und richtig priorisieren. Zur Kommunikation mit Anforderungsgebern wurden konkrete Modelle, wie GUI-Mockups als bestgeeigneter Artefakttyp ge-

nannt. Insgesamt wurden immer mehrere Arten von Artefakten verwendet, um die verschiedenen Blickwinkel abzudecken. Dabei haben sich neue Herausforderungen aus den Abhängigkeiten zwischen den Artefakten herausgestellt.

Mehr als 70% der Interview-Teilnehmer sind beim Umgang mit mehreren Artefakten auf Probleme gestoßen. Neben dem Zusatzaufwand zum Dokumentieren und Finden von Informationen an mehreren Orten, hatten sie Schwierigkeiten durch Inkonsistenzen erlebt. Außerdem wurde erwähnt, dass es schwierig war, zu prüfen, ob alle relevanten Elemente aus einem Artefakttyp auch korrekt in einem anderen abhängigen Artefakt auftraten. Die potenziellen Probleme führten zu zusätzlichem Aufwand zur Vermeidung des Eintretens der Probleme. Wenn Informationen übersehen wurden, so traten Missverständnisse oder falsche Annahmen auf, welche wiederum potenziell zu höheren Kosten aufgrund von Fehlern führten. Ein weiterer berichteter Effekt war, dass Projektteilnehmer sehr schnell das Vertrauen in ein Dokument verloren – und aufhörten es zu verwenden – wenn sie mehrmals auf Inhalte stießen, die nicht aktuell oder inkonsistent waren.

3. Ein Konzept zur Unterstützung der Koexistenz von agilen und traditionellen Anforderungsartefakten

In der Interview-Studie wurde gezeigt, dass häufig mehrere Artefakttypen gemeinsam verwendet werden und dass es häufig zu Schwierigkeiten kommt, weil die Artefakte zusammenhängen, diese Zusammenhänge jedoch nicht ausreichend unterstützt werden. Um mit diesen Problemen besser umgehen zu können, ist das Ziel des in der Dissertation vorgestellten Konzeptes die *Verbesserung der Unterstützung der Koexistenz von agilen und traditionellen Anforderungsartefakten*.

Eine Verbesserung lässt sich auf *zwei Ebenen* erreichen. Zum einen ist es auf der *methodischen Ebene* möglich, die Koexistenz von Anforderungsartefakten zu beeinflussen, indem eine explizite Analyse und Planung dazu durchgeführt wird, welche Typen von Anforderungsartefakten eingesetzt und welche Arten von Verknüpfungen besonders unterstützt werden. Der Aufbau mit verwendeten *Typen* von Artefakten und Verknüpfungen wird als **Anforderungslandschaft** bezeichnet. Die zweite Ebene betrifft die *konkrete Projektebene* mit den konkreten darin erstellten *Dokumenten*. Hier lässt sich die Koexistenz von Anforderungsartefakten beeinflussen, indem Verknüpfungen zwischen konkreten Anforderungen gefördert und auch besser ausgenutzt werden. Dies geschieht durch sogenannte **Trace-Operationen**. Abbildung 1 zeigt die beiden Ebenen und ihre Zusammenhänge.

4. Anforderungslandschaften

Anforderungslandschaften zu einem Projekt sind Modelle der verwendeten *Typen* von Anforderungen und Verknüpfungen im Projekt. Sie stellen dar, wie die Anforderungsmenge aufgebaut ist und zeigen Eigenschaften auf, die relevant für den guten Umgang mit Anforderungen und die Koexistenz von Anforderungsartefakten sind. Das Modell

