Chris Rupp, Rainer Joppich

10

# Anforderungsschablonen – der MASTER-Plan für gute Anforderungen


- Zahlreiche Regeln für jede Anforderung zu überprüfen, kostet enorm viel Zeit. Nutzen Sie deshalb die Anforderungsschablonen, um Anforderungen zu dokumentieren.
- Der Umgang mit den Anforderungsschablonen ist leicht erlernbar. Sie erhöhen damit die Qualität Ihrer Anforderungen bereits beim ersten Aufschreiben.
- Definieren Sie Prozesswörter, Begriffe und alle weiteren Bestandteile Ihrer Anforderungen verbindlich, um zu gewährleisten, dass alle Projektbeteiligten die gleiche Sprache sprechen.

*Ramona, Ralph, Herr Büchle und Katharina haben sich dazu entschlossen, für ihre Anforderungsdokumentation sowohl Modelle als auch natürliche Sprache zu verwenden.*

*Jetzt geht es daran, die einzelnen Anforderungen aus den eher unstrukturierten Interviewprotokollen, Flipchartblättern und sonstigen Ergebnisartefakten der Workshops herauszufiltern und zu dokumentieren. „Wie schreibt man das alles denn nun strukturiert und möglichst vollständig auf?" fragt Herr Büchle. Da Ramona und Ralph den schablonenbasierten Ansatz zur Dokumentation von natürlichsprachlichen Anforderungen kennen und schätzen gelernt haben, wollen sie diesen Ansatz auch in diesem Projekt verwenden. Sie erklären Herrn Büchle und Katharina die Schablonen und üben gleich das Formulieren der ersten Anforderungen mit den beiden.*

## 10.1 Linguistische und philosophische Grundlagen

Anforderungen müssen oder sollen oft in natürlicher Sprache verfasst werden. Die Anforderung in Textform ist nach wie vor sehr verbreitet und die Regeln des SOPHIST-*RE*gelwerks, das Sie in Kapitel 7 „Das SOPHIST-*RE*gelwerk" kennengelernt haben, können bei deren Formulierung sehr gute Dienste leisten.

Dabei hilft die priorisierte Abarbeitung der Regeln wie in Kapitel 7 „Das SOPHIST-*RE*gelwerk" gezeigt.

Egal, ob Sie Anforderungen direkt bei der Ermittlung mit schreiben oder erst im nachhinein genauer analysieren und aufschreiben wollen, das SOPHIST-*RE*gelwerk ist sehr gut geeignet, um eine sehr hohe Qualität von Anforderungen zu erhalten.

Der Aufwand jedoch, jede Anforderung auf eine gewisse Anzahl von Regeln zu untersuchen, ist hoch und kann in der realen Projektsituation oftmals nicht geleistet werden. Auch enden Sie bei stilistisch sehr unterschiedlichen Anforderungen, was der Formalität der Inhalte einer Spezifikation widerspricht. Die folgenden drei Anforderungen aus unserem Bibliotheksprojekt zeigen dies:

Anforderung des Bürgermeisters als Vertreter des Stadtrats

Bei jeder Registrierung wird die Kontoverbindung des Kunden festgehalten.

Das will der Chefbibliothekar Herr Büchle.

Falls der Kunde nicht bereits registriert ist, soll der Bibliothekar seine persönlichen Daten eintragen können und der Kunde bekommt anschließend vom System eine eindeutige Nummer zugewiesen.

Und dies ist die Meinung der neuen Bibliothekarin Katharina Feger.

Falls ein nicht registrierter Ausleiher Bücher ausleihen möchte, muss er erst seinen Namen, seine Adresse und sein Geburtsdatum über den Touchscreen eingeben und damit ein Benutzerkonto anlegen.

Übrigens: Ramona und Ralf hatten bei der Ermittlung der Anforderungen in den Workshops und Interviews Herrn Büchle, Frau Feger und den Bürgermeister nach dem gleichen Sachverhalt gefragt – der Registrierung eines Kunden in der Bibliothek – und erhalten drei unterschiedliche Antworten. Wie sollte es auch anders sein! Die drei Befragten haben eben ganz unterschiedliche Ansichten und Erfahrungen.

Die Antworten unterscheiden sich nicht nur in ihrer Satzstellung, Wortwahl und den verwendeten Begriffen, sondern auch in inhaltlicher Hinsicht. Es gibt vermutlich keine Stakeholdergruppe, die auf Anhieb denselben Prozess mit all seinen Aspekten vollständig nennen kann.





Zudem werden verwandte, aber nicht zwangsläufig semantisch identische Wörter verwendet, wie zum Beispiel Kunde/Ausleiher oder zuweisen/abbilden.

RE-Bauernregel: Ist's am Sonntag nass und grau, nimm dies Buch und mach dich schlau.

Letztendlich lebt jeder Stakeholder in seiner eigenen Wirklichkeit und die Frage ist, ob es überhaupt möglich ist, dass alle Beteiligten nach und nach ein gemeinsames, einheitliches Verständnis von dem zu erstellenden System entwickeln können.

Diese Problematik ist seit langer Zeit Gegenstand intensiver wissenschaftlicher Forschung. Angefangen bei den Sophisten des antiken Griechenlands über Ludwig Wittgenstein, Gottlob Frege bis hin zu Noam Chomsky und John Searle sowie vielen anderen modernen Linguisten und Philosophen. Sie alle beschäftigen sich mit Fragen der Art:

- Wie „funktioniert" Sprache?
- Was ist die innerste Natur von Sprache?
- Wie lässt sich Sprache formal beschreiben?
- Wie lernen Menschen Sprache?
- Können sich Menschen durch Sprache überhaupt verständigen?

Die beruhigende Antwort auf Letzteres ist: grundsätzlich JA.

Unser schablonenbasierter Ansatz baut auf diesen linguistischen und philosophischen Grundlagen auf, deren ausführlichere Erläuterung jedoch den Umfang des Buchs sprengen würde. Deshalb halten wir für alle Grundlagenforscher, Linguistik- und Philosophie-Fans einige Artikel auf www.sophist.de/re6/kapitel10 bereit.

„Besser" heißt: Die Anforderung entspricht bereits gewissen Qualitätskriterien, die bei den „schlechteren" Anforderungen noch nicht erreicht sind.

Im Folgenden werden wir uns vor allem mit der Tatsache beschäftigen, dass es Anforderungen gibt, die „besser" sind als andere. Daraus ergibt sich die Frage:

Wie können wir DIREKT zur „besten" Formulierung gelangen?

Die Schablonen sind unsere Antwort darauf.

## 10.2 Der schablonenbasierte Ansatz

Um Ihre Aufwände beim Ermitteln und Schreiben von Anforderungen deutlich zu minimieren, zeigen wir Ihnen, wie Sie mit erstaunlich einfachen Mitteln Anforderungen hoher Qualität in einem optimalen Zeit- und Kostenrahmen verfassen können. Dafür geben wir Ihnen Schablonen an die Hand, die Ihnen bei der Konstruktion und bei der Qualitätssicherung von Anforderungen behilflich sind. Sie erzeugen damit von Grund auf qualitativ hochwertige Anforderungen – sichern die Qualität Ihrer Anforderungen also konstruktiv. Mehr zur konstruktiven Prüfung in Kapitel 14 „Prüftechniken für Anforderungen".

D. h., dass manche sprachliche Effekte von vornherein NICHT in Ihren Anforderungen auftauchen.

Wir setzen bei der Frage an, wie eine optimale Anforderung syntaktisch aussehen soll. Damit wird von vornherein ein erheblicher Anteil typischer Formulierungsfehler ausgeschlossen, wie zum Beispiel Passivsätze, die meist keine Auskunft darüber geben, wer die spezifizierte Funktionalität erwartet. Die Konstruktion der Anforderungen nach den Anforderungsschablonen ermöglicht es, jeder Anforderung eine ähnliche Struktur aufzuprägen. Ähnlich, wie ein Haus nach dem Plan eines Architekten gebaut wird, kann jede Anforderung nach

einem Plan – besser nach einer Schablone (engl. Template) – zusammengesetzt werden. Wir sprechen von syntaktischen Anforderungsschablonen, da hier allein die Syntax (Struktur) einer Anforderung festgelegt wird, nicht aber ihre Semantik (Inhalt) – dazu mehr in Abschnitt 10.4 „Semantische Präzisierung der Anforderungsschablone“ in diesem Kapitel.

Eine Anforderungsschablone ist ein Bauplan, der die Struktur eines einzelnen Anforderungssatzes festlegt.

Als Grundannahme für alle Anforderungsschablonen in diesem Kapitel gilt: Sie haben Vorschlagscharakter. Nehmen Sie sie als gut durchdachte Idee, um zu guten Anforderungen zu gelangen, aber zwängen Sie nicht jeden Satz unbedingt hinein. Das heißt auch, dass Sie die Schablonen gerne verändern dürfen. Sie sind nicht in Stein gemeißelt, sondern sollen Ihnen einen möglichen Ansatz zur Dokumentation geben. Sie haben sich in der Praxis bewährt und werden von den meisten Stakeholdern mit Begeisterung angenommen – da sie leicht zu erlernen und einfach zu verstehen sind.

Unwilligen Stakeholdern stellen Sie die Schablonen am besten erst gar nicht vor, sondern überrumpeln diese Personen hinterrücks, indem Sie beispielsweise in Reviewsitzungen Anforderungen nach und nach anhand der Schablonen umformulieren.

Eine Art von „Einführungsstrategie“ für die Schablonen

Wir bieten Ihnen eine Vielzahl unterschiedlicher Schablonen für natürlichsprachliche Anforderungen an. Von Schablonen für funktionale Anforderungen über Schablonen für nicht-funktionale Anforderungen unterschiedlicher Kategorien bis hin zu Schablonen für Bedingungssätze. Die gesamte Familie aller Schablonen bezeichnen wir mit dem Begriff MASTER. Jede einzelne Schablone erhält zudem einen eigenen und eindeutigen Namen, so dass Sie immer genau wissen, auf welche Schablone wir uns bei Erklärungen beziehen. Um Ihnen eine Übersicht über die in diesem Kapitel vorgestellten Schablonen zu geben, haben wir dies in Abbildung 10.1 visualisiert.

