

# Anforderungen an Software Requirement Pattern in der Entwicklung sozio-technischer Systeme

Axel Hoffmann, Holger Hoffmann, Jan Marco Leimeister

Fachgebiet Wirtschaftsinformatik  
Universität Kassel  
Pfannkuchstr. 1  
34121 Kassel  
axel.hoffmann@uni-kassel.de  
holger.hoffmann@uni-kassel.de  
leimeister@uni-kassel.de

**Abstract:** Software Requirement Pattern werden in der Systementwicklung zur Erhebung und Analyse von Anforderungen eingesetzt. Die Requirement Pattern kapseln Wissen unterschiedlicher Domänen und können dem Anforderungsanalysten in der sozio-technischen Systementwicklung bei der Kommunikation mit den Stakeholdern helfen. Der Beitrag beschreibt Anforderungen an Requirement Pattern in der sozio-technischen Systementwicklung, die in einem Expertenworkshop erhoben wurden. Zur Erfüllung der Anforderungen wird anschließend eine Struktur für Requirement Pattern vorgeschlagen. Die Verwendung der Struktur erweitert die Nützlichkeit von Requirement Pattern in der Entwicklung sozio-technischer Systeme und kann als Grundlage für einen einheitlichen Pattern Katalog genutzt werden.

## 1 Einleitung

Als eine der wesentlichen Herausforderungen der sozio-technischen Systementwicklung führen Baxter und Sommerville die interdisziplinäre Zusammensetzung des Entwicklerteams auf [BS11]. Jedoch ist ein interdisziplinäres Vorgehen zur Entwicklung notwendig und sollte demnach auch genutzt werden. In unseren eigenen Entwicklungsprojekten haben wir durch das Zusammenbringen von Experten aus den Bereichen Recht, Vertrauen, Gebrauchstauglichkeit und Informatik [HL11; HSF11] in Entwicklerteams für sozio-technische Systeme die Auswirkungen dieser Herausforderung erlebt: Stakeholder eines Bereiches erfassen oft nicht die Fähigkeiten und Kompetenzen anderer Stakeholder, mit der Folge, dass Artefakte, die zwischen den Beteiligten ausgetauscht werden, von den anderen Stakeholdern nicht genutzt werden können [BN98; BS11]. Daraus resultieren für die Anforderungsanalyse Verständnis- und Kommunikationsprobleme. Zusätzlich sind in der Anforderungsanalyse überlappende Anforderungen mit widersprüchlichen Eigenschaften üblich [BS11]. Diese gegensätzlichen Standpunkte können durch die unterschiedlichen Interessen und Motivationen der beteiligten Stakeholder nicht rational miteinander vereinbart werden.

Zur Überwindung der Herausforderung empfehlen Baxter und Sommerville [BS11], dass eine Person (oder ein Team) über die Kenntnisse und Fähigkeiten der anderen Bereiche verfügen sollte, um mit allen Beteiligten der sozio-technischen Systementwicklung effektiv kommunizieren zu können. Diese Funktion kann in der Anforderungsanalyse der Anforderungsanalyst einnehmen. Das notwendige Wissen zur Unterstützung bei der Sammlung und Analyse von Anforderungen verschiedener Stakeholder und zur Umsetzung dieser Anforderungen in eine Anforderungsspezifikation kann mit Hilfe von Software Requirement Pattern bereitgestellt werden.

Requirement Pattern können zusätzlich dazu dienen, den durch den Einsatz verschiedener Experten verursachten Mehraufwand der sozio-technische Systementwicklung gegenüber traditionelle Entwicklungsansätze [CEG12] zu reduzieren. In der Erfahrung hat sich gezeigt, dass gleiche oder ähnliche Anforderungen der Experten immer wieder von Bedeutung sind. Durch die Identifikation wiederkehrender Anforderungen und der Bereitstellung als Requirement Pattern kann der Aufwand während der Anforderungsanalyse gesenkt werden. Anforderungsanalysten wären in die Lage versetzt, einen die Anforderungen selbst zu erkennen und aufzunehmen.

Der Beitrag untersucht Anforderungen aus der Sicht der Anforderungsanalysten und Softwareentwickler an die Struktur von Requirement Pattern, um die Aufgabe erfüllen zu können. Die Anforderungen wurden mittels eines Expertenworkshops gesammelt. Zusätzlich wird eine angepasste Struktur für sozio-technische Requirement Pattern vorgeschlagen. Diese basiert auf dem Metamodell von Franch et al. [FPQ10], die dazu aufrufen zu untersuchen, ob die von Ihnen vorgeschlagene Struktur, die für Requirement Pattern der Softwarequalität entwickelt wurden, auch für Requirement Pattern andere Bereiche eignet [FPQ10].

Im folgenden Kapitel werden die Eigenschaften sozio-technische Entwicklung kurz vorgestellt und Entwicklungsansätze für deren Umsetzung skizziert. Kapitel 3 beschreibt den Einsatz von Pattern in der Softwareentwicklung im Allgemeinen und in der Anforderungsanalyse im Speziellen. In Kapitel 4 wird das Forschungsdesign beschrieben, bevor in Kapitel 5 die gesammelten Anforderungen und die vorgeschlagene Struktur für Requirement Pattern für die sozio-technische Systementwicklung vorgestellt werden. In Kapitel 6 wird diskutiert, in wie weit die Anforderungen von der Struktur erfüllt werden und welche Limitationen zu beachten sind. Anschließend wird der Beitrag durch eine Zusammenfassung abgeschlossen.

