

# Musterbasierte Fragenchecklisten zur Ableitung von Anforderungen aus modellierten Geschäftsprozessen

Bernhard M. Turban<sup>1</sup> und Johannes Schmitz-Lenders<sup>2</sup>

**Abstract:** Bei der Umsetzung geplanter Geschäftsprozesse auf IT-Systeme (IT-Alignment) kommt der Anforderungserhebung eine zentrale Bedeutung zu. Dabei ist das Erfahrungswissen der Anforderungsingenieure, die entscheidenden Fragen zur möglichst frühzeitigen Aufdeckung noch unbekannter Anforderungen zu stellen, sehr entscheidend. Dieser Artikel stellt eine Möglichkeit vor, wie diese Art von Erfahrungswissen bzgl. der Ableitung von Anforderungen aus modellierten Geschäftsprozessen in Form einer musterbasierten Fragencheckliste aufgezeichnet und unerfahrenen Anforderungsingenieuren als Hilfsmittel zur Aufdeckung von Anforderungen bereitgestellt werden kann.

**Keywords:** BPMN, Geschäftsprozessmodellierung, Fragenkatalog, Checkliste, Muster, Pattern, Anforderungserhebung, Requirement Elicitation, IT Business Alignment

## 1 Einleitung

Beim Entwurf neuer Geschäftsprozesse (GPs) oder der Umgestaltung vorhandener GPs müssen zumeist IT-Systeme angepasst werden. Bei der Umsetzung der einzelnen Aufgaben eines GPs ist es wichtig, möglichst alle Anforderungen für alle beteiligten IT-Systeme aufzudecken. Diese Tätigkeit stellt sich i.d.R. als sehr komplex heraus, deren Ausgang oft sehr stark vom jeweiligen Erfahrungswissen der beteiligten Anforderungsingenieure abhängt.

Geschäftsprozesse werden oft in einer Modellierungssprache wie BPMN [OMG11] als GP-Modell modelliert. Dieser Artikel zeigt, wie Erkenntnisse der Mustertheorie genutzt werden können, um Erfahrungswissen von Anforderungsingenieuren bzgl. der Transformation von GP-Modellen in Anforderungen an IT-Systeme in einer *musterbasierten Fragencheckliste* festzuhalten, die es ermöglicht, das Erfahrungswissen an weniger erfahrene Anforderungsingenieure weiterzugeben, um deren Effektivität zu verbessern.

In Kap.2 werden zunächst die Grundideen hinter dem vorgeschlagenen Ansatz genauer erläutert. Kap.3 schildert den Ansatz, indem zunächst der allgemeine Aufbau einer solchen *musterbasierten Fragencheckliste* erklärt und anschließend ein Auszug aus einer nach diesem Prinzip erarbeiteten *musterbasierten Fragencheckliste* präsentiert wird. Kap. 4 fasst dann die Möglichkeiten und Potentiale des Ansatzes zusammen.

---

<sup>1</sup> Hochschule RheinMain, DCSM WI, Unter den Eichen 5, 65195 Wiesbaden, bernhard.turban@hs-rm.de  
<sup>2</sup> parcs IT-Consulting GmbH, Spandauer Str. 32, 57072 Siegen, jsl@parcs.de

## 2 Die Grundideen hinter dem hier beschriebenen Ansatz

Im Gegensatz zu der automatisierten Umsetzung von modellierten GPs über Workflow-Engines existieren zu dem oben beschriebenen Szenario einer manuellen Transformation von GP-Modellen in Anforderungen wenige Hinweise in der Literatur, wie hier vorgegangen werden soll<sup>3</sup>. Vielmehr entscheidet hier die Erfahrung der beteiligten Anforderungsingenieure, wie hier ein nahtloser Übergang zu den Anforderungen für die einzelnen IT-Systeme erreicht werden kann. Hierzu ist zu erwähnen, dass es zu diesem Problem keine einfache, allgemeine Lösungsbeschreibung geben kann, da es sich um einen komplexen, diffus-situations- und erfahrungsabhängigen Prozess handelt.