Langform: Mustergültige Anforderungen – die SOPHIST-Templates für Requirements





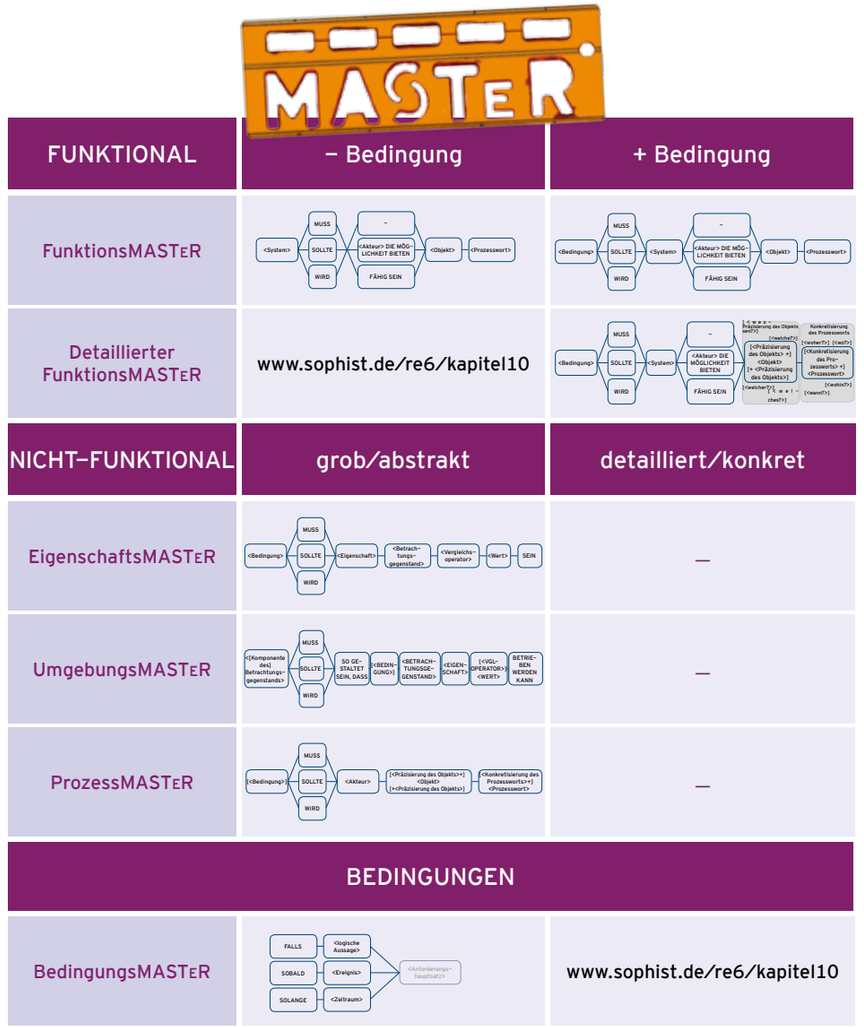



**Abbildung 10.1** Die Anforderungsschablonen in der Übersicht

## 10.3 Schritt für Schritt zur Anforderung

RE-Bauernregel: Sind deine Anforderungen nicht ganz knusper, nimm doch einfach diese Muster.

Genug Spannung aufgebaut. Hier sehen Sie unsere erste Schablone: den FunktionsMASTER für funktionale Anforderungen. In diesem Abschnitt erklären wir Ihnen, welche Schritte Sie durchführen müssen, um die einzelnen Bestandteile im FunktionsMASTER mit Inhalt zu füllen.

<Ausdruck> bedeutet einen Platzhalter: Für Ausdruck müssen Sie etwas einsetzen.

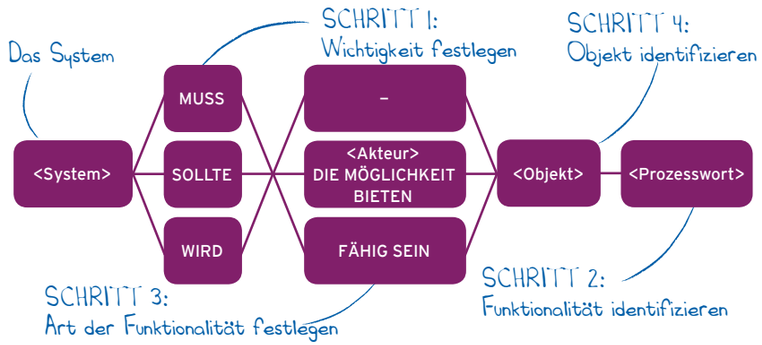



**Abbildung 10.2:** FunktionsMASTER ohne Bedingung

Gehen Sie für jede funktionale Anforderung von Schritt 1 bis Schritt 4 vor und füllen Sie die Anforderungsschablone aus. Dabei bewegen Sie sich anhand der Verbindungslinien in der Schablone von einem Bestandteil der Anforderung zum nächsten. Haben Sie dies einige Male gemacht, werden Sie merken, dass Sie den FunktionsMASTER bereits verinnerlichen. Nach einigen Anforderungen mehr werden Sie automatisch in Schablonenform schreiben können, ohne nachdenken zu müssen.

Falls Sie die Schablonen verwenden wollen, so müssen Sie deren Verwendung inklusive einem Beispiel, die etwaige verbindliche Nutzung etc. in Ihrem RE-Leitfaden erklären.

Eine Anforderung werden wir gemeinsam erzeugen. Üben müssen Sie dann selbst. Den Bestandteil <System> nehmen wir als gegeben an. Er tritt in der Schablone immer als Subjekt auf. Wir können deshalb gleich mit dem ersten Schritt beginnen.

Falls Ihr System noch keinen Namen hat, sollten Sie einen definieren und hier einsetzen! Wollen Sie dies nicht, schreiben Sie einfach „Das System" oder wie wir dies weiter unten auch getan haben „Das Bibliothekssystem."

Wir entscheiden uns für eine juristisch verbindliche Anforderung und müssen deshalb das Modalverb MUSS einsetzen.

Rechtliche Verbindlichkeit: siehe Kapitel 1: „In medias RE"

**Schritt 1:** Legen Sie fest, wie wichtig Ihnen die Anforderung ist.

Das Bibliothekssystem muss ... Unser erstes Anforderungsfragment

**Schritt 2:** Identifizieren Sie die geforderte Funktionalität.

Siehe Regel 2 in Kapitel 7 „Das SOPHIST-*RE*gelwerk"

Im Mittelpunkt jeder Anforderung steht die geforderte Funktionalität (wie drucken, speichern, übertragen, berechnen), die wir im Folgenden mit dem Begriff *Prozess* bezeichnen. Prozesse sind Vorgänge oder Tätigkeiten und sollten ausschließlich durch Vollverben definiert werden – die Prozesswörter. Ein Vollverb besitzt grammatikalisch gesehen eine hervorgehobene

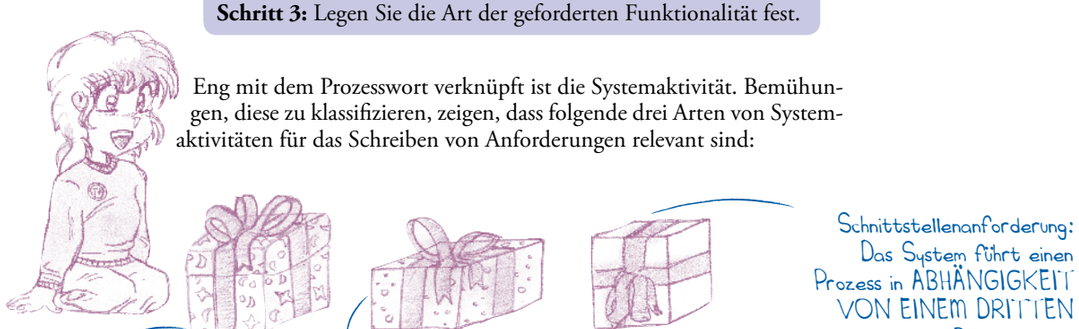




Stellung, da sich der gesamte Satz – oder in unserem Fall die gesamte Anforderung – an das Prozesswort bindet.

Das Bibliothekssystem muss drucken. Unsere erste Anforderung – als vollständiger Satz (Anforderung LH-Bib-001, Version 1)

**Schritt 3:** Legen Sie die Art der geforderten Funktionalität fest.




Eng mit dem Prozesswort verknüpft ist die Systemaktivität. Bemühungen, diese zu klassifizieren, zeigen, dass folgende drei Arten von Systemaktivitäten für das Schreiben von Anforderungen relevant sind:

**Abbildung 10.3:** Arten von Systemaktivitäten

Für jede dieser Systemaktivitäten gibt es einen eigenen Weg durch den in Abbildung 10.2 gezeigten FunktionsMASTER. Ihnen mag die Frage auf den Lippen liegen, ob nicht auch eine Anforderung an die Benutzerinteraktion einfach nur eine Schnittstellenanforderung ist. Das stimmt bis zu einem gewissen Grad: Bei einer Schnittstellenanforderung ist unserem System gleichgültig, welches System an die entsprechende Schnittstelle angedockt ist. Bei einer Benutzerinteraktion wird sehr häufig genau spezifiziert, welches Verhalten welcher Benutzergruppe auf welche Art und Weise zur Verfügung gestellt wird. Daher haben wir die Benutzerinteraktion in einen eigenen Pfad der Schablone gepackt, der diesen Aspekt mit spezifiziert.

Das Bibliothekssystem muss drucken.

Anforderung LH-Bib-001, Version 1: Selbsttätige Systemaktivität. Das System löst das Drucken aus. Wir müssen an der Anforderung laut Schablone nichts ändern.

Anforderung LH-Bib-002, Version 1: Benutzerinteraktion. Der Benutzer löst das Drucken aus.

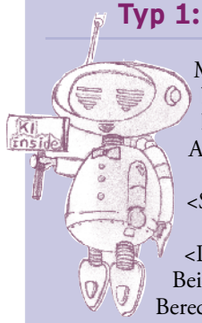
Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten zu drucken.

Anforderung LH-Bib-003, Version 1: Schnittstellenanforderung. Ein eingehendes Ereignis aus einem externen System löst das Empfangen aus.

Das Bibliothekssystem muss fähig sein, Ausleihdaten einer anderen Bibliothek zu empfangen.

## Die unterschiedlichen Systemaktivitäten

### Typ 1: Selbsttätige Systemaktivität




Mit dem ersten Schablonentyp werden Anforderungen konstruiert, bei denen das System den Prozess selbsttätig startet und ausführt. Der Benutzer tritt dabei nicht in Erscheinung. Es ergibt sich folgendes Anforderungsgerippe:

<System> <muss|sollte|wird> <Prozesswort>

<Prozesswort> bezeichnet das in Schritt 2 ausgewählte Prozesswort, zum Beispiel „drucken" für eine Druckfunktionalität oder „berechnen" für eine Berechnung, die das System durchführt.

### Typ 2: Benutzerinteraktion

Stellt das System dem Nutzer eine Funktionalität zur Verfügung, die der Benutzer startet (etwa über eine Eingabe- oder Auswahlmaske) oder tritt es mit ihm in Interaktion, dann werden die Anforderungen nach Schablonentyp 2 konstruiert:

Bitte konkrete Rollenbezeichnungen wie „der Bibliothekar" verwenden.