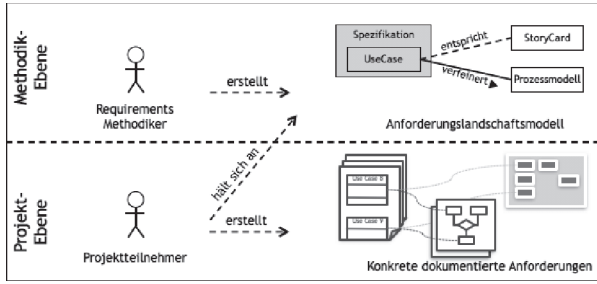


Abb. 1: Übersicht über Anwendungsebenen der hier erarbeiteten Methodik

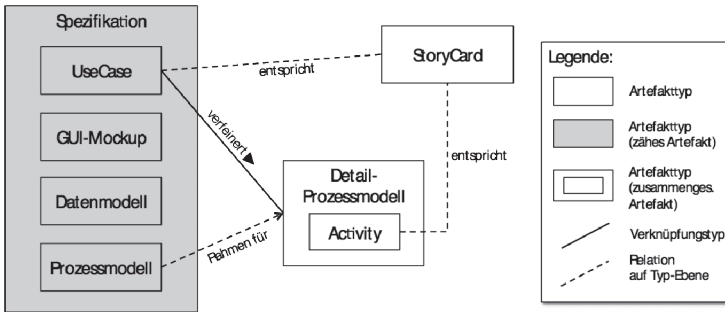


Abb. 2: Beispiel für ein Anforderungslandschaftsmodell

der Anforderungslandschaft zusammen mit der Darstellung wird in der Dissertation basierend auf Erkenntnissen aus der Interview-Studie erarbeitet. Abbildung 2 zeigt ein Beispiel für eine Anforderungslandschaft zusammen mit den relevanten Eigenschaften.

Zunächst ist zu klären, welche Arten von Anforderungsartefakten verwendet werden – beispielsweise ob es eine Spezifikation gibt, ob diese Spezifikation auch Geschäftsprozessmodelle enthält, ob zudem GUI-Mockups verwendet werden. Zu den Artefakttypen ist zudem relevant, in welchen Beziehungen diese zueinander stehen.

In dieser Arbeit wird – im Gegensatz zu vielen Ansätzen in der Literatur – insbesondere zwischen einer *Relation zwischen zwei Artefakten* und einer *expliziten Verknüpfung zwischen zwei Artefakten* unterschieden. Mit *Verknüpfung* wird ein dokumentierter Link bezeichnet. Zwei Artefakte, die in *Relation* zueinander stehen, tun dies nun aber unabhängig davon, ob eine explizite Verknüpfung zwischen ihnen definiert ist oder nicht. Auch wenn zwei Artefakte nicht verknüpft sind, kann es beispielsweise gewünscht sein, nach der Änderung eines Artefaktes ein *abhängiges* anderes Artefakt anzupassen.

Weiterhin ist wichtig, zu einem Artefakt zu betrachten, wie es sich bei Änderungen verhält. Ob die Anforderungen, die in einem Artefakt festgehalten sind, ohne weiteres geändert werden können oder nicht, wird durch die *Zähheit* festgehalten. Ein Artefakt gilt als

zäh, wenn es nur in Absprache geändert werden darf oder wenn es aufgrund seiner Struktur schwierig ist, dort eine Änderung einzuarbeiten.

Eine weitere eigene Darstellung gibt es für zusammengesetzte Artefaktarten. Das sind Artefakte, die in einer *enthält*-Relation stehen. Beispielsweise kann ein Use Case mehrere Use Case-Schritte enthalten. Diese Relation stellt eine besondere Form der Dokumentation dar: Das enthaltene Artefakt ist Teil des übergeordneten enthaltenden Artefaktes und wird zusammen mit diesem dargestellt. Das besondere an dieser Relation ist, dass hier Abhängigkeiten leichter behandelt werden können, da beide Artefakte zusammen aufgeführt sind.