Die Mustertheorie<sup>4</sup> (Patterns) bietet einen alternativen Ansatz, derart komplexe, erfahrungsabhängige Probleme zu adressieren. Neben den Software Design Patterns (vgl. z.B. [Ga95]) haben sich auch Anforderungs- [Wi07] und Analysemuster [Fo97] etabliert. Muster versuchen, komplexe wiederkehrende Situationen so zu beschreiben, dass sie bei ähnlichen Aufgabenstellungen einfach angewendet werden können. Muster kodieren somit Erfahrungs- bzw. Expertenwissen [Mo04; S.139], [Tu13; S.101], das auf diese Art leichter an unerfahrenere Personen weitergegeben werden kann<sup>5</sup>, und Experten können Muster nutzen, um leichter Ideen auszutauschen [PU99], [Tu13; S.101]. Genau dieser Effekt wird sich bei dem hier vorgestellten Ansatz zu Nutze gemacht.

Bei der Anforderungsermittlung besteht ein erfolgskritischer Aspekt darin, frühzeitig möglichst alle relevanten Anforderungen aufzudecken bzw. möglichst wenige Anforderungen zu übersehen. Hierbei hilft eine systematische Suche, weshalb in der Literatur z.B. empfohlen wird, Anforderungen mittels Checklisten systematisch zu ermitteln<sup>6</sup>.

Im folgenden Kapitel wird gezeigt, wie die hier erarbeiteten Überlegungen genutzt werden können, um eine *musterbasierte Fragencheckliste* zu entwickeln, die verschiedene Modellierungssituationen in modellierten GPs beschreibt, um dann zu diesen Modell-situationen eine Fragencheckliste zur Aufdeckung von Anforderungen<sup>7</sup> bereitzustellen.

<sup>3</sup> [FR17], [Ro11] weisen zwar darauf hin, dass bei einer GP-zentrierten Anforderungsanalyse für die Implementierung –insbesondere ohne Workflow-Engine– GPs alleine nicht ausreichend und zusätzliche „klassische“ Anforderungen erforderlich sind, und benennen auch die üblichen dafür verwendbaren Elemente (z.B. Geschäftsobjekte/Klassendiagramme, Use Cases, Geschäftsregeln, Benutzerdialoge), lassen aber weitestgehend offen, wie diese Anforderungen auf Basis und im Einklang mit GPs ermittelt werden können.

<sup>4</sup> [Tu13; Kap. I.6.2.4] bietet ein relativ umfassende Beschreibung sowohl der Kernideen hinter dem Musterge-danken als auch einen Überblick über vielfältigen Arten von Mustern, deren Wirkungen, Nachteile und den Bezug zu anderen designtheoretischen Auffassungen in der Informatik.

<sup>5</sup> Prechelt und Unger [PU99] zeigen z.B., dass unerfahrene Softwareentwickler die Qualität Ihres Softwaredesigns durch die Anwendung von Mustern signifikant verbessern konnten.

<sup>6</sup> Rupp et al. schlagen dies z.B. für die Ermittlung von Stakeholdern [RS14; S.79] oder nichtfunktionalen An-forderungen [RS14; S.270ff] oder generell zum Prüfen von Anforderungen [RS14; S.329] vor.

<sup>7</sup> Wie Maiden et al. [Ma05; S.286] darstellen, fällt es Stakeholdern leichter, Fehler bei erst einmal vorhandenen Anforderungen als generell das Fehlen von Anforderungen aufzudecken. Deshalb ist es bei der Konzipierung der Fragen in der Fragencheckliste selbst nicht so entscheidend, ob damit sofort die korrekten Anforderungen aufgedeckt werden können. Stattdessen sollte hier das Augenmerk vielmehr auf das Potential zur Aufdeckung noch tatsächlich fehlender Anforderungen gelegt werden.