## **2 Sozio-technisches System Design**

Als Grundlage für die Betrachtung der Entwicklung sozio-technischer Systeme wurde der VENUS-Ansatz genutzt, der mit Hilfe von Experten die Eigenschaften Rechtsverträglichkeit, Gebrauchstauglichkeit und Vertrauenswürdigkeit während der Systementwicklung besonders berücksichtigt. Der erste Abschnitt fasst die Eigenschaften der sozio-technischen Systementwicklung zusammen. Der zweite Abschnitt gibt einen kurzen Überblick über die Phasen und Aktivitäten des konkreten

Entwicklungsansatzes, in dem die Herausforderungen aufgetreten sind und der durch die Requirement Pattern unterstützt werden soll. Eine ausführlichere Beschreibung des Ansatzes ist in Hoffmann et al. [HSF11], die Anwendung in einer Fallstudie in Comes et. al. [CEG12] zu finden.

## **2.1 Eigenschaften sozio-technischer Systementwicklung**

Sozio-technisches Systemdesign betrachtet die Gestaltung eines technischen Systems, das in einem sozialen System genutzt werden soll. Bei der Entwicklung eines technischen Systems ist hiernach immer dessen Einfluss auf das soziale System, in dem es eingesetzt werden soll zu beachten. Die Gestaltungsparameter des sozio-technischen Systemdesigns gehen über die technischen Komponenten hinaus, so werden bspw. Verwendungsprozesse, Anreizstrukturen, u.v.m. berücksichtigt und mit gestaltet. Zielsetzung des sozio-technischen Systemdesigns ist ein genutztes technisches System, das in seiner Verwendung sozial akzeptabel, technisch stabil und ökonomisch sinnvoll sein soll [LSK06].

Der Zusammenhang zwischen sozialem und technischem System ist bei der Entwicklung von sogenannten sozio-technischen Systemen, also von Menschen genutzten Systemen, zentral. Folglich gilt es, prognostizierbare Ursache-Wirkungs-Zusammenhänge zwischen technischem und sozialem System in der späteren Nutzung möglichst im Entwicklungsprozess bereits zu berücksichtigen.

Da es das Ziel des sozio-technischen Systemdesigns ist, ein genutztes System zu entwickeln, gewinnen Aspekte, welche die Akzeptanz sozio-technischer Systeme beim Nutzer unterstützen, z.B. Vertrauen in das System, bei der Entwicklung solcher Systeme an Bedeutung. Durch die Beachtung dieser Aspekte entsteht eine höhere Chance, dass das entwickelte sozio-technische System auch von einer Vielzahl von Anwendern genutzt wird und der Einsatz sowohl die gewünschten Vorteile für den Nutzer mit sich bringt, als auch keine ungewünschten, möglicherweise negativen Auswirkungen auf das soziale System, in dem es eingesetzt wird, nach sich zieht.

## **2.2 Der VENUS-Ansatz zur Entwicklung sozio-technischer Systeme**

Die Herausforderungen im Bereich von Kommunikation und Verständnis der beteiligten Stakeholder, welche die Entwicklung von Requirement Pattern motivieren, sind bei der Verwendung des VENUS-Ansatzes zur Entwicklung sozio-technischer Systeme [HSF11] aufgetreten. Der Kern des Entwicklungsansatzes ist eine iterative Entwicklung, welche aus Analyse, Konzeption und Softwaregestaltung, sowie aus der Implementierung und Evaluierung besteht (Abbildung 1). Die Phase des Anforderungsmanagement, das durch die Requirement Pattern unterstützt werden soll, und deren Einbettung in das gesamte Vorgehen wird im folgenden Abschnitt kurz beschrieben.

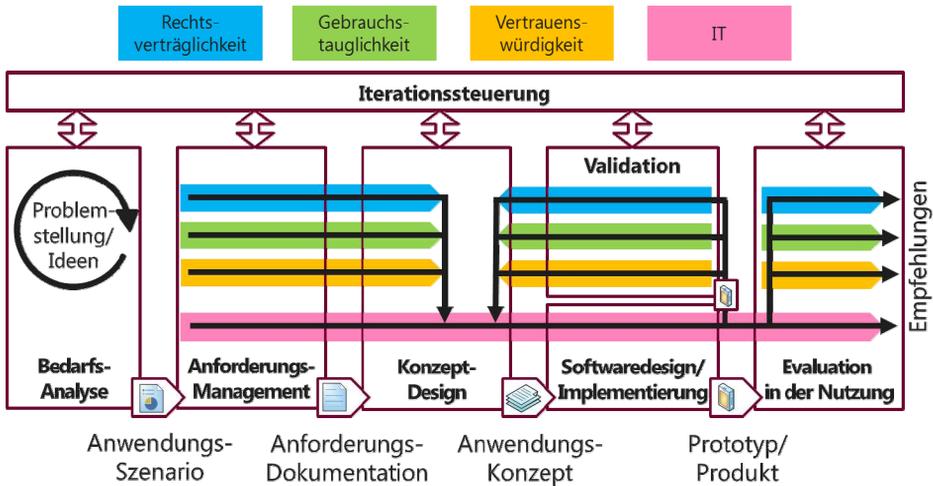


Abbildung 1: Überblick über den VENUS Ansatz zur sozio-technischen Systemgestaltung