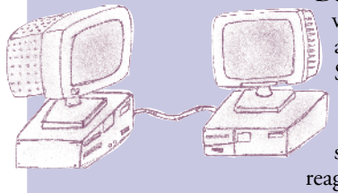
<System> <muss|sollte|wird> <Akteur> DIE MÖGLICHKEIT BIETEN <Prozesswort>

„Benutzer" oder „Anwender" sind zu allgemein und finden nur dann Verwendung, falls Sie ausdrücken wollen, dass die Funktionalität für Rollen konfigurierbar ist.

Das sind ihre Benutzerrollen. Bitte mit dem Rechte- und Rollenkonzept, der Benutzerverwaltung bzw. dem Glossar abgleichen.

Der Nutzer, zum Beispiel der Bibliothekar, der über die Funktionalität verfügen soll, wird mit <Akteur> in die Anforderung eingebunden. Der <Akteur>-Bestandteil kann nur dann als optional betrachtet werden, wenn es eindeutig ist, wem das System die geforderte Funktionalität liefert. Wir haben sie im Rahmen der Schablonen nicht als optional gekennzeichnet, da das Weglassen einen Tilgungseffekt darstellt. Selbst wenn die Funktionalität allen potenziellen Nutzern zur Verfügung gestellt wird, sollte dies notiert werden.

### Typ 3: Schnittstellenanforderung



Deckt den Fall ab, in dem das System nur in Abhängigkeit von Dritten potenziell eine Aktion ausführt – ausgenommen der Benutzer. Stellen Sie sich dazu ein System A vor, das seine Informationen nicht vom Benutzer, sondern von einem Nachbarsystem B erhält. Diese Eingaben können azyklisch und unvorhersehbar sein. Immer wenn Nachrichten oder Daten eintreffen, reagiert unser System A und führt eine Funktion aus.





Achtung: Hier müssen Sie prüfen: Falls System A eine Typ-3-Anforderung besitzt, muss es in System B eine Typ-1- oder Typ-2-Anforderung dazu geben.

Zur Formulierung dieses geforderten Sachverhalts sind sowohl Typ 1 (selbsttätige Systemaktivität) als auch Typ 2 (Benutzerinteraktion) ungeeignet. Letzterer scheidet aufgrund der mangelnden Benutzeraktivität aus. Typ 1 sollte nicht verwendet werden, da das System auf Grund eines externen Ereignisses einen Prozess startet und ausführt. Aus diesem Grund hat sich die folgende Formulierung als geeignet erwiesen.

<System> <muss|sollte|wird> FÄHIG SEIN <Prozesswort>

Tipp aus der Praxis: Anforderungen mit dem Prozesswort „senden" sind keine Typ-3-Anforderungen, sondern entweder als Typ 1 oder als Typ 2 zu formulieren.

**Schritt 4:** Identifizieren Sie das Objekt, für das die Funktionalität gefordert wird.

Aus den bisherigen Beispielen wird bereits deutlich, dass es sich nur um den Kern einer Anforderung handeln kann. Zur Vollständigkeit fehlen noch weitere Bestandteile. In Aus den bsiherigen Beispielen wird bereits deutlich, dass es sich nur um den Kern einer Anforderung handeln kann. Zur Vollständigkeit fehlen noch weitere Bestandteile. In der Anforderung LH-Bib-002, Version 1 ist keineswegs klar, **WAS** gedruckt werden soll. Sprachwissenschaftler würden hierbei von fehlenden Objekten sprechen. Einfach ausgedrückt, fehlt es an der näheren oder ergänzenden Bestimmung des Prozesworts „drucken". Objekte werden in der Anforderungsschablone vor dem Prozesswort eingefügt. Unsere Anforderung sieht demnach so aus:

Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten, einen Benutzerausweis zu drucken.

Ergänzt um Objekt: Anforderung LH-Bib-002, Version 2

Damit liegt eine Anforderung vor, die bereits eine gute Qualität aufweist. Die Qualität können wir allerdings noch steigern, indem wir zwei weitere Schritte unternehmen. Zum einen können wir in die Anforderung Bedingungen integrieren, unter denen der Prozess ausgeführt werden soll. Zum anderen sollten Sie die entstandene Anforderung noch einmal mittels des SOPHIST-*RE*gelwerks auf noch vorhandene Effekte prüfen. Das sind die Schritte fünf und sechs bei der Erzeugung von Anforderungen mittels Anforderungsschablone.

**Schritt 5**: Legen Sie die Bedingungen fest, unter denen die geforderte Funktionalität durchgeführt wird.

Für Anforderungen ist es typisch, dass die geforderte Funktionalität nicht fortwährend, sondern nur unter bestimmten zeitlichen oder logischen Bedingungen ausgeführt oder zur Verfügung gestellt wird.

Bedingungen stehen in einem Nebensatz am Anfang der Anforderung. Das Voranstellen der Bedingungen hat zur Folge, dass die Satzstellung der Anforderung umgebaut werden muss. So rückt das Modalverb „muss" vor das Subjekt „System". Wir unterscheiden daher den FunktionsMASTER mit Bedingung (Abbildung 10.4) von dem FunktionsMASTER ohne Bedingung (Abbildung 10.2) und erweitern unsere Schablone:

SCHRITT 5: Formulieren Sie logische und zeitliche Bedingungen.

SCHRITT 6: SOPHIST-*RE*gelwerk anwenden

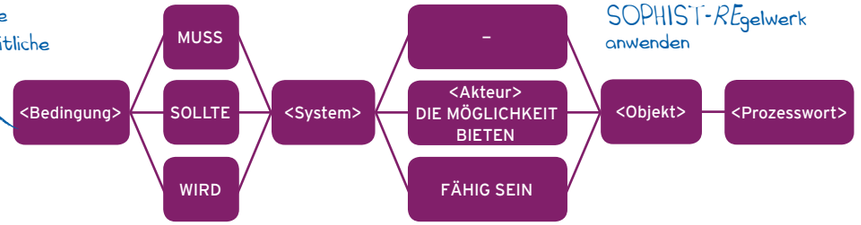



**Abbildung 10.4:** FunktionsMASTeR mit Bedingung

Anforderung LH-Bib-002, Version 3: jetzt mit zeitlicher Bedingung

Sobald das Bibliothekssystem die Benutzerdaten gespeichert hat, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, einen Benutzerausweis zu drucken.

Den nachfolgenden schablonenbasierten Beispielanforderungen sind Bedingungen vorangestellt:

Beispiel Typ 1: Selbsttätige Systemaktivität

Falls der eingegebene Kunde bereits im System vorhanden ist, muss das System eine Fehlermeldung ausgeben.

Beispiel Typ 2: Benutzerinteraktion

Falls ein Kunde noch nicht im Bibliothekssystem vorhanden ist, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, Kundendaten einzugeben.
Falls ein Bibliothekskunde keine Leihobjekte entliehen hat, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, diesen Bibliothekskunden zu löschen.

Beispiel Typ 3: Schnittstellenanforderung

Solange das Bibliothekssystem in Betrieb ist, muss das Bibliothekssystem fähig sein, Daten für ein Software-Update zu empfangen.

Detaillierte Informationen zu Aufbau und Inhalt von Bedingungsnebensätzen finden Sie in diesem Kapitel in Abschnitt 10.8 „Bedingungen in Anforderungen".
Sobald Sie einen komplexeren Prozess beschreiben, der aus mehreren Schritten besteht, müssen Sie die Beschreibung in mehrere Sätze zerlegen. Jeder dieser Sätze kann dabei nach einem anderen Schablonentyp gebaut werden. Das folgende Beispiel veranschaulicht diesen Aspekt.

Das Bibliothekssystem muss dem Administrator die Möglichkeit bieten, Leihobjekte von externen Datenmedien in den Datenbestand des Bibliothekssystems zu importieren.

Falls während des Importierens ein semantischer oder syntaktischer Datenfehler auftritt, muss das Bibliothekssystem dem Administrator für einzelne zu importierende Leihobjekte den jeweiligen Fehler (Fehlernummer und Fehlerbeschreibung) auf dem Bildschirm anzeigen.

Nachdem das Bibliothekssystem Daten neu eingegebener Bibliothekskunden oder Leihgegenstände vom benachbarten Bibliothekssystem empfangen hat, muss das Bibliothekssystem die empfangenen Daten persistent speichern.

**Schritt 6:** Wenden Sie das SOPHIST-*RE*gelwerk auf Ihre konstruierte Anforderung an.





Nach dem fünften Schritt weist jede nach FunktionsMASTeR konstruierte funktionale Anforderung eine vollständige Struktur auf. Unabhängig davon ist die Anforderung noch nicht perfekt, da mit wenigen Ausnahmen die Semantik immer noch in der Gestaltungsfreiheit des Autors liegt. So lässt unser Vorgehen in Schritt 4 dem Anfordernden die Möglichkeit, das Satzobjekt weitestgehend frei zu gestalten. Konsistenz und Vollständigkeit liegen somit in der Hand des Verfassers. Dadurch ist es nicht ausgeschlossen, dass sprachliche Effekte auftreten. Daher ist es sinnvoll, abschließend das SOPHIST-*RE*gelwerk (siehe Kapitel 7 „Das SOPHIST-*RE*gelwerk") zur Vervollständigung der semantischen Bedeutung anzuwenden.

Siehe Regel 10 aus dem SOPHIST-*RE*gelwerk in Kapitel 7 "Das SOPHIST-*RE*gelwerk"

Durch die Vorstrukturierung der Anforderung mittels FunktionsMASTeR wird die Anwendung des SOPHIST-*RE*gelwerks nicht mehr alle sprachliche Effekte aufweisen, sodass der Aufwand für deren Behebung deutlich geringer ist, als bei Anforderungen, die ohne den Einsatz des FunktionsMASTeR formuliert wurden.

Im folgenden Beispiel beinhaltet die Anforderung ein verwendetes Mengenwort (**ALLE** Daten), obwohl die Anforderung nach dem Schablonentyp 1 konstruiert wurde.

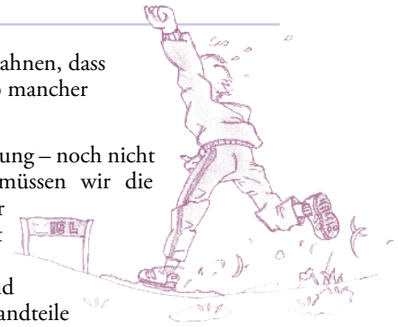
Hier entdecken Sie das Mengenwort ALLE und müssen fragen: „Wirklich alle Daten?