4.1. Hybride Anforderungslandschaften

Hybride Entwicklungsansätze vereinen agile und traditionelle Entwicklungsmethoden, um die Vorteile aus beiden Ansätzen zu kombinieren. Ein hybrider Entwicklungsansatz soll ermöglichen, schnell und flexibel auf Änderungen zu reagieren, gleichzeitig aber auch Sicherheit und Nachvollziehbarkeit bieten. Daran angelehnt, kombinieren hybride Anforderungslandschaften agile und traditionelle Anforderungsartefakte und ermöglichen damit auch, ihre jeweiligen Vorzüge zu nutzen. Charakteristisch für eine hybride Anforderungslandschaft ist, dass diese flexible sowie zähe Anforderungsartefakte besitzt. Hybride Landschaften entstehen beispielsweise, weil in traditionellen Projekten agile Artefakte, wie User Stories hinzugefügt werden, um besser auf Änderungen reagieren zu können und höhere Transparenz zur Umsetzung von Kundenwünschen zu bekommen. Alternativ entstehen hybride Landschaften, wenn in agilen Projekten zusätzliche Artefakte hinzugefügt werden, die einen festen Rahmen darstellen. Dies kann notwendig sein, um einen besseren Gesamtüberblick für große Teams zu schaffen oder um Überanpassung durch zu viele Anforderungsgeber zu vermeiden.

4.2. Risikobasierte Optimierung von Anforderungslandschaften

In der Dissertation wird ein risikobasiertes Verfahren vorgestellt, das ermöglicht, den inhaltlichen Aufbau einer Anforderungslandschaft zu optimieren. Es baut auf dem *risikobasierten Ansatz zum Ausbalancieren von agilen und plangetriebenen Methoden* von Boehm und Turner auf [BT03]. Das Verfahren zielt darauf ab, die Projektrisiken, die mit einer Vorgehensweise verbunden sind, zu minimieren. Mögliche Alternativen bergen unterschiedliche Risiken, die im eigenen Projektkontext bewertet werden müssen. Hat eine der Alternativen ein geringeres Risiko, so wird die Anforderungslandschaft auf diese Alternative ausgerichtet.

Hierzu leitet die Dissertation aus den Vor- und Nachteilen, die in der Interview-Studie genannt wurden, **Risikofaktoren** ab. Diese Risikofaktoren können nun bewertet werden, indem das Risiko durch das Produkt („*Risk Exposure*“) seiner Eintrittswahrscheinlichkeit und des resultierenden Schadens charakterisiert wird. Durch eine Bewertung der Risikofaktoren kann der Requirements Methodiker sich schließlich ein Bild verschaffen, ob eine der Alternativen risikogünstiger ist und dies als Entscheidungshilfe bei der Optimierung nutzen. Abbildung 3 zeigt hierzu ein Beispiel. Im Beispiel wird der zweite Risikofaktor in der Tabelle mit einer 10%igen Eintrittswahrscheinlichkeit und einem Schaden

von 50 eingeschätzt. Das Gesamtrisiko der Alternative, den Artefakttyp aufzunehmen wird mit einer gesamten Risk Exposure von 8 bewertet. Das Gesamtrisiko der Alternative, den Artefakttyp nicht aufzunehmen beträgt 26, es ist also deutlich höher. Das Artefakt sollte daher in die Landschaft aufgenommen werden.

Optimierungsdimension Artefakt			
Konkretes bewertetes Artefakt: User Stories			
Risikofaktor	Eintrittswahrscheinlichkeit (0-1)	Schaden (0-100)	Risk Exposure
Artefakttyp wird aufgenommen			Summe: 8
Analysephase verlängert sich	0,1	30	3
Zu detaillierte Spezifikation legt Projekt auf ungünstige Lösungen fest	0,1	50	5
Artefakttyp wird nicht aufgenommen			Summe: 26
Projekt läuft aus dem Rahmen	0,6	30	18
Viele Stakeholder -> Über-Anpassung wegen fehlendem Bezugspunkt	0,2	40	8

Bewertung eines einzelnen Risikofaktors

Bewertung des Gesamtrisikos einer Alternative

Abb. 3: Optimierung der Nutzung von User Stories durch Bewertung der Risikofaktoren

Insgesamt stellt die Arbeit aus unterschiedlichsten Aussagen zu Vor- und Nachteilen bezüglich des Aufbaus von Anforderungslandschaften 27 Risikofaktoren zu 5 Optimierungsdimensionen auf. Das Verfahren stellt eine Möglichkeit dar, systematisch auf die vielfältigen Praxiserfahrungen aus der Interviewstudie zuzugreifen.