Das Anforderungsmanagement bezieht Methoden für den expliziten Erwerb von Anforderungen aus Recht, Gebrauchstauglichkeit und Vertrauen mit ein. Der Ansatz berücksichtigt außerdem allgemein übliche Anforderungen der Systementwicklung [So11], wie beispielsweise Anforderungen der Stakeholder (z. B. Endnutzer, Manager oder Kunden). Um die Merkmale Rechtsverträglichkeit, Gebrauchstauglichkeit, Vertrauenswürdigkeit und IT in der Phase des Anforderungsmanagements zu koordinieren, ist es wichtig, bereits im Vorfeld ein gemeinsames Verständnis von der Anwendung (dem Produkt) zu entwickeln. Diese gemeinsame Version wird in der Bedarfsanalyse in Form eines Anwendungsszenarios erstellt. Das Anwendungsszenario stellt sicher, dass vor der Anforderungserhebung ein gemeinsames Verständnis über die Anwendung existiert und die Anforderungen der Merkmale Rechtsverträglichkeit, Gebrauchstauglichkeit, Vertrauenswürdigkeit und IT zielgerichtet erhoben werden können.

In einem ersten Evaluationsschritt werden die definierten Anwendungsszenarios Vertretern der Nutzergruppe vorgelegt und diese werden hinsichtlich ihrer Akzeptanz für ein solches Szenario und seine Teilaspekte befragt. Zudem wird der Einfluss von möglichen Veränderungen an dem Szenario auf die Nutzerakzeptanz abgefragt. Auf diese Weise können sehr frühzeitig für die spätere Akzeptanz kritische bzw. förderliche Designelemente identifiziert und die gewonnenen Erkenntnisse in der Gestaltung berücksichtigt werden.

Für den Bereich der rechtlichen Anforderungen wird in der VENUS-Methode der KORA-Ansatz verwendet. KORA steht für **K**onkretisierung **r**echtlicher Anforderungen. Die KORA Methode gewinnt technische Anforderungen aus rechtlichen Vorgaben, die sich aus den grundlegenden dauerhaft gültigen Rechtsnormen auf den oberen Ebenen der Rechtshierarchie – insbesondere der Verfassung – ergeben. Dabei geht die Rechtsverträglichkeit über das Konzept der Rechtmäßigkeit hinaus, indem sie nicht nur eine minimale, sondern eine optimale Umsetzung rechtlicher Anforderungen anstrebt.

Um Gebrauchstauglichkeitsanforderungen zu erheben, muss der Kontext des Gebrauchs verstanden werden. Dieser Kontext besteht aus den Nutzern, ihre Tätigkeiten, die Ausrüstung (z. B. Hardware und Software) sowie den körperlichen sowie dem sozialen Umfeld. Neben Anforderungen, welche üblicherweise durch die Einbeziehung von potentiellen Usern erarbeitet wurden, können Gebrauchstauglichkeitsbedingungen in Normen wie beispielsweise DIN ISO 9241-110 und DIN ISO 9241-12 genutzt werden. Die Gebrauchstauglichkeitsstimmungen werden analog zu den rechtlichen Bestimmungen von Experten, welche die Anwendung der Applikation im Hinterkopf haben, konkretisiert [BJH12]. Nach diesem Vorgehen können auch Gebrauchstauglichkeitsanforderungen, welche den Nutzern nicht bewusst sind, erfasst werden. Allerdings spielen Fachkenntnis über Bestimmungen und das Wissen über den Kontext des Gebrauchs eine entscheidende Rolle für den Erwerb von Gebrauchstauglichkeitsanforderungen.

Für den Bereich Vertrauen integriert der VENUS-Ansatz das Vorgehen des Trust Engineering. Trust Engineering nutzt theoretische Grundlagen von Vertrauen, um vertrauensunterstützende Komponenten einer Anwendung zu entwerfen. Dabei adressiert die Methode die Determinanten von Vertrauen, die durch spezielle Designentscheidungen beeinflusst werden können. Mit Hilfe des Szenarios werden im Interaktionsprozess des Nutzers mit der Anwendung Situationen identifiziert, in denen er mit verschiedenen Unsicherheiten konfrontiert ist. Die Unsicherheiten werden nach Relevanz priorisiert. Für die Unsicherheiten werden Determinanten von Vertrauen herausgesucht, mit denen diesen begegnet werden kann. Hier werden Anforderungen formuliert, welche die Erfüllung der vertrauensfördernden Determinanten verlangen.

Die Anforderungen aller Domänen (Rechtsverträglichkeit, Gebrauchstauglichkeit, Vertrauenswürdigkeit, IT) werden im nächsten Schritt vereinbart. Dies geschieht nach dem EasyWinWin-Vorgehen, ein werkzeuggestütztes Verfahren zur systematischen Erhebung und Verhandlung von Anforderungen in Software-Projekten. Als Ergebnis entsteht eine Anforderungsdokumentation, die als Artefakt in die nächste Phase überführt wird.