### 3 Prinzipdarstellung einer *musterbasierten Fragencheckliste*

In diesem Kapitel wird das Prinzip einer möglichen *musterbasierten Fragencheckliste* vorgestellt. Die Fragen werden in eine Beschreibungsstruktur eingebettet, die sich an typischen Strukturen von Mustern (Musterkatalog und -schablone) orientiert. Die gesamte Beschreibungsstruktur (äquivalent zu einem Musterkatalog) ist in einzelne Kapitel unterteilt. Jedes Kapitel beschreibt musterartig eine bestimmte modellierte Situation (äquivalent zu einem einzelnen Muster) mit den dazugehörigen Fragen. Jedes einzelne Kapitel folgt hierbei einer klar definierten Struktur (äquivalent zur Musterschablone):

- Die Überschrift besteht aus einer allgemeinen –leicht merkbaren– Regel, als eine umfassende Themenklammer für die sie enthaltenden Fragen. Hierbei handelt es sich immer um die adressierte musterartig beschriebene Modellierungssituation. Um Verweise zwischen Regeln zu erleichtern, steht vor jeder Regel eine Regel-Id (R...).
- Im Abschnitt unter der Überschrift werden zunächst die adressierte Modellierungssituation und der Bezug zu den folgenden Fragen erläutert.
- Anschließend werden die Fragen aufgelistet. Den Fragen ist zunächst eine Fragen-Id (Rx.F...) vorangestellt, um auch hier das Verweisen an anderer Stelle zu erleichtern. Dem folgt dann die Frage. Falls die Frage einer weiteren Erläuterung bedarf, wird anschließend –über einen Bindestrich getrennt– eine weitere Erläuterung geboten.

Im nächsten Kapitel folgt ein Auszug aus einer umfangreicheren von den Autoren erarbeiteten Beispielfragencheckliste. Diese Fragencheckliste konzentriert sich auf typische Modellierungssituationen, die bei der GP-Modellierung in BPMN auftreten können.

#### 3.1 R1: Fast jede IT-bezogene Aufgabe benötigt mind. ein Interaktionselement

R1 beschreibt die Tatsache, dass fast jede IT-bezogene Aufgabe in einem GP-Modell mindestens ein Mal mit einer Benutzerinteraktion verbunden ist und hierfür –zumeist– mindestens ein Nutzerinteraktionselement<sup>8</sup> (NIAE) in dem betreffenden System vorhanden sein muss<sup>9</sup>. Deshalb muss für jede IT-Aufgabe folgendes geklärt werden:

- R1.F1: Welche Informationen werden hierfür benötigt? – Es ist zu klären<sup>10</sup>, welche Informationen notwendig und welche evtl. noch nützlich für die Aufgabe wären.
- R1.F2: Wie sollen die benötigten Informationen dargestellt und durch den Benutzer

---

<sup>8</sup> Z.B. ein GUI-Fenster (GUI=Graphical User Interface), Teil eines GUI-Fensters, ein GUI-Control, oder sonst ein Konzept zur Nutzerinteraktion (z.B. Sprachkommandos, Steuergesten, ...).

<sup>9</sup> Vgl. hierzu auch die Bemerkungen von [FR17; S.156f] zur Modellierung von Maskenflüssen und BPMN.

<sup>10</sup> Soweit für den für den Zweck der Anforderungserhebung angestrebten Detailgrad erforderlich; dies gilt auch im Folgenden für alle anderen Fragen des Fragenkatalogs;

editiert werden können? – Für jede Information muss geklärt werden, mit welchen NIAEs (z.B. GUI-, „Controls“) die entsprechende Information dargestellt und evtl. die Eingabe von Daten gehandhabt werden soll.