5. Trace-Operationen

Modellierung sowie Planung und Optimierung des Aufbaus von Anforderungslandschaften sind wichtige erste Schritte zum effektiven Umgang mit Anforderungen und damit zu guter Anforderungskommunikation. Dennoch ist es für eine gute Koexistenz genauso wichtig, auch die Projektteilnehmer in ihren alltäglichen Aufgaben im Umgang mit Anforderungen zu unterstützen. Mithilfe von Verknüpfungen (*Traces*) können Teile anforderungsbezogener Aktivitäten teil-automatisiert werden. Ein System kann anhand von Verknüpfungen die Identifikation und Präsentation von abhängigen Elementen übernehmen und die Projektteilnehmer an den richtigen Stellen auf mögliche Abhängigkeiten aufmerksam machen. Dabei ist auch hier wieder von Projekt zu Projekt verschieden, welche Verknüpfungen welcher automatisierten Unterstützung bedürfen. Entsprechend wird eine Möglichkeit benötigt, automatisierbare Operationen individuell für ein Projekt zu konfigurieren. Hierzu erarbeitet die Dissertation das Konzept der *Trace-Operationen*. Abbildung 4 zeigt eine Auswahl an Trace-Operationen und wie diese für die unterschiedlichen Verknüpfungen verwendet werden können.

Jede Trace-Operation besitzt eine *Trace Query* im Zentrum. Damit wird festgelegt, wie von einem bestimmten Ausgangsartefakt aus zu anderen relevanten Artefakten oder Informationen navigiert wird. *Kritiken* definieren zusätzlich eine *Prüfung* des Trace Query-Resultates. Wenn die Auswertung *wahr* liefert, so wird die *Reaktion* ausgelöst, die beispielsweise eine Warnung anzeigt.

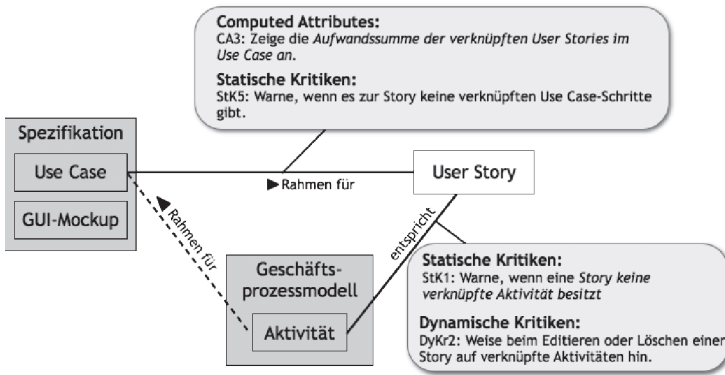


Abb. 4: Beispiel für Trace-Operationen in einer Anforderungslandschaft

In der Dissertation werden zwei Möglichkeiten vorgestellt, solche Trace-Operationen zu beschreiben. Sie unterscheiden sich im Wesentlichen darin, wie die Trace Query dargestellt wird. Zum einen werden die Trace Query und ihre Überprüfung textuell mittels der Object Constraint Language (OCL) beschrieben. In der zweiten Herangehensweise wird die Trace Query grafisch dargestellt. Diese Darstellung basiert auf der Visual Trace Modeling Language (VTML), die durch Mäder und Cleland-Huang [MC13] entwickelt wurde. VTML-Queries basieren auf UML-Klassendiagrammen, die einen Teil eines Artefaktmodells aufgreifen und durch zusätzliche Symbole für Einschränkungen und Aggregationen ergänzen. Sie erlauben es, eine Navigation im Landschaftsmodell visuell darzustellen, was für Nutzer einfacher sein kann als textuelle Beschreibungen. Abbildung 5 zeigt ein Beispiel, in dem eine Trace-Operation jeweils mit beiden Methoden formalisiert ist.