Im Konzeptdesign werden die Anforderungen in ein Anwendungskonzept überführt. Das Anwendungskonzept wird dann in einem iterativen Prozess implementiert. Dabei werden die entstehenden Prototypen durch Experten darauf überprüft, ob die vorher definierten Anforderungen bei der Umsetzung berücksichtigt wurden. Durch diese Validierung können Änderungen am Anwendungskonzept durchgeführt werden, die in die nächste Iteration eingehen. Zudem werden die im Prozess entstehenden Teilfunktionen aus Nutzersicht evaluiert. Am Ende des Entwicklungsprozesses steht die Evaluation in der Nutzung durch reale Versuchsnutzer. Hiermit soll sichergestellt werden, dass sich das entwickelte technische System im tatsächlichen sozio-technischen Kontext so verhält, wie es im Anwendungsszenario angedacht war. Dabei können durch die Evaluation in den Merkmalen Rechtsverträglichkeit, Gebrauchstauglichkeit, Vertrauenswürdigkeit und IT Empfehlungen für weitere Projekte generiert werden.

### 3 Software Requirement Pattern

Wiederverwendung ist eine anerkannte Praktik im Software-Engineering. Im Requirements Engineering kann die Wiederverwendung Anforderungsanalysten helfen Softwareanforderungen zu identifizieren und zu dokumentieren [RR06]. Software Requirement Pattern ist ein Ansatz um Anforderungen wiederzuverwenden [FPQ10]. Eine Pattern beschreibt generell eine Problemstellung, welche häufig auftaucht, und beschreibt den Kern der Lösung des Problems [Al79]. Requirement Pattern werden für die Anforderungserhebung und Anforderungsanalyse eingesetzt. Dabei gibt es verschiedene Ansätze, welche sich in Umfang, Darstellung und Einsatzbereich unterscheiden [FPQ10; HC07]. Neuere Ansätze, die Requirement Pattern zum Erheben von Anforderungen nutzen sind der Ansatz von Withall [Wi08] und das Pattern-based Requirements Elicitation (PABRE) von Renault, Mendez-Bonilla, Franch, und Quer [RM09a; RM09b].

Ein Pattern-basierter Ansatz kann den Aufwand für den Erhebung von Anforderungen für viele Entwicklungsprojekte verringern [HSH12]. Die möglichen Vorteile für Anforderungsanalysen sind nicht nur die Zeitersparnis für die Anforderungserhebung, sondern auch die Verbesserung der Qualität der erhaltenen Anforderungen [RM09b]. Daher ist die Wiederanwendung von Requirement Pattern die entscheidende Voraussetzung für die Anwendbarkeit in der Praxis.

### 4 Forschungsdesign

Der folgende Abschnitt nennt die Forschungsfragen und beschreibt das Vorgehen, dass zur Beantwortung herangezogen wurde. Folgende Forschungsfrage wurde untersucht:

- Welche Anforderungen ergeben sich an Software Requirement Pattern für die Entwicklung sozio-technischer Systeme.
- Welche Attribute einer Struktur ergeben sich aus den Anforderungen für Software Requirement Pattern?

#### 4.1 Anforderungserhebung für Software Requirement Pattern

Für die Beantwortung der ersten Forschungsfrage wurde ein Workshop mit sechs Experten durchgeführt. Der Workshop dauerte zwei Stunden und wurde als Präsenzworkshop durchgeführt. Die Teilnehmer verfügten alle über ein abgeschlossenes Hochschulstudium im Bereich der Informatik oder verwandten Bereichen, ein Teilnehmer verfügte über eine abgeschlossene Promotion im Bereich Informatik. Die Erfahrung auf dem Gebiet der Entwicklung sozio-technischer Systeme belief sich bei einem Teilnehmer auf ein Jahr, bei jeweils zwei Teilnehmern auf zwei beziehungsweise drei Jahre und bei einem Teilnehmer auf sechs Jahre.

Nach einer kurzen Begrüßung wurden die Herausforderungen in der Anforderungserhebung in sozio-technischen Entwicklungsprojekten in einer kurzen Präsentation den Teilnehmern zusammengefasst. Zusätzlich wurde das Konzept der Software Requirement Pattern mit Ihrem Zweck am Beispiel *Failure Alerts* [FPQ10] für die Reife aus dem Bereich der Zuverlässigkeit der Softwarequalität verdeutlicht. Fragen waren zugelassen und wurden beantwortet.

Mit Hilfe der computerbasierten Groupware ThinkTank von GroupSystems wurde ein OnePage-Brainstorming [BD09] zu der Frage: „Welche Anforderungen sollen Software Requirement Pattern für die Entwicklung sozio-technischer Systeme erfüllen?“ durchgeführt. Die Zeit für das Brainstorming war nicht begrenzt. Die Teilnehmer sollten ihre Antworten direkt in die Groupware eingeben und sahen synchron die Antworten der anderen Teilnehmer. Während des Brainstormings wurden unterschiedliche Herausforderungen als Impulstreiber wiederholt. Das Brainstorming wurde beendet, als die Teilnehmer alle ihre Anforderungen eingegeben hatten. Das Ergebnis des Brainstormings waren 31 Antworten.