An jedem ersten Tag eines Monats um 8.00 Uhr und unter der Bedingung, dass der Administrator die automatische Backup-Funktion aktiviert hat, muss das Bibliothekssystem alle Daten archivieren.

An jedem ersten Tag eines Monats um 8.00 Uhr und unter der Bedingung, dass der Administrator die automatische Backup-Funktion aktiviert hat, muss das Bibliothekssystem die Daten des Bibliothekssystems, die seit dem letzten Backup verändert wurden, automatisch archivieren.

Daraus ergibt sich diese verbesserte Anforderung.

## 10.4 Semantische Präzisierung der Anforderungsschablone



Der Blick in vorhandene Anforderungsdokumente lässt erahnen, dass wir qualitativ vielleicht schon weitergekommen sind als so mancher Anforderungsschreiber zuvor.

Dennoch haben wir das Ziel – eine wirklich gute Anforderung – noch nicht erreicht. Um die Anforderung perfekt zu gestalten, müssen wir die semantischen Bestandteile konkretisieren. Denn der FunktionsMASTeR, wie wir ihn bis jetzt kennen, bildet nur eine Hülle, die die Bestandteile integriert und eine Anleitung gibt, wie diese abgestimmt, redundanzfrei und optimiert eingesetzt werden können. Die einzelnen Bestandteile müssen wir mit Semantik, also Inhalt, versehen. Konkret heißt das: Wir müssen semantische Definitionen der Inhalte der Bestandteile der Anforderungsschablone erstellen, sowie logische Operatoren und semantisch prägnante Darstellungen verwenden. Wir zeigen Ihnen, wie Sie die ersetzbaren bzw. wählbaren Bestand-teile der Schablonen mit definierten Begriffen und Wörtern füllen.

Unterschiedliche Verfasser müssen bei der Formulierung des gleichen Sachverhalts die gleiche Wortwahl treffen.

Wie die Anforderungen selbst erhalten die Definitionen eine rechtliche Verbindlichkeit.

Dass unterschiedliche Personen wirklich die gleiche Wortwahl treffen, ist jedoch nur dann möglich, wenn der zur Verfügung stehende Wortschatz begrenzt und die Bedeutung der verwendbaren Wörter eindeutig definiert ist.

### 10.4.1 Rechtliche Verbindlichkeiten

Grundlagen dazu in Kapitel 1 „In medias RE"

Zunächst müssen die Schlüsselwörter für die rechtliche Verbindlichkeit einer Anforderung sowie die Bedeutung dieser Schlüsselwörter festgelegt werden. Dies geschieht in einer zweispaltigen Tabelle – egal in welcher Sprache – und könnte so aussehen:

Ausflug zum Test: Alle Muss-Anforderungen werden getestet. Nur realisierte Sollte-Anforderungen werden getestet. Wird-Anforderungen werden nicht getestet.

| Rechtliche Verbindlichkeit | Schlüsselwort |
|---|---|
| Der Ausdruck *muss* wird benutzt, um verpflichtende Anforderungen zu definieren.<br>■ Die definierte Anforderung ist verpflichtend.<br>■ Die Erfüllung der Anforderung im Produkt ist verpflichtend.<br>■ Die Abnahme des Produkts kann verweigert werden, wenn einer *muss*-Anforderung nicht entsprochen wurde. | **muss** |
| Der Ausdruck *sollte* wird benutzt, um verpflichtende Anforderungen zu definieren.<br>■ sind nicht verpflichtend.<br>■ müssen nicht erfüllt werden.<br>■ dienen der besseren Zusammenarbeit von Stakeholdern und Entwicklern.<br>■ erhöhen die Zufriedenheit der Stakeholder. | **sollte** |
| Der Ausdruck *wird* wird benutzt, um Anforderungen zu definieren, die in der Zukunft integriert werden.<br>■ Zukünftige Anforderungen sind verpflichtend.<br>■ Sie helfen in der aktuellen Lösung, Vorbereitungen zu treffen, um Zukünftiges später optimal zu integrieren. | **wird** |

**Abbildung 10.5:** Definition der Schlüsselwörter zur rechtlichen Verbindlichkeit

In vielen Projekten machen wir die Erfahrung, dass nur Muss-Anforderungen benötigt werden.

Dazu schreiben Sie eine kurze Erklärung (ein ausführliches Beispiel, das Ihnen als Vorlage dienen kann, steht unter www.sophist.de/re6/kapitel10 bereit). Etwa so: „Anforderungen enthalten unterschiedliche rechtliche Verbindlichkeiten. Die rechtliche Verbindlichkeit ist in der jeweiligen Anforderung durch ein Schlüsselwort gekennzeichnet. Mögliche Schlüsselwörter entnehmen Sie der folgenden Tabelle:"

Beschreiben Sie in der Tabelle nur die Schlüsselwörter, die Sie benötigen. Dazu erklären Sie dem Leser, was die rechtliche Verbindlichkeit bedeutet und wie sie eingesetzt wird. Haben Sie die Definitionen fertig, legen Sie diese mit Erklärung in Ihrem RE-Leitfaden ab, denn jedem Leser und Bearbeiter von Anforderungen müssen diese Informationen zur Verfügung stehen. In einem Anforderungsdokument integrieren Sie die rechtliche Verbindlichkeit in eines der ersten Kapitel – bei Inhalten wie Einführung oder Leseanleitung.





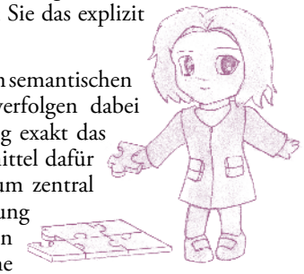
### 10.4.2 Verben – Prozesswörter

Rein sprachlich gesehen sind Prozesswörter Vollverben. Sie geben die Funktionalität des Systems wieder und damit den Kern der Anforderung. Zur Klarstellung dieser Funktionalität ist es wichtig, genau zu wissen, was der Autor der Anforderung fordert.

Was genau versteht er unter einem bestimmten Verb?

Den Umgang mit Prozesswörtern und mit deren Wiederverwendung beschreiben Sie ebenso in Ihrem RE-Leitfaden.

Geht es in Ihrem Umfeld bei „eingeben", „einsetzen" und „einfügen" („enter", „input", „insert") um verschiedene Dinge? Falls ja, müssen Sie das explizit darstellen.



Es gilt, die Prozesswörter genauer zu untersuchen und in deren semantischen Bedeutung konkret und eindeutig zu definieren. Wir verfolgen dabei das Ziel, dass jeder Verfasser oder Leser einer Anforderung exakt das Gleiche unter einem bestimmten Verb versteht. Als Hilfsmittel dafür eignet sich eine Prozesswortliste. Darin werden – wiederum zentral – alle Vollverben, die in den Anforderungen Verwendung finden, gesammelt und definiert. Gehen Sie dabei am besten so vor, dass Sie in den ersten Wochen Ihres Projekts sämtliche Unterlagen, die bereits vorhanden sind, nach Prozesswörtern durchsuchen.

Hier dürfen Sie sich ins stille Kämmerchen zurückziehen und alle Verben in eine Liste schreiben!

Prozesswörter erhalten Sie – neben den üblichen Quellen für Anforderungen (siehe Kapitel 5 „Ziele, Informanten und Fesseln") – auch aus UML-Diagrammen (siehe Kapitel 9 „Systemanforderungen dokumentieren"). Das Aktivitätsdiagramm beinhaltet Verben sowohl im Namen von Aktivitäten als auch im Namen von Aktionen oder Verhaltensaufrufen. Im Use-Case-Diagramm finden Sie Verben sowohl im Namen des Use-Case-Diagramms als auch in den Namen der Use-Cases. Auch in Zustandsdiagrammen sind Verben enthalten: Aktivitäten an Transitionen oder Trigger, die eine Benutzerinteraktion oder Schnittstellenfähigkeit beschreiben. Zu guter Letzt stoßen Sie in Klassendiagrammen auf Verben. Operationen, die den einzelnen Klassen zugeordnet sind, oder Assoziationen dienen hier als Quelle.

Kommen auch in agilen Dokumentationsformen wie User-Storys vor und sollten auch dort möglichst eindeutig sein. Siehe Kapitel II „Dokumentation im agilen Umfeld"

Diese Liste der Vollverben erweitern Sie zu einer Tabelle, in der es drei Spalten gibt: eine für das Prozesswort, eine für die Bedeutung des Prozessworts und eine für Synonyme. Machen Sie sich ebenso Gedanken über Homonyme. Benötigen Sie für Ihre Anforderungen ein Homonym, so müssen Sie in der Erklärung der Bedeutung dieses Worts klar machen, welche der möglichen Bedeutungen Sie verwenden.

Als SYNONYM wird ein Wort bezeichnet, das mit einem anderen oder mehreren anderen Wörtern derselben Sprache bedeutungsgleich ist.

Etwas von A nach B SCHICKEN, ÜBERMITTELN, SENDEN, ÜBERTRAGEN

Als HOMONYM wird ein Wort bezeichnet, das in derselben Sprache mehrere unterschiedliche Bedeutungen besitzt.

ABSETZEN: 1. Einen Gegenstand auf dem Boden absetzen
2. Etwas von der Steuer absetzen
3. Jemanden von einem hohen Amt absetzen

Prozesswörter und deren Bedeutung müssen als rechtlich verbindlich gelten. In der semantischen Definition machen Sie dies mit dem Signalwort MUSS kenntlich.

Nur diese Prozesswörter werden in Anforderungen genutzt.

Wörter, die hier auftauchen, sind für Anforderungen tabu!

| Prozesswort | Semantische Definiton des Prozessworts | Synonyme |
|---|---|---|
| anzeigen | Im Bibliothekssystem muss *anzeigen* den Prozess bedeuten, dem Benutzer Informationen auf dem Bildschirm darzustellen. | darstellen, ausgeben |
| speichern | Im Bibliothekssystem muss *speichern* den Prozess bedeuten, Informationen persistent aufzubewahren. | sammeln, aufbewahren |

Verbindliche Muss-Definition: Verwenden Sie das Prozesswort „anzeigen" und die Anzeige erfolgt auf einem Nebendisplay statt auf dem Bildschirm, so ist das ein schwerwiegender Fehler.

**Abbildung 10.6:** Auszug aus Prozesswortliste

In einem Anforderungsdokument wandert die Liste meist in das Glossarkapitel oder in den Anhang, auf den referenziert werden kann.

Die größte Herausforderung beim Füllen der Prozesswortliste ist sicherlich die Erstellung der semantischen Definition in Spalte zwei. Auch hierfür gibt es Schablonen, die wir im Internet unter www.sophist.de/re6/kapitel10 bereitstellen.

Regeln Sie auch, wer die Konsistenz zwischen der Prozesswortliste und ihrer Spezifikation sicherstellt.