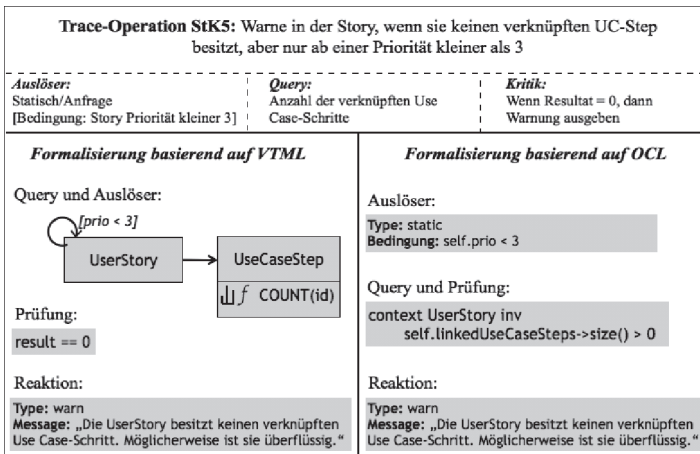


Abb. 5: Beispielhafte Formalisierung einer Trace-Operation basierend auf VTML und OCL

5.1. Validierung des Nutzens von Trace-Operationen

Inwiefern Trace-Operationen Projektteilnehmer tatsächlich dabei unterstützen können, besser mit Anforderungen umzugehen, wird in einem *randomisierten Experiment* untersucht. Teilnehmer zweier Gruppen erhalten dieselbe *Anforderungsmenge* in derselben *Sicht* und führen dieselben *anforderungsbezogenen Aktivitäten* durch. Die *experimentelle Behandlung*, die eine der beiden Gruppen (*LinkedArtifacts*) erhält, ist, dass ihre Anforderungsmenge mit expliziten Verknüpfungen versehen ist und sie durch Trace-Operationen unterstützt wird.

Die Studie zeigt, dass die Performance der *LinkedArtifacts*-Gruppe bei komplexen Aufgaben, die erfordern, dass verschiedene Arten von Artefakten betrachtet werden, tatsächlich besser ist. Bei simplen Aufgaben, bei denen es ausreicht, Artefakte von nur einer Art zu betrachten, ist die Performance bei beiden Gruppen ähnlich. Das typische Problem, dass ein Projektteilnehmer schnell ein primäres Artefakt ändert, jedoch vergisst, die Änderung auch auf andere verwandte Artefakte zu übertragen, ist unabhängig davon aufgetreten, ob Trace-Operationen verwendet wurden. Es hat sich gezeigt, dass stärkere Mechanismen, als die bloße Hervorhebung verknüpfter Artefakte, notwendig sind, um eine ausreichende Berücksichtigung von Abhängigkeiten zu bewirken.

6. Zusammenfassung und Ausblick

Es wurde ein Ansatz für die Unterstützung der Koexistenz von Anforderungsartefakten erarbeitet. Dazu wurde zunächst in einer Studie herausgearbeitet, welche Aktivitäten von Anforderungsartefakten beeinflusst werden. Ein auf den identifizierten Herausforderungen aufbauendes Konzept zur Modellierung ermöglicht es, in einem Projekt eine Anforderungslandschaft aufzubauen, welche die Blickwinkel und Erfordernisse der Projektteilnehmer abdeckt und die entsprechenden Artefakte integriert. Das Konzept der Trace-Operationen erweitert Anforderungslandschaften um zusätzliche Tool-Unterstützung für den alltäglichen Umgang mit Anforderungen und deren Abhängigkeiten.