Die Liste der Antworten war die Grundlage für die folgende FastFocus-Aktivität [BD09]. Zudem kam eine zunächst leere Liste für die wichtigen Anforderungen. Die Teilnehmer waren aufgefordert, nacheinander eine wichtige Anforderung aus der Liste herauszusuchen, die noch nicht in der Liste der wichtigen Anforderungen vorhanden war, und diese der Gruppe laut vorzulesen. Mit diesem Schritt wurden unwichtige und redundante Anforderungen aussortiert. Die Gruppe war angehalten, die Verständlichkeit der Anforderung zu prüfen. War eine Anforderung nicht für alle eindeutig, so wurde in der Gruppe nach einer alternativen Formulierung gesucht und diese umgesetzt. War eine allgemeine Verständlichkeit gegeben, so wurde die Anforderung in die Liste der wichtigen Anforderungen übernommen. Mit diesem Schritt konnten elf wichtige Anforderungen identifiziert werden.

Während der FastFocus-Aktivität wurden alle Anforderungen übernommen, die ein Teilnehmer als wichtig erachtete. In der nachfolgenden MultiCriteria-Aktivität waren die Teilnehmer dazu aufgefordert, die verbleibenden Anforderungen bezüglich der Wichtigkeit und der Einfachheit der Umsetzung auf einer 7er-Likert-Skala zu bewerten.

In einem nächsten Schritt konnten die Teilnehmer eigene Bedenken zu den Anforderungen hinzufügen, für welche dann wiederum nach Verbesserungsvorschlägen gefragt wurden. In einer abschließenden Diskussion der Teilnehmer wurden die Bedenken und Vorschläge diskutiert. Eine Änderung einzelner Anforderung war hier möglich, wurde von der Gruppe aber einstimmig als nicht sinnvoll eingeschätzt.

## **4.2 Ableitung der Attribute für Requirement Pattern**

Als Grundlage der Struktur für Requirement Pattern für die Entwicklung sozio-technischer Systeme wurde das von Franch et al. [FPQ10] vorgeschlagene Metamodell für Requirement Pattern aus dem Bereich der Software Qualität auf die Anforderungen hin überprüft und durch fehlende Attribute ergänzt. Mit Hilfe einer abschließenden Kriterien-basierten Evaluation wurde die vorgeschlagene Struktur hin überprüft.

## 5 Ergebnisse

Das Kapitel fasst die im Workshop erhobenen Anforderungen zusammen und beschreibt danach die abgeleitete Struktur für Requirement Pattern.

### 5.1 Anforderung an Software Requirement Pattern

Das im vorherigen Kapitel beschriebene Vorgehen führte zu elf Anforderungen an Software Requirement Pattern in der Entwicklung sozio-technischer Systeme, die in Tabelle 1 zusammengefasst sind. Die Anforderungen sind nach dem von dem Teilnehmer vergebenen Durchschnittswert für die Wichtigkeit (W) sortiert. Zusätzlich ist die angenommene Einfachheit der Umsetzung (E) angegeben.

Nr.	Anforderung	E	W
1	Patterns sollten aus zwei Teilen bestehen. Ein allgemeiner Teil (z.Bsp. Goal) in einer verständlichen "Sprache" verfasst. Ein spezieller Teil für die jeweiligen Zielgruppen (Jurist/Entwickler/Designer) in Fachsprache ausformuliert (Extra Beschreibung in Fachsprache)	5,67	7,00
2	In der allgemeinen Beschreibung des Patterns die Quelle(n) angeben, von wo die Anforderung kommt	7,00	6,67
3	Verweise auf Pattern, die für dieses Pattern benötigt werden	5,17	6,33
4	Rahmenbedingungen für den Patterneinsatz (fachspezifisch) beschreiben	4,17	6,33
5	Verweise auf andere Patterns, die gleiche fachspezifische Ziele erfüllen können	5,00	6,00
6	Auflisten konfliktärer Software Requirement Pattern	3,33	5,83
7	Direkte Handlungsimplicationen sind für die jeweiligen Domänen zu kennzeichnen	5,50	5,17
8	Durchsuchbarkeit des Pattern-Kataloges	5,67	4,83
9	Ablauf- und Überprüfungsdaten für die Pattern erfassen	3,83	4,83
10	Das Pattern für die beteiligten Domänen in deren Fachsprache übersetzen	3,00	3,67
11	Beispiele für konkrete Anforderungen aus vergangenen Projekten	5,00	3,50

Tabelle 1: Anforderung an Software Requirement Pattern

Die erhobenen Anforderungen können in drei Kategorien unterschieden werden. Zum einen betreffen die Anforderungen den Aufbau der Requirement Pattern an sich (Nr. 1, 2, 4, 7, 9, 10, 11), andere Anforderungen betreffen Verknüpfungen zu anderen Requirement Pattern (Nr. 3, 5, 6) und die Anforderung 8 bezieht sich auf die Umsetzung des Kataloges. Die ersten beiden Kategorien wurden bei einer übergreifenden Struktur der Requirement Pattern für sozio-technische Systeme berücksichtigt.