Auch die Prozesswortliste stellen Sie zentral für alle Projektmitglieder zusammen mit dem Glossar (siehe Kapitel 8 „Grundlagen für die Systemanalyse dokumentieren" bzw. folgender Abschnitt 10.4.3) zur Verfügung. Sie unterliegt über die gesamte Projektlaufzeit eventuellen Änderungen – die sich aber meist im Lauf der Projektzeit verringern. Sie sollten deshalb einen Verantwortlichen bestimmen, der diese Liste fortwährend pflegt und aktuell hält.

Hinweis: Prozesswörter lassen sich sehr gut wiederverwenden. Beispiele hierfür sind: anlegen, bearbeiten, speichern, archivieren, historisieren, drucken, anzeigen etc. Definieren Sie diese Prozesswörter einmal, erzeugen Sie daraus eine Prozesswortliste und verwenden Sie diese auch in zukünftigen Projekten. Damit sparen Sie sich den Aufwand, diese Verben immer wieder neu zu definieren. Mehr zu Wiederverwendungskonzepten finden Sie in Kapitel 21 „Wiederverwendung".

### 10.4.3 Substantive – Akteure, Rollen, Objekte, Eigenschaften und Abkürzungen

Neben der juristischen Verbindlichkeit und den Prozesswörtern werden beim Ausfüllen der Anforderungsschablone in Schritt 4 die für die jeweilige Anforderung relevanten Objekte eingetragen. Bei den Objekten handelt es sich um Substantive. Diese treten in Systembeschreibungen etwa als Objekte bzw. Klassen, Eigenschaften/Attribute, Akteure, Rollen oder externe Systeme oder Organisationen auf. Wie bei den Prozesswörtern gilt auch hier die Regel, die zu verwendenden Substantive zu begrenzen und eindeutig zu definieren.

Weitere Informationen zum Glossar finden Sie in Kapitel 8 „Grundlagen für die Systemanalyse dokumentieren."

Erzeugen Sie dafür ein Glossar, in dem Sie die für Ihr Projekt relevanten Begriffe aufführen und definieren. Für den Anfang reicht es aus, eine Liste zu erstellen. Zum Aufbau dieses Glossars gehen Sie genauso vor wie schon bei den Prozesswörtern: Prüfen Sie zu einem frühen Zeitpunkt im Projekt alle schon vorhandenen Artefakte auf Substantive und fügen Sie diese in Ihr Glossar ein. Zu jedem Begriff müssen Sie eine semantische Definition sowie dessen Synonyme integrieren. Auch hier können Sie eine Schablone verwenden. Sie steht bereit unter: www.sophist.de/re6/kapitel10.

Wichtiges Ziel: Alle Projektbeteiligten sprechen die gleiche Sprache.

Denken Sie daran, dass Sie die semantische Definition der Substantive rechtlich verbindlich mit MUSS gestalten. Über die Bedeutung von einzelnen Wörtern sollte nach deren Abstimmung und Festlegung nicht mehr diskutiert werden.





Mit dieser Liste von Begriffen – dem Glossar – stellen Sie sicher, dass jeder Projektbeteiligte mit einem Begriff die gleiche Bedeutung verbindet, und beugen somit Missverständnissen und Unklarheiten in der Kommunikation vor.

Außer in Texten tauchen Substantive auch in verschiedenen semiformalen Notationsformen auf, so etwa in UML-Modellen (Anforderungsmodell, Analysemodell) als Klassen oder Attribute in Klassendiagrammen. In Aktivitätsdiagrammen finden Sie Substantive im Aktivitätsnamen, in Aktionen oder Verhaltensaufrufen, in Aktivitätsbereichen oder als Akteure. Akteure treten ebenso in Use-Case-Diagrammen auf. Weitere Substantive finden Sie im Bezeichner von Use-Cases. Im Sequenzdiagramm bilden die Kommunikationspartner die Quelle für Substantive oder Sie finden diese in Nachrichtennamen.

Mehr zur UML und wie Sie deren Diagramme zur Anforderungsdokumentation verwenden lesen Sie in Kapitel 9 „Systemanforderungen dokumentieren".

Egal in welchen Artefakten: Sobald Sie bei der Analyse auf ein Substantiv stoßen, prüfen Sie, ob es bereits in Ihrem Glossar vorhanden ist. Ist es dort noch nicht aufgenommen und existiert auch kein Synonym dazu, erstellen Sie einen neuen Begriff im Glossar und definieren Sie diesen. So erweitern Sie kontinuierlich Ihre Begriffswelt. Natürlich gilt auch hier, dass sich ein Projektmitglied verantwortlich um die Erstellung und Pflege des Glossars kümmern muss. Abbildung 10.7 zeigt einen Auszug aus dem Glossar unseres Bibliothekssystems.

Fachbegriffe können sehr gut als Begriffsmodell dargestellt werden. Optimal ist eine Kombination von Begriffsmodell (Strukturen und Beziehungen) und sprachlichem Glossar (Inhalt).

| Begriff | Semantische Definiton des Begriffs | Synonyme |
|---|---|---|
| Kunde | Im Bibliothekssystem muss *Kunde* eine natürliche Person, die berechtigt ist, ein Leihobjekt zu reservieren und auszuleihen, bedeuten | Bibliothekskunde, Ausleiher |
| Leihobjekt | Im Bibliothekssystem muss *Leihobjekt* ein physisches Exemplar eines Buchs, einer DVD oder einer Zeitschrift, das entliehen werden kann, bedeuten. | Leihgegenstand |

**Abbildung 10.7:** Auszug aus Glossar

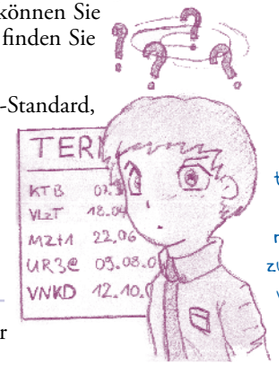
Regeln für den Umgang mit dem Glossar! Ab damit in den RE-Leitfaden.

Zu guter Letzt geht es um die sehr oft in der täglichen Kommunikation verwendeten Abkürzungen, die in Anforderungen als Vertreter für jegliche Art von Substantiven stehen – und die von vielen Lesern nicht verstanden werden. Verwendete Abkürzungen müssen deshalb in Ihrem Glossar aufgeführt und einem Substantiv zugeordnet werden. Fügen Sie dazu in Ihrem Glossar eine weitere Spalte „Abkürzung" ein. Alternativ können Sie losgelöst vom Glossar ein Abkürzungsverzeichnis erstellen. Auch hierzu finden Sie weitere Informationen unter: www.sophist.de/re6/kapitel10.

Beispiel: ISBN für Internationale Standardbuchnummer

Sie können dabei gerne eine Referenz auf die offizielle Quelle (einen ISO-Standard, eine DIN-Norm oder einfach eine URL) einfügen.

Klären Sie auch, wie die Konsistenz zwischen Glossar und Spezifikation sichergestellt wird.




Kennen Sie das auch? Projektbeteiligte reden mit Ihnen nur noch in Abkürzungen und Sie verstehen nur Bahnhof?

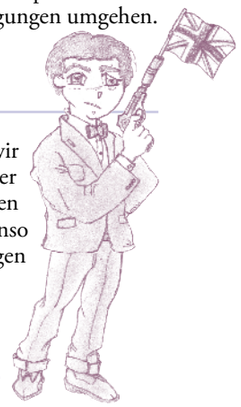
### 10.4.4 Bedingungen

Komplexe Systeme neigen dazu, bestimmte Funktionalitäten erst unter einer Reihe von Vorbedingungen zur Verfügung zu stellen.

Sie müssen eindeutig klarstellen, unter welchen Voraussetzungen die geforderte Funktionalität vom System bereitgestellt wird.

Durch eine genauere Formulierung der Bedingungen in Schritt 5 schaffen wir eine einheitliche Schreibweise und stellen Konjunktionen als Signalwörter an den Beginn jeder Bedingung. Lesen Sie dazu Abschnitt 10.8 „Bedingungen in Anforderungen“ in diesem Kapitel. Dort ist detailliert beschrieben, wie Sie mit dem Aufbau und dem Inhalt von Bedingungen umgehen.

## 10.5 Konstruieren in englischer Sprache



Diese Schablone verhilft Kollegen mit schlechterem Englisch zu grammatikalisch richtigen Anforderungen.

Da viele Spezifikationen in englischer Sprache geschrieben werden, haben wir den FunktionsMASTER in die englische Sprache übertragen. Wir gehen hier nicht mehr auf jeden einzelnen Schritt ein. Nehmen Sie den englischsprachigen FunktionsMASTER und füllen Sie die einzelnen Bestandteile aus – ebenso wie wir dies in den vorhergehenden Abschnitten für den deutschsprachigen FunktionsMASTER beschrieben haben – wir folgen dem gleichen Prinzip.

### 10.5.1 Der Syntaxbauplan im Englischen

RE-Bauernregel: Hilft dir die deutsche Sprache nix, mach's Englisch, das geht auch ganz fix.

Für Ihre Anforderungen finden Sie hier den FunctionalMASTER in englischer Sprache. Da der Satzbau mit und ohne Bedingung im Englischen gleich ist, müssen wir keine zwei Varianten der Schablone betrachten.

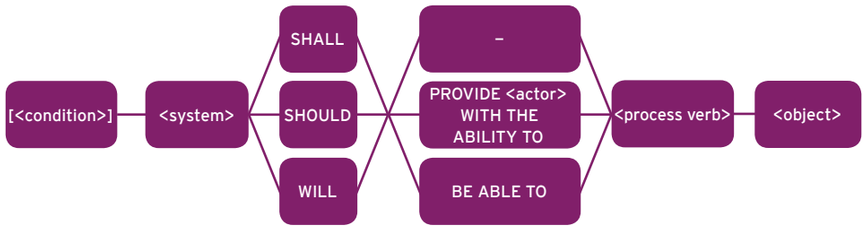



**Abbildung 10.8:** FunctionalMASTER

Am prinzipiellen Vorgehen – der Konstruktion in sechs Schritten – ändert sich im Englischen nichts. Zur Verdeutlichung noch einige Beispiele von englischen, schablonenbasierten Anforderungen.

Typ 1: Selbsttätige Systemaktivität

If a video is returned AND if this video was lent the 30th time, the library system shall display a message that the video has to be replaced.

If no library item is lent by a customer, the library system shall provide the librarian with the ability to remove this customer.

Typ 2: Benutzerinteraktion

If the library system is in use, the library system shall be able to receive data for a software-update from a central administration server.

Typ 3: Schnittstellenanforderung





### 10.5.2 Semantische Normierung im Englischen

Falls Sie Stakeholder haben, die nicht immer das richtige englische Verb wählen, hilft eine englische Prozesswortliste enorm.