In Zukunft könnten durch die stärkere Verbindung von Verknüpfungstypen und Trace-Operationen neue Möglichkeiten geschaffen werden, die beiden Konzepte noch umfangreicher zu verwenden. Beispielsweise kann das Konzept so erweitert werden, dass eine *Verknüpfungsstärke* als Attribut von Verknüpfungstypen direkt festlegt, welche Richtlinien und Trace-Operationen mit der Verknüpfung verbunden werden. Andersherum kann der Umgang mit Anforderungen Hinweise auf Abhängigkeiten geben.

7. Literaturverzeichnis

- [Ad13] J. Adersberger, Modellbasierte Extraktion, Repräsentation und Analyse von Traceability-Informationen. Dissertation. Herbert Utz Verlag, 2013.
- [BA04] K. Beck and C. Andres, Extreme Programming Explained: Embrace Change, 2nd Edition. Addison Wesley, 2004.

- [BT03] B. Boehm and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed*. Pearson Education, 2003.
- [Bo16] O. Boruszewski: *Unterstützung der Koexistenz von agilen und traditionellen Anforderungsartefakten*. epubli, 2016.
- [GS67] B. G. Glaser and A. L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine de Gruyter, 1967.
- [GF15] M. Glinz and S. A. Fricker, “On shared understanding in software engineering: an essay,” *Comput. Sci. - Res. Dev.*, vol. 30, no. 3, pp. 363–376, 2015.
- [Pa14] J. Patton, *User Story Mapping*. O’Reilly Media, 2014.
- [Kr04] P. Kruchten, *The Rational Unified Process: an Introduction*. Addison-Wesley Professional, 2004.
- [MC13] P. Mäder and J. Cleland-Huang, “A visual language for modeling and executing traceability queries,” *Softw. Syst. Model.*, vol. 12, no. 3, pp. 537–553, 2013.
- [MGP09] P. Mäder, O. Gotel, and I. Philippow, “Getting back to basics: Promoting the use of a traceability information model in practice,” in *Proceedings of the ICSE Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE)*, 2009, pp. 21–25.
- [RB08] A. Rausch and M. Broy, “Die V-Modell XT Grundlagen,” in *Das V-Modell XT: Grundlagen, Methodik und Anwendungen*, R. Höhn and S. Höppner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–27.
- [SB02] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. Prentice Hall, 2002.
- [Th15] G. Theocharis, M. Kuhrmann, J. Münch, and P. Diebold, “Is Water-Scrum-Fall Reality? On the Use of Agile and Traditional Development Practices,” in *Proceedings of the 16th International Conference on Product-Focused Software Process Improvement (PROFES)*. Springer International Publishing, 2015, pp. 149–166.

Olga Boruszewski



Olga Boruszewski (geb. Liskin) hat an der Gottfried Wilhelm Leibniz Universität in Hannover Mathematik mit der Studienrichtung Informatik studiert. Dort hat sie an verschiedenen Softwareprojekt-Lehrveranstaltungen teilgenommen und auch als Tutorin den Übungsbetrieb in der Softwaretechnik und Softwarequalität unterstützt. In einem Auslandssemester an der University of Colorado in Boulder, USA, hat sie weitere Themen der Softwareentwicklung, wie Usability Engineering, vertieft sowie ein Praktikum als Softwareentwicklerin in einem agilen Entwicklungsunternehmen absolviert.

Im Anschluss an das Studium hat sie am Fachgebiet für Software Engineering der Leibniz Universität Hannover promoviert. Dort hat sie sich mit Requirements Engineering, agilen Softwareentwicklungsmethoden sowie Informationsflüssen in der Teamkommunikation beschäftigt. Ihre Interessenschwerpunkte liegen auf Anforderungsartefakten, der Anforderungskommunikation sowie auf agilen Softwareentwicklungsmethoden.