- R1.F3: Wie viele NIAEs sind involviert<sup>11</sup>? – Gibt es keine, eine oder mehrere NIAEs. Die folgenden Fragen müssen für jedes NIAE beantwortet werden.
- R1.F4: Ist ein eigenes (neues) NIAE erforderlich?
- R1.F5: Kann ein anderes schon vorhandenes NIAE (z.B. GUI-, „Maske“) erweitert werden? – Diese Frage ist eine Art Gegenfrage zu R1.F4. Hier schließen sich folgende Unterfragen an, die v.a. den Aspekt der Gebrauchstauglichkeit adressieren:
  - R1.F5.1: Passen die Daten des NIAEs dann überhaupt zusammen?
  - R1.F5.2: Passen die verschiedenen Aufgaben des NIAEs zusammen?
  - R1.F5.3: Wird das NIAE dadurch nicht überladen?
- R1.F6: Wie gelangt der Benutzer zum NIAE? – Durch Suche, Filtern, ...?
- R1.F7: Ist die Navigation bzgl. der anderen Aktivitäten schlüssig? – Fast jede Aktivität hat eine oder mehrere Vor- und Nachaktivitäten. Hier ein schlüssiger Übergang zwischen den Aktivitäten und den damit assoziierten NIAEs erforderlich.

Diese Fragen müssen für alle in einem GP modellierten Aufgaben gestellt und beantwortet werden, weil dies essentielle Informationen zu jeder Aufgabe sind.

### 3.2 Derzeit bekannte Regeln in der ausgearbeiteten Beispielfragencheckliste

Derzeit haben die Autoren im erarbeiteten Beispielfragenkatalog folgende weitere Regeln identifiziert und dazu jeweils Musterkapitel entsprechend obiger Struktur erarbeitet:

- „R2: Fast jedes Datenartefakt stellt zumeist ein eigenes Artefakt oder sonst etwas Wesentliches dar“ – adressiert Datenartefakte und diesbzgl. typische Fragen;
- „R3: Start-/Endereignisse müssen genau analysiert werden“ – adressiert typische Fragestellungen bzgl. Start-/Endereignissen sowie den Bezügen dazwischen;

---

<sup>11</sup> Diese und folgende Fragen adressieren die Anforderungen an die logische Struktur / den logischen Zuschnitt von NIAEs. Die konkrete Ausgestaltung hängt vom angestrebten Detaillierungsgrad– und den geforderten / durch Rahmenbedingungen vorgegebenen Technologien ab, z.B. Web/Browser, Desktop, Mobilgeräte, ...

- „R4: Fehler- und Abbruchkanten/-ereignisse müssen genau analysiert werden“ – adressiert typische Fragestellungen in Bezug auf Fehler oder Nutzerabbrüche, wie etwa Konsistenzerhaltung der Daten
- „R5: Systemgrenzen überschreitende Pfeile sind besonders kritisch“ – adressiert Medienbrüche, die typischerweise beim Übergang zwischen Systemen auftreten.
- „R6: Auch Benutzergrenzen überschreitende Pfeile sind sehr wichtig“ – adressiert mögliche Brüche bei Übergängen von GP-Verantwortungsbereichen.

## 4 Möglichkeiten und Potentiale des Ansatzes

Dieser Artikel zeigt, wie der Mustergedanke verwendet werden kann, um Erfahrungswissen von Anforderungsingenieuren bzgl. der Übertragung von modellierten Geschäftsprozessen auf Anforderungen für IT-Systeme festzuhalten.

Da es sich bei modellierten GPs um sehr abstrakte Beschreibungen handelt und eine Übertragung in IT-Systeme in Hinblick auf Rahmenbedingungen und sonst zu beachtende Aspekte sehr unterschiedlich geartet sein kann, ist es für diese Fälle schwierig, allgemeingültige, rezepthafte Regeln für die Ableitung von Anforderungen aufzustellen. Deshalb schlagen wir einen Ansatz vor, der eher musterartig typische Modellierungssituationen beschreibt und dazu einen checklistenartigen Fragenkatalog bietet, der bei der frühzeitigen Aufdeckung möglichst aller relevanten Anforderungen helfen soll. Eine solche *musterbasierte Fragencheckliste* ermöglicht es dann, modellierte GPs systematisch nach beschriebenen Modellierungssituationen abzugehen und mittels der jeweils gesammelten Fragen möglichst alle relevanten Anforderungen zu erfragen bzw. erstellte Anforderungen bzgl. ihrer Qualität und Vollständigkeit zu prüfen.