Auch im Englischen sollten Sie Prozesswörter in einer Prozesswortliste und Begriffe in einem Glossar semantisch definieren und diese Definitionen als rechtlich verbindlich kennzeichnen. Ebenso kümmern Sie sich bitte um Synonyme und Homonyme und machen klar, welche Abkürzung für welchen Begriff steht.

Durch die Änderung der Sprache hat sich an Vorgehen, Tätigkeiten und zu erzeugenden Artefakten nichts geändert. Gehen Sie genauso vor, wie wir es Ihnen für die deutsche Sprache im Abschnitt 10.4 „Semantische Präzisierung der Anforderungsschablone“ erklärt und gezeigt haben.

So entsteht auch im Englischen eine Prozesswortliste.

| Process | Semantic definition of process word | Synonyms |
|---|---|---|
| to save | In the library system, *to save* shall be defined as the storing of information in the persistent memory of the library system. | to store |
| to select | In the library system, *to select* shall be defined as the process of choosing one or more elements of an endless amout of elements by the user. | to mark |

**Abbildung 10.9:** Auszug aus englischer Prozesswortliste

Als Wiederholung: Wir empfehlen, Abkürzungen in das Glossar aufzunehmen.

Ein in Tabellenform gebrachtes Glossar von Substantiven sehen Sie in Abbildung 10.10. Bitte die Abkürzungen nicht als weitere Spalte vergessen, falls Sie diese nicht in einem gesonderten Abkürzungsverzeichnis dokumentieren.

| Term | Semantic definition of the term | Synonyms | Abbreviation |
|---|---|---|---|
| library item | In the library system, a *library item* shall be defined as a physical entity of a book, a DVD or a magazine which can be borrowed by a customer. | borrowable item | LI |
| customer | In the library system, a *customer* shall be defined as a natural person, who is able to reserve and borrow the library items. | library customer, client | - |

**Abbildung 10.10:** Auszug aus englischem Glossar

Auch hier gilt: Es ist möglich, die deutschen Übersetzungen mit einzufügen. Das hilft den Kollegen, die im Englischen unsicher sind.

Diverse Schablonen zur semantischen Normierung von Prozesswörtern, Begriffen, allgemeingültigen Begriffen und Abkürzungen in englischer Sprache stellen wir bereit unter: www.sophist.de/re6/kapitel10.

Mit den vier bzw. fünf Schritten zur Füllung der Bestandteile des FunktionsMASTER und den semantischen Bedeutungsdefinitionen einzelner Wörter können Sie nun beginnen, Anforderungen zu schreiben. Es ist gewährleistet, dass in Anforderungssätzen benötigte Bestandteile vorhanden sind und einzelne Formulierungsregeln eingehalten werden.

Vereinzelte und vollständige Sätze oder Aktivformulierung

Für detailliertere Informationen zum Ausfüllen des FunktionsMASTER lesen Sie hier bitte weiter. Im folgenden Abschnitt gehen wir noch näher auf bestimmte Satzbestandteile ein.

## 10.6 Details für die Konstruktion

Der bis hierher vorgestellte FunktionsMASTER stellt die einfachste Schablone für funktionale Anforderungen dar und ist vor allem für Einsteiger geeignet. Für eine semantisch noch genauere Formulierung von Anforderungen untersuchen wir die beiden semantisch wichtigsten Bestandteile des Anforderungssatzes und machen uns das Wissen zur natürlichsprachlichen Analyse aus dem SOPHIST-*RE*gelwerk (siehe Kapitel 7 „Das SOPHIST-*RE*gelwerk“) zu Nutze.

Die beiden Bestandteile **<Objekt>** und **<Prozesswort>** des FunktionsMASTER lassen inhaltlich noch viel Gestaltungsspielraum beim Autor der Anforderung. Linguistische Untersuchungen zeigen, dass die inhaltlichen Vertreter dieser Bestandteile – das Substantiv bei <Objekt> und das Verb bei <Prozesswort> – mit weiteren Informationen angereichert werden können. Wir gelangen zu noch genauer spezifizierten Anforderungen und sprechen von:

- Präzisierung des Objekts
- Konkretisierung des Prozesswortes

Für alle Leser, die sich tiefer mit der Anforderungsformulierung nach Schablonen beschäftigen wollen, stellen wir den Detaillierten FunktionsMASTER vor. Er stellt keine neue oder andere Schablone dar. Er ist vielmehr eine Erweiterung des FunktionsMASTER, angereichert mit sinnvollen W-Fragen rund um das <Objekt> und das <Prozesswort>.

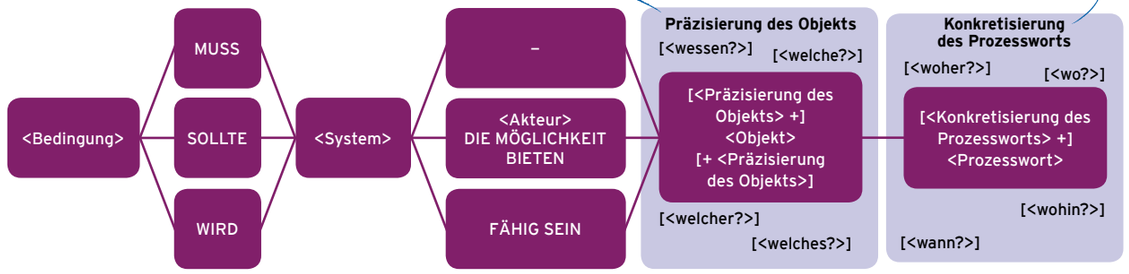



**Abbildung 10.11:** Detaillierter FunktionsMASTER mit Bedingung

Wir erinnern uns zurück an unser Anforderungsbeispiel des Typs 2, der Benutzerinteraktion:

Sobald das Bibliothekssystem die Benutzerdaten gespeichert hat, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, einen Benutzerausweis zu drucken.

### 10.6.1 Präzisierung des Objekts

Um den Bestandteil **<Objekt>** des Anforderungssatzes zu präzisieren, erweitern wir Schritt 4 aus der Anleitung im Abschnitt 10.3 „Schritt für Schritt zur Anforderung“.





Der syntaktische Aufbau dieses Bestandteils im Detaillierten FunktionsMASTER ändert sich auf die Form, die bereits in Abbildung 10.11 ersichtlich ist:

> **Detaillierter Schritt 4:** Identifizieren Sie das Objekt, für das die Funktionalität gefordert wird, sowie zusätzliche Informationen, die sich auf das Objekt beziehen und dieses genauer beschreiben.

**[<Präzisierung des Objekts> +] <Objekt> [+ <Präzisierung des Objekts>]**

Daran lässt sich ablesen, dass Informationen, die das Objekt näher beschreiben, im Anforderungssatz entweder vor dem Objekt ODER nach dem Objekt stehen können.
Wir erweitern unsere Anforderung um weitere Informationen zum Objekt und erhalten:

Sobald das Bibliothekssystem die Benutzerdaten gespeichert hat, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, einen **gültigen** Benutzerausweis **eines registrierten Bibliothekskunden** zu drucken.

Mit den Fragewörtern *welche/r/s* werden Objekte um Eigenschaften oder Zustände ergänzt. Diese Präzisierungen werden dem Objekt vorangestellt. Inhalte zum Fragewort *wessen* präzisieren die Rolle oder den Akteur, für die/den das Objekt gilt, und stehen im Satzbau hinter dem Objekt.

### 10.6.2 Konkretisierung des Prozesswortes

Der Bestandteil **<Prozesswort>** wird konkretisiert, indem wir Schritt 2 aus der Schritt-für-Schritt-Anleitung erweitern.

> **Detaillierter Schritt 2:** Identifizieren Sie die geforderte Funktionalität sowie zusätzliche Informationen, die sich auf das Prozesswort beziehen und dieses genauer beschreiben.

Wir erhalten zu diesem Bestandteil die in Abbildung 10.11 dargestellte Syntax:

**[<Konkretisierung des Prozesswortes> +] <Prozesswort>**

Informationen, die das Prozesswort näher beschreiben, stehen im Anforderungssatz immer vor dem Prozesswort selbst, das den Hauptsatz abschließt.
Die um diese Information beispielhaft erweiterte Anforderung heißt dann:

Sobald das Bibliothekssystem die Benutzerdaten gespeichert hat, muss das Bibliothekssystem dem Bibliothekar die Möglichkeit bieten, einen gültigen Benutzerausweis eines registrierten Bibliothekskunden **auf einem Netzwerkdrucker** zu drucken.

Das Prozesswort wird mit Hilfe der Fragewörter *wann, wo, wohin* oder *woher* konkretisiert. Fragen, die mit diesen Fragewörtern gebildet werden, führen z.B. zu Zeitpunkten, zu denen ein Prozess gestartet werden soll oder verfügbar sein soll, oder zu Systemen, die mit dem betrachteten System interagieren bzw. einen Prozess ausführen.

Das Fragewort *wie* ist nicht als sinnvolles Fragewort vorgesehen, da sich daraus oft nichtfunktionale Aspekte ergeben, die in die funktionalen Anforderungen integriert werden würden.

Dies soll bei der Erzeugung von Anforderungen mittels Schablonen verhindert werden. Für nicht-funktionale Aspekte unterschiedlicher Natur gibt es gesonderte Schablonen aus der MASTER-Familie, die wir in Abschnitt 10.7 „Nicht-funktionale Anforderungen“ erklären.

### 10.6.3 Die Details in englischer Sprache

Da viele Projekte in englischer Sprache spezifizieren, gilt es die beiden Bestandteile <Objekt> und <Prozesswort> hinsichtlich ihrer Präzisierung und Konkretisierung auch im Englischen zu untersuchen, um einen korrekten Syntaxbauplan zu gewährleisten. Abbildung 10.12 zeigt den Detailed FunctionalMASTER in englischer Sprache.

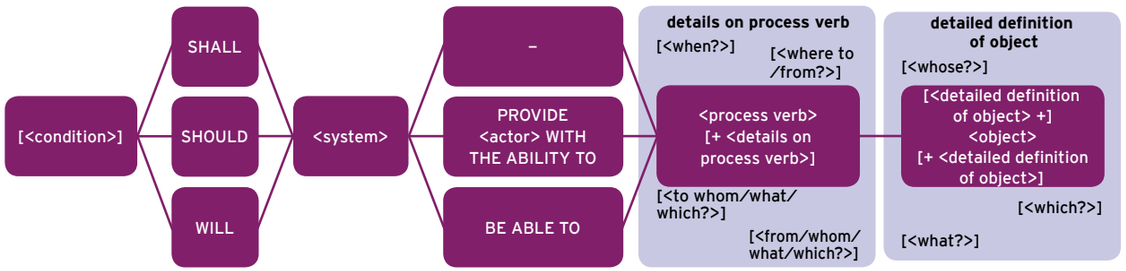



**Abbildung 10.12:** Detailed FunctionalMASTER

Die Präzisierung des Objekts sowie die Konkretisierung des Prozessworts folgen grundlegend den gleichen Aspekten und Fragestellungen wie im Deutschen.