## 5.2 Vorgeschlagene Struktur für Software Requirement Pattern

Aus der vorgeschlagenen Struktur für Requirement Pattern von Franch et al. [FPQ10] und den Anforderungen aus dem Expertenworkshop werden folgende Attribute für Requirement Pattern in der sozio-technischen Systementwicklung vorgeschlagen. Die Attribute sind in Metadaten und Vorlagen unterteilt. Tabelle 2 zeigt die wichtigsten Metadaten. Für die Verwendung der Requirement Pattern in einem Katalog sind zusätzliche Attribute wie Autor und Schlagwörter nützlich, letzteres kann die Suche im Katalog erleichtern (siehe Anforderung 8: Durchsuchbarkeit des Pattern-Kataloges).

Attribut	Funktion in der Struktur des Requirement Pattern
Name	Prägnanter Name des Requirement Pattern
Ziel	Gibt das Ziel an, welches das System erfüllt, wenn die Anforderung umgesetzt wurde. Das Ziel ist entsprechend der allgemeinen Pattern-Eigenschaften der Problem-Teil und dient bei der Anwendung als Entscheidungshilfe für die Auswahl relevanter Requirement Pattern für ein konkretes System.
Grundlage	Gibt die Hintergründe für das Requirement Pattern an. Dies können zum Beispiel Normen, rechtliche Vorgaben oder auch wissenschaftliche Erkenntnisse (Bsp. zur Steigerung der Akzeptanz) sein.
Rahmenbedingungen	Gibt an, für welche äußeren Bedingungen das Requirement Pattern vorgesehen ist. Das kann zum Beispiel der Geltungsraum zugrundeliegender rechtlicher Vorgaben sein.
Abhängigkeiten	Auflistung von Requirement Pattern, die zur Umsetzung der Anforderung ebenfalls erfüllt sein müssen.
Verknüpfungen	Auflistung von Requirement Pattern, die eine ähnliches Ziel verfolgen.
Konflikte	Auflistung von Requirement Pattern, deren Ziele im Konflikt zum Ziel des aktuellen Requirement Pattern stehen.

Tabelle 2: Metadaten des Requirement Pattern

Für das in den Metadaten definierte Ziel des Requirement Pattern können mehrere Vorlagen angeboten werden. Die Vorlage ist der Lösung-Teil entsprechend der allgemeinen Pattern-Eigenschaften. Die Attribute der Vorlage werden in Tabelle 3 beschrieben.

Template	Das Template entspricht einer Anforderung, die direkt in eine Anforderungsspezifikation übernommen werden kann. Sie ist so formuliert, dass sie die Erreichung des Ziels einfordert.
Erweiterung	Die Erweiterung ist eine Variante des Templates, die mit Hilfe von Parametern die Formulierung detaillierterer Anforderungen ermöglichen. Somit kann für die Anforderungsspezifikation genauere Eingrenzungen gemacht werden.
Parameter	Parameter sind variable Teile des erweiterten Template, die durch unterschiedliche Werte belegt werden können.
Werte	Werte sind Elemente, Wörter oder Phrasen, welche die Plätze der Parameter im erweiterten Template einnehmen können.
Beispiele	Konkrete Beispiele, wie das erweiterte Template in frühen Entwicklungsprojekten genutzt wurde.
Handlungs-implikationen	Wenn die Anforderung spezielle Aktivitäten von Experten notwendig machen, so werden diese in Form von Handlungsimplikationen hier angegeben.

Tabelle 3: Attribute der Vorlage innerhalb der Requirement Pattern

## 7 Diskussion

Im vorherigen Abschnitt wurden die Anforderungen an Requirement Pattern und ihre Umsetzung in einer Struktur vorgestellt. Im folgenden Abschnitt werden die Ergebnisse in Bezug auf die Forschungsfragen diskutiert. Zusätzlich werden die Limitationen beschrieben.

### 7.1 Anforderungen und Struktur der Requirement Pattern

Die Anforderung 1 verlangt, dass das Requirement Pattern aus zwei Teilen bestehen soll. Ein Teil muss allgemeinverständlich formuliert sein und der andere Teil sollte an die Spezifika der Einsatzsituation angepasst sein. Die Anforderung wird dadurch erfüllt, dass die Metadaten des Requirement Pattern unabhängig von den Domänen und Fachsprachen allgemeinverständlich verfasst werden. Für die Vorlagen als Lösungsteil der Requirement Pattern dagegen gelten keine Einschränkung hinsichtlich der Formulierungen.

Die Angaben zu der Herkunft oder der Begründung der Anforderungen, die im Requirement Pattern eingebunden ist (Anforderung 2), sind unter dem Attribut Grundlagen in den Metadaten mit aufgenommen. Für rechtliche Anforderungen können

hier zum Beispiel Gesetze, für Anforderungen der Gebrauchstauglichkeit Normen angegeben werden.

Um auf Requirement Pattern zu verweisen, die für eine erfolgreiche Umsetzung der Anforderung an ein sozio-technisches System ebenfalls in die Anforderungsspezifikation mit aufgenommen werden sollten (Anforderung 3), wurden in den Metadaten die Abhängigkeiten zu anderen Requirement Pattern mit aufgenommen. So können auch Requirement Pattern beschrieben werden, die eine Komposition aus anderen Requirement Patterns sind. Redundanzen und die doppelte Pflege ähnlicher Teile in mehreren Requirement Pattern können so verringert werden.