Eine solche *musterbasierte Fragencheckliste* kann auch benutzt werden, um Ungenauigkeiten und Inkonsistenzen innerhalb eines GP-Modells aufzudecken (auch unter Berücksichtigung von etwaigen Modellierungskonventionen), getroffene Entscheidungen bzgl. Systemgrenzen zu hinterfragen (z.B. um Datenredundanzen zu reduzieren), oder bei einer automatisierten Umsetzung via Workflow-Engine helfen, wichtige Aspekte bzw. Anforderungen aufzudecken.

Weitere Vorteile eines solchen Vorgehens sind, dass eher ein ganzheitliches Denken über einzelne Systemgrenzen hinweg gefördert wird und dieses Wissen in ein vermittelbares Format gebracht werden kann.

Zur Validierung des hier vorgestellten Prinzips haben die Autoren eine Fragencheckliste für BPMN-Modelle entwickelt und diese in Kap 3.1 und 3.2 auszugsweise vorgestellt. Diese Fragencheckliste adressiert noch nicht alle in BPMN vorkommende Situationen, weshalb die Autoren die Analyse weiterer Situationen planen.

Ein anknüpfender Aspekt ist die Frage nach einer geeigneten Dokumentationsstruktur für die aufgedeckten Anforderungen. Während GP-Modelle eher eine fachliche Prozesssicht adressieren, ist bzgl. der Umsetzung eher eine Strukturierung anhand der IT-Systeme erforderlich, wo sich eher eine anwendungsfallorientierte Sichtweise [Ja04] etabliert hat, weshalb eine Verknüpfung dieser beiden Sichtweisen notwendig ist (vgl. z.B. auch [FR17; S.158f], [RS14; S.170f]). Dies wiederum wirft die Frage auf, ob ein ähnliches, muster-basiertes Konzept auch diesbzgl. Hilfestellung leisten kann.

## Literaturverzeichnis

- [Fo97] Fowler, M.: Analysis Patterns: Reusable Object Models. Addison-Wesley, Reading, MA, 1997.
- [FR17] Freund, J.; Rücker, B.: Praxishandbuch BPMN. Hanser Verlag, München, 2017.
- [Ga95] Gamma, E. et al.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA, 1995.
- [Ja04] Jacobson, I.: Object Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, Red Wood City, CA, USA, 2004.
- [Ma05] Maiden, N. et al.: Generating requirements from systems models using patterns: a case study. In: Requirements Engineering November 2005, Vol. 10, Issue 4, pp. 276–288, Springer (2005).
- [Mo04] Moro, M.: Modellbasierte Qualitätsbewertung von Softwaresystemen. Books on Demand GmbH, 2004.
- [OMG11] Object Management Group (OMG): Business Process Model and Notation (BPMN) V. 2.0, 2011 (<http://www.omg.org/spec/BPMN/2.0> (Zugriff: 06.17)).
- [PU99] Prechelt, L.; Unger, B.: Methodik und Ergebnisse einer Experimentreihe über Entwurfsmuster. In: Informatik Spektrum 14 Nr. 3, März 1999.
- [Ro11] Rohfleisch, F.: Geschäftsprozessorientierte Anforderungsanalyse – Business Analyse mit ARIS und UML. Edition Octopus, MV-Verlag, Münster, 2011.
- [RS14] Rupp, Chr.; Sophist Group: Requirements-Engineering und Management, 6. Auflage, Hanser Verlag, München, 2014.
- [Tu13] Turban, B.: Tool-Based Requirement Traceability between Requirement and Design Artifacts. Vieweg+Teubner Verlag, Wiesbaden, 2013.
- [Wi07] Withall, S.: Software Requirement Patterns. Microsoft Press, WA, Redmond, 2007.