So könnte eine Anforderung nach dem Detailed FunctionalMASTER lauten.

As soon as the library system has saved the user data, the library system shall provide the librarian with the ability to print a valid user identification card of a registered library customer on a network printer.

## 10.7 Nicht-funktionale Anforderungen

Damit meinen wir einerseits "bis hierher in diesem Kapitel" und andererseits "in den Vorgängerauflagen dieses Werks".

Versuche, nicht-funktionale Anforderungen mit Hilfe der bisher bekannten Anforderungsschablonen zu formulieren, schlagen fehl. Die Begründung dafür ist einfach. Die bekannten Anforderungsschablonen wurden nur für funktionale Anforderungen entwickelt.

Um allgemein akzeptierte und korrekte Aussagen des Requirements-Engineerings wie

- „Nicht-funktionale Anforderungen sind nicht weniger wichtig als funktionale Anforderungen.“
- „Nicht-funktionale Anforderungen müssen bereits in der Analysephase ermittelt und dokumentiert werden.“
- „Nicht-funktionale Anforderungen sind in jedem Fall Bestandteil der fachlichen Anforderungsspezifikation.“

Wie Prozesswörter lassen sich auch NFAs gut wiederverwenden. Siehe dazu Kapitel 21.5.1 "Der Ansatz nach IVENA XT".

auch mit dem Ansatz der Anforderungsschablonen zu untermauern und den nicht-funktionalen Anforderungen ihren verdienten Stellenwert zukommen zu lassen, führen wir Anforderungsschablonen für nicht-funktionale Anforderungen ein.





Die sprachlichen Forschungen ergaben, dass es nicht genau einen Syntaxbauplan für alle Kategorien der nicht-funktionalen Anforderungen geben kann. Die Inhalte der Kategorien unterscheiden sich zu stark. Es war deshalb notwendig, mehrere Schablonen für nicht-funktionale Anforderungen zu entwickeln.

Die drei unterschiedlichen Schablonen für nicht-funktionale Anforderungen

| | Eigenschafts-MASTER | Umgebungs-MASTER | Prozess-MASTER |
|---|---|---|---|
| Qualitätsanforderung | X | | |
| Technologische Anforderung | X | Umgebung, Mengengerüst | |
| Benutzeroberfläche | X | | |
| Sonstige Lieferbestandteile | X | | |
| Durchzuführende Tätigkeit | X | | X |
| Rechtlich-vertragliche Anforderung | X | | X |

Hier wird die Frage beantwortet, welcher MASTER für welche Kategorie von NFAs eingesetzt wird.

Für Anforderungen zu diesen Unterkategorien klingen Sätze nach UmgebungsMASTER einfach besser als nach EigenschaftsMASTER.

**Abbildung 10.13:** Zuordnung NFA-Kategorien zu MASTER-Schablonen

Für eine ausführliche Erklärung der einzelnen Kategorien von nicht-funktionalen Anforderungen mit Beispielen verweisen wir auf Kapitel 12 „Nicht-funktionale Anforderungen“.

### 10.7.1 Eigenschaften

Wie Abbildung 10.13 zeigt, lässt sich eine große Anzahl von nicht-funktionalen Anforderungen mit Hilfe des EigenschaftsMASTER formulieren.

RE-Bauernregel: Schablonen nutzen ist recht fein, auch NFAs gehn da jetzt rein.

Auch hier wird ein Schlüsselwort für die rechtliche Verbindlichkeit gefordert, wie aus Abschnitt 10.4 bekannt.

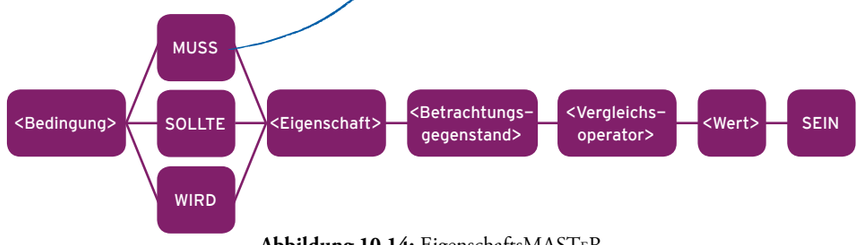



**Abbildung 10.14:** EigenschaftsMASTER

Wir beschränken uns hier und bei den nächsten beiden Schablonen aus Platzgründen auf die Variante mit Bedingung.

Der EigenschaftsMASTER in Abbildung 10.14 beinhaltet als ersten Bestandteil eine **<Bedingung>**. Benötigt man diese nicht, so ändert sich im Deutschen der Satzbau und der Bestandteil der rechtlichen Verbindlichkeit (MUSS, SOLLTE, WIRD) rutscht im Satz nach hinten zwischen die Bestandteile <Betrachtungsgegenstand> und <Vergleichsoperator>.

Eine nach dem EigenschaftsMASTER formulierte Anforderung ohne Bedingung beginnt mit einer **<Eigenschaft>** und einem **<Betrachtungsgegenstand>**. Die Eigenschaft stellt ein Merkmal dar, das den Betrachtungsgegenstand ausmacht.

Z.B. Größe oder Gewicht eines Produkts, Farbe eines Gehäuses oder von Gestaltungselementen der Benutzeroberfläche, materielle Zusammensetzung eines Hardwareteils, Dateiformat eines Reports, Dauer einer Funktion, Format des Benutzerausweises

Als Betrachtungsgegenstand, auf den sich diese Eigenschaft bezieht, können ganz unterschiedliche Umfänge gelten. Das hängt von der Spezifikationsebene oder dem fachlichen Inhalt ab, die/der spezifiziert werden soll.

Z.B. ein Gesamtsystem, ein Teilsystem, eine Komponente oder auch eine Funktion oder ein materielles Artefakt wie der Benutzerausweis.

Z.B. die RAL-Farbsammlung oder das RAL-Farbsystem für die Eigenschaft „Farbe" des Betrachtungsgegenstands „Fahrertür" eines Feuerwehrautos = RAL 3024

Stehen Betrachtungsgegenstand und dessen Eigenschaft fest, verlangt der EigenschaftsMASTER einen **<Vergleichsoperator>** und einen **<Wert>**. Der Wert stellt einen konkreten Zahlenwert mit einer dazu semantisch passenden Einheit wie Gramm, Millimeter, Sekunde, etc. dar. Genauso kann an dieser Stelle aber auch eine Wortgruppe stehen, die einen Wert repräsentiert, oder eine genormte bzw. standardisierte Größe gewählt werden.

Im EigenschaftsMASTER ist der Bestandteil <Vergleichsoperator> als zwingend auszufüllen angegeben. Dies ist derart definiert, da sich der Anforderungsautor zwingend Gedanken darüber machen soll. Für die Anwendung in der Praxis ist das obligatorische Formulieren des Vergleichsoperators in manchen Fällen jedoch dem Wohlklang der deutschen Sprache nicht förderlich. Falls eine konkrete Anforderung also sehr konstruiert klingt, so lassen Sie diesen Bestandteil auch gerne weg. Diese beiden Anforderungssätze verdeutlichen dies:

Wir finden, dass „muss weiß sein" schöner klingt als „muss gleich weiß sein."

Die Farbe des Benutzerausweises muss weiß sein. — Ohne <Vergleichsoperator>

Die Farbe des Benutzerausweises muss gleich weiß sein. — Mit <Vergleichsoperator>

Ein nach dem EigenschaftsMASTER formulierter Anforderungssatz endet mit dem Vollverb SEIN. Andere Prozesswörter, die Funktionen beschreiben (z. B. drucken, berechnen), sind an dieser Stelle unangebracht, denn der EigenschaftsMASTER beschreibt keine Funktionen. Das geschieht mit dem FunktionsMASTER aus Abschnitt 10.3 „Schritt für Schritt zur Anforderung".

Die Erfahrung zeigt, dass nicht-funktionale Anforderungen oft nicht testbar formuliert sind – also dem Qualitätskriterium Prüfbar nicht entsprechen. Dies liegt in den häufigsten Fällen daran, dass keine konkreten Werte angegeben werden. Der EigenschaftsMASTER fordert diesen Wert jedoch explizit, so dass die Prüfbarkeit der Anforderung durch Nutzung des EigenschaftsMASTER leicht gewährleistet werden kann.





## 10.7.2 Umgebungen und Kontext

Die nächste Anforderungsschablone – der UmgebungsMASTER – wurde konzipiert, um nicht-funktionale Anforderungen aus der Kategorie der technologischen Anforderungen zu formulieren. Die Schablone hält Sie dazu an, über die technologischen Aspekte nachzudenken. Die visualisierte Darstellung des UmgebungsMASTER zeigt Abbildung 10.15.

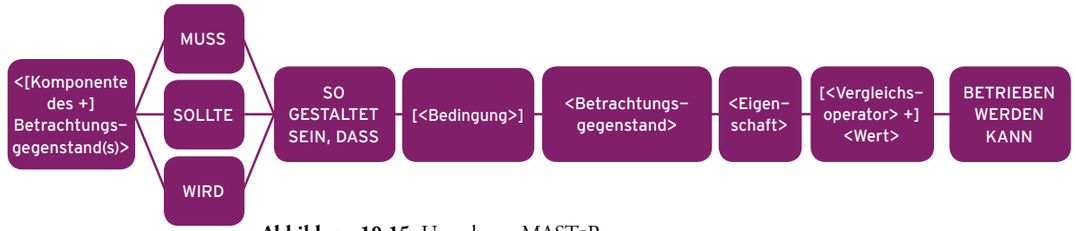



**Abbildung 10.15:** UmgebungsMASTER

Systeme müssen heutzutage eine stattliche Menge Anforderungen aus der Umgebung erfüllen. Sie müssen z. B. bei Extremtemperaturen oder unter Wasser funktionieren. In der realen Projektwelt liest sich eine Anforderung aus der Umgebung des Systems (mehr zur Umgebung des Systems in Kapitel 5 „Ziele, Informanten und Fesseln") bisher oft so:

Die Umgebungstemperatur des Bibliothekssystems muss 5 bis 50°C sein. — So sieht die Anforderung oft in der Praxis aus ...