Um die Auswahl der Requirement Pattern zu erleichtern, sind die Rahmenbedingungen (Anforderung 4) für die Gültigkeit in den Metadaten angegeben. Wird eine Anforderung zum Beispiel aus einem gesetzlichen Vorgaben gewonnen, so kann hier angegeben werden, für welchen Rechtsraum die Vorgaben Gültigkeit besitzt.

Die Anforderung 5, dass auf Requirement Pattern mit gleichen Zielen aufmerksam gemacht werden soll, wird nicht als Verweis in der Struktur umgesetzt. Da die Requirement Pattern aus den Metadaten (inklusive Ziel) und den Vorlagen bestehen, wobei ein Requirement Pattern mehrere Vorlagen besitzen kann, werden Anforderungen zum gleichen Ziel als unterschiedliche Vorlagen einem Requirement Pattern zugeordnet. Ein entsprechendes Attribut ist nicht nötig.

Im Gegensatz zu Anforderungen mit dem gleichen Ziel ist der Verweis auf konfliktäre Requirement Pattern (Anforderung 6) notwendig. Es können sich Anforderungen unterschiedlicher Domänen oder Anforderungen einer Domäne widersprechen. So steht zum Beispiel das Gebot der Datensparsamkeit im direkten Widerspruch zur Protokollierung für Haftungsnachweise. Hier muss zwischen den Anforderungen abgewogen und entsprechend der Anwendungsziele entschieden werden. Der Hinweis auf konfliktäre Requirement Pattern in den Metadaten kann dazu beitragen, dass diese Konflikte von vornherein erkannt und beseitigt werden können.

Handlungsimplicationen für die jeweiligen Domänen (Anforderung 7) sind in die Vorlage mit aufgenommen worden. So können, wenn für die Umsetzung eines Requirement Pattern weitere Aktivitäten notwendig sind, diese mit aufgenommen werden.

Die Durchsuchbarkeit des Pattern-Kataloges (Anforderung 8) kann in der Struktur nicht sichergestellt werden. Durch die Aufnahme von Keywords in die Metadaten kann aber eine spätere Indizierung vorbereitet werden.

Das Ziel von Requirement Pattern ist eine zeitlich unbegrenzte Nutzung in Entwicklungsprojekten. Ablauf und Überprüfungsdaten (Anforderung 9) widersprechen dieser Idee, sind aber bei genauerem Hinsehen notwendig. Werden Requirement Pattern zum Beispiel auf Grundlage von gesetzlichen Vorgaben geschaffen, müssen diese von Zeit zu Zeit dahingehend überprüft werden, ob die rechtliche Vorgabe weiter Bestand hat. Ein solches Datum sollte jedoch vorsichtig genutzt werden. Zeitunabhängige

Requirement Pattern, die zum Beispiel anstatt der Rechtsmäßigkeit die Rechtsverträglichkeit adressieren [HSH12], sind vorzuziehen.

Die Übersetzung der Requirement Pattern in die Fachsprachen der jeweiligen Domänen (Anforderung 10) wurde in der Struktur nicht umgesetzt. Die allgemeinverständlichen Attribute der Metadaten sollen die Funktion der breiten Verständlichkeit übernehmen. Die Vorlagen fachspezifisch für weitere Domänen aufzuarbeiten wird vorerst nicht verfolgt.

Beispiele für angewandte Requirement Pattern (Anforderung 11) sind vor allem für die erweiterten Templates der Vorlagen sinnvoll. So kann die Anwendung der Templates verdeutlicht werden.

## **7.2 Limitationen**

Die Anforderungen an die Requirement Pattern wurden mit sechs Experten aus dem Bereich der Entwicklung sozio-technischer Systeme in einem Workshop gesammelt. Die resultierenden Anforderungen spiegeln somit die Meinungen der Teilnehmer wieder und betrachten die Sicht der Entwicklerseite. Als Grundlage für die Struktur der Requirement Pattern wurde die von Franch et al. [FPQ10] vorgeschlagene Struktur genutzt. Somit werden die grundlegenden Anforderungen an die Requirement Pattern weiterhin erfüllt, die durch Anforderungen speziell für den Einsatz in der sozio-technischen Entwicklung erweitert wurden. Auf spezielle Anforderungen von Domänenexperten aus den Bereichen Recht, Vertrauen und Gebrauchstauglichkeit wurde in diesem Schritt verzichtet, um eine übergreifende Struktur zu definieren. Ob die Domäne Einfluss auf die Struktur der Requirement Pattern hat, soll in einem der nächsten Schritte untersucht werden.

Während der Diskussionen im Workshop zur Erhebung der Anforderungen an Requirement Pattern stellte sich heraus, dass es unterschiedliche Verständnisse von Anforderungen, Software Requirement Pattern und Design Pattern gab und diese teilweise als identisch angesehen wurden. Beim Erkennen von unterschiedlichen Auffassungen wurde sofort versucht, mit Hilfe von Erklärungen ein gemeinsames Verständnis zu schaffen.

## **8 Zusammenfassung**

Anforderungsanalysten können bei der Spezifikation von sozio-technischen Systemen Requirement Pattern zur Erhebung und Analyse von Anforderungen verschiedener Stakeholder nutzen. In diesem Beitrag wurden in einem Expertenworkshop Anforderungen an solche Requirement Pattern gesammelt. Die wichtigen Anforderungen dabei waren zum Beispiel, dass das Requirement Pattern aus einem allgemein verständlichen und einem fachspezifischen Teil bestehen sollten, dass die Grundlagen der Anforderungen und die Rahmenbedingungen des Pattern-Einsatzes vorhanden sind, sowie Verweise zu konfliktären und abhängigen Requirement Pattern. Die von Franch et

al. [FPQ10] vorgeschlagene Struktur für Requirement Pattern aus dem Bereich Software Qualität wurde hinsichtlich der erhobenen Anforderungen angepasst und ergänzt. Entsprechend der Struktur erstellte Requirement Pattern bieten die Möglichkeit, das Verständnis der Anforderungen bei den Stakeholdern zu erhöhen und somit Interessenskonflikte aufzudecken und durch bereitgestellte Vorschläge aufzulösen.

In zukünftigen Studien soll die Struktur mit Hilfe weiterer Experten weiter angepasst und für den praktischen Einsatz verbessert werden. Dazu wird ein Katalog mit Requirement Pattern für die Entwicklung sozio-technischer Systeme erstellt und in Fallstudien eingesetzt.

## Literaturverzeichnis

- [AI79] Alexander, C.: The timeless way of building (Bd. 1), Oxford University Press, USA, 1979.
- [BN98] Bader, G., and Nyce, J.M.: When only the self is real: theory and practice in the development community. SIGDOC Asterisk J. Comput. Doc. 22, 1 (1998), 5-10.
- [BS11] Baxter, G., and Sommerville, I.: Socio-technical systems: From design methods to systems engineering. Interacting with Computers 23, 1 (2011), 4-17.
- [BJH12] Behrenbruch, K., Jandt, S., Hoberg, S., Roßnagel, A., and Schmidt, L.: Normative Anforderungsanalyse für ein RFID-basiertes Assistenzsystem für Arbeitsgruppen. Proc. GfA-Frühjahrskongress(2012).
- [BD09] Briggs, R.O., and De Vreede, G.-J.: ThinkLets - building blocks for concerted collaboration, Univ. of Nebraska, Center for Collaboration Science, Omaha, Neb., 2009.
- [CEG12] Comes, D., Evers, C., Geihs, K., Hoffmann, A., Kniewel, R., Leimeister, J.M., Niemczyk, S., Roßnagel, A., Schmidt, L., Schulz, T., Söllner, M., and Witsch, A. Designing Socio-Technical Applications for Ubiquitous Computing - Results from a Multidisciplinary Case Study. Proc. 12th IFIP International Conference on Distributed Applications and Interoperable Systems(2012).
- [FPQ10] Franch, X., Palomares, C., Quer, C., Renault, S., and De Lazzer, F.: A Metamodel for Software Requirement Patterns. Requirements Engineering: Foundation for Software Quality (2010), 85-90.
- [HC07] Henninger, S., and Corrêa, V.: Software pattern communities: Current practices and challenges. Proc., ACM (2007), 14.
- [HL11] Hoffmann, A., and Leimeister, J.M. Fachwissen nutzen – Kombination von Anforderungen verschiedener Disziplinen bei der Entwicklung ubiquitärer Anwendungen. Proc. 9. Berliner Werkstatt Mensch-Maschine-Systeme (BWMMS), VDI Verlag (2011).
- [HSH12] Hoffmann, A., Schulz, T., Hoffmann, H., Jandt, S., Roßnagel, A., and Leimeister, J.M.: Towards the Use of Software Requirement Patterns for Legal Requirements. Proc. 2nd International Requirements Engineering Efficiency Workshop (REEW 2012) at REFSQ 2012(2012).
- [HSF11] Hoffmann, A., Söllner, M., Fehr, A., Hoffmann, H., and Leimeister, J.M.: Towards an Approach for Developing socio-technical Ubiquitous Computing Applications. Proc. Sozio-technisches Systemdesign im Zeitalter des Ubiquitous Computing (SUBICO 2011) im Rahmen der Informatik 2011 - Informatik schafft Communities, Bonner Köllen Verlag (2011).

- [LSK06] Leimeister, J.M., Sidiras, P., and Krcmar, H.: Exploring Success Factors of Virtual Communities: The Perspectives of Members and Operators. *Journal of Organizational Computing & Electronic Commerce* 16, 3/4 (2006), 279-300.
- [RM09a] Renault, S., Mendez-Bonilla, O., Franch, X., and Quer, C.: PABRE: Pattern-based Requirements Elicitation. *Proc. Research Challenges in Information Science, 2009. RCIS 2009. Third International Conference on(2009)*, 81-92.
- [RM09b] Renault, S., Mendez-Bonilla, O., Franch, X., and Quer, C.: A Pattern-based Method for building Requirements Documents in Call-for-tender Processes. *International Journal of Computer Science and Applications* 6, 5 (2009), 175 - 202.
- [RR06] Robertson, S., and Robertson, J.: *Mastering the requirements process*, Addison-Wesley Professional, 2006.
- [So11] Sommerville, I.: *Software engineering*, Pearson, 2011.
- [Wi08] Withall, S.: *Software Requirements Patterns*, Microsoft Press, Redmont, Washington, 2008.