Hier wurde der Fokus der Anforderung falsch gesetzt. Die Anforderung wurde **an die Umgebungstemperatur** gestellt und nicht **an das betrachtete System**. Der Unterschied wird mit einem Vergleich sehr gut deutlich. Dazu formulieren wir das eben genannte Beispiel nach dem UmgebungsMASTER:

Das Lesegerät des Bibliothekssystems muss so gestaltet sein, dass das Bibliothekssystem bei einer Umgebungstemperatur von 5°C bis 50°C betrieben werden kann. — ... und so nach Umgebungs-MASTER

Bei unserem Beispiel handelt es sich um eine Anforderung, die sich aus einer möglichen Umgebung des Betrachtungsgegenstands heraus ergibt. Wird das Lesegerät bei einer Temperatur von 10°C benutzt, weil die Heizung in der Bibliothek ausgefallen ist, wäre sein Besitzer sehr enttäuscht, könnte er allein aufgrund der etwas kälteren Umgebungstemperatur die Funktionen des Lesegeräts nicht verwenden.

Wie schon mit dem EigenschaftsMASTER können Sie auch mit dem UmgebungsMASTER Anforderungen an einzelne Teile eines Betrachtungsgegenstands schreiben. So wird die Anforderung genauer und die Entwicklung und der Test haben klare Anhaltspunkte für ihre Arbeit. Im Hauptsatz des UmgebungsMASTER führen wir daher den Teilbestandteil **[Komponente des +]** als fakultativen Inhalt ein. Der im Nebensatz geforderte **<Betrachtungsgegenstand>** ist der gleiche wie im Hauptsatz. Die Unterscheidung in Komponente und Betrachtungsgegenstand im UmgebungsMASTER macht die Anforderung exakter und erleichtert das Verständnis.

In unserem Beispiel das Lesegerät als ein Teil des Gesamtsystems „Bibliothekssystem"

Die restlichen Bestandteile des UmgebungsMASTER sind einerseits feststehende Formulierungen wie SO GESTALTET SEIN, DASS oder BETRIEBEN WERDEN KANN. Andererseits existieren noch weitere (optional) auszufüllende Bestandteile, die Sie jedoch bereits vom EigenschaftsMASTER kennen. Wir werden deshalb hier nicht noch einmal nä-

her darauf eingehen. Zwei weitere Anforderungsbeispiele auf Basis des UmgebungsMASTeR sollen diesen nochmals verdeutlichen:

Das Bibliothekssystem muss so gestaltet sein, dass das Bibliothekssystem mit einer Anzahl gleichzeitig surfender Gäste von bis zu 100 betrieben werden kann.

Das Bibliothekssystem muss so gestaltet sein, dass das Bibliothekssystem bei einer Spannung von 220 Volt +- 10 Volt betrieben werden kann.

### 10.7.3 Prozesse

Der ProzessMASTeR ist die dritte Anforderungsschablone für nicht-funktionale Anforderungen. Mit dieser Schablone werden spezielle Inhalte der Kategorien

- Anforderungen an durchzuführende Tätigkeiten,
- rechtlich-vertragliche Anforderungen

beschrieben. In diese Kategorien der nicht-funktionalen Anforderungen fallen neben Gesetzen oder Normen auch Anforderungen an den Auftragnehmer oder an andere Personen. Die Schablone in Abbildung 10.16 und folgende Beispielanforderung zeigen dies:

Der Auftragnehmer muss das Wartungspersonal des Bibliothekssystems einmal jährlich schulen.

Hinter dieser Anforderung verbirgt sich ein Prozess. Die Anforderung ist nicht an das betrachtete System gestellt, sondern muss neben der Entwicklung des Systems vom Auftragnehmer realisiert werden.

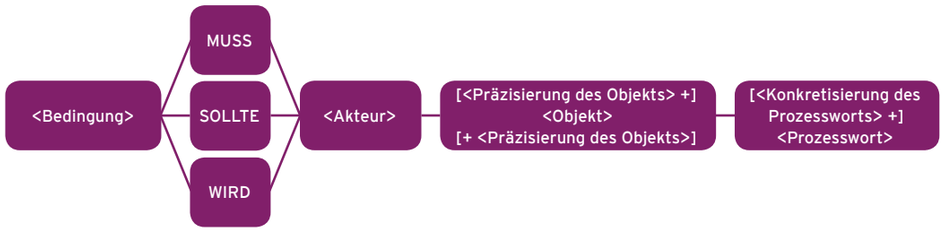



**Abbildung 10.16:** ProzessMASTeR

Der ProzessMASTeR greift auf einen Zweig des FunktionsMASTeR zurück. Der entscheidende Unterschied ist, dass hier Anforderungen an einen Akteur und nicht an ein System gestellt werden. In unserem Beispiel wird der Auftragnehmer in die Pflicht genommen. Für ihn ergibt sich aus dieser Anforderung eine Aufgabe. Da der Akteur der Schwerpunkt im ProzessMASTeR ist, wird von der Schablone im Gegensatz zu allen anderen MASTeR-Schablonen auch nicht die Angabe des Systems explizit gefordert. Sie ist fakultativ und fällt im obigen Beispiel unter die Präzisierung des Objekts.

Die sonstigen Bestandteile des ProzessMASTeR sind auch hier bereits bekannt. Rechtliche Verbindlichkeit, das Objekt und dessen Präzisierung sowie das Prozesswort und dessen Konkretisierung wurden bereits bei dem FunktionsMASTeR in den Abschnitten 10.3 „Schritt für Schritt zur Anforderung“ bis 10.6 „Details für die Konstruktion“ ausführlich besprochen.





Auch an dieser Stelle soll ein weiteres Anforderungsbeispiel auf Basis des ProzessMASTeR helfen, dessen Einsatz zu verdeutlichen.

Der Auftragnehmer muss die für die Schulungen benötigten Computersysteme inklusive der benötigten Software zur Verfügung stellen.

### 10.7.4 Konstruieren in englischer Sprache

Wie schon den FunktionsMASTeR haben wir auch die nicht-funktionalen MASTeR-Schablonen ins Englische übersetzt. Die Abbildungen 10.17 bis 10.19 zeigen diese drei Schablonen.

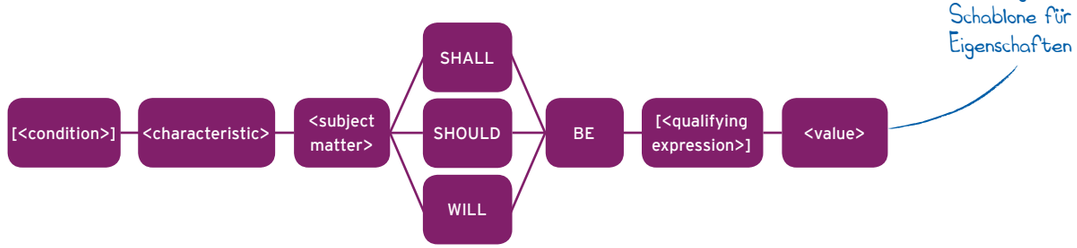



**Abbildung 10.17:** PropertyMASTeR

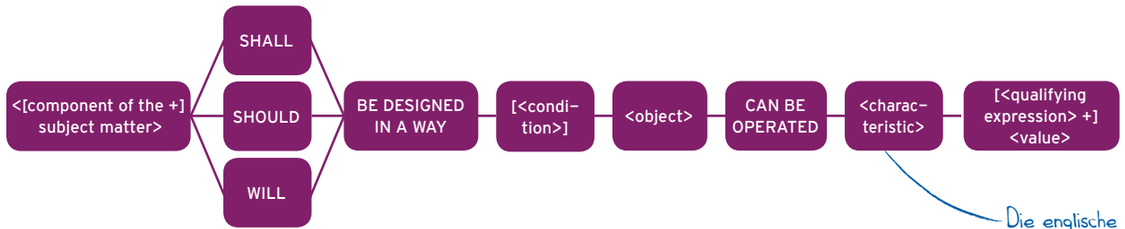



**Abbildung 10.18:** EnvironmentMASTeR

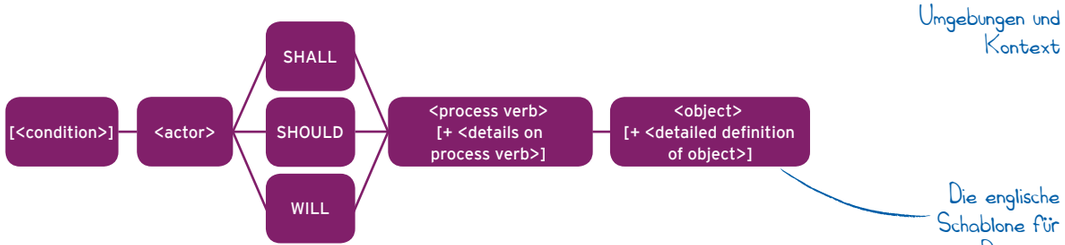



**Abbildung 10.19:** ProcessMASTeR

In den bisherigen Abschnitten dieses Kapitels wurden wiederholt Bedingungen und ihre Formulierungsmöglichkeiten erwähnt. Der nächste Abschnitt widmet sich nun genau diesen Möglichkeiten und den semantischen Unterschieden, die mit unterschiedlichen Formulierungen einhergehen.

## 10.8 Bedingungen in Anforderungen

Bedingungen sind in vielen Anforderungen immens wichtige Informationen, ohne die eine Anforderung oft nicht vollständig ist. Dies ist darin begründet, dass Anforderungen in vielen Fällen nur unter bestimmten logischen oder zeitlichen Bedingungen gelten.

Deshalb wurde bereits im FunktionsMASTER mit Bedingung (siehe Abbildung 10.4) ein Bestandteil <Bedingung> eingeführt und dort in Schritt 5 der Ausfüllanleitung kurz besprochen. Schritt 5 lautete: „Legen Sie die Bedingungen fest, unter denen die geforderte Funktionalität durchgeführt wird." Da wir nun zusätzlich Kenntnis über Anforderungsschablonen für nicht-funktionale Anforderungen erhalten haben, können wir diesen Schritt erweitern und formulieren:

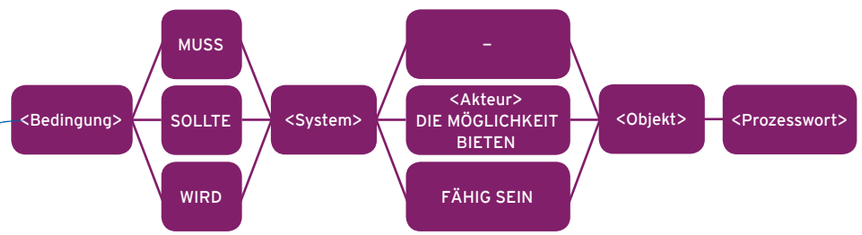
Schritt 5 wurde um nicht-funktionale Aspekte erweitert, da Bedingungen auch in NFAs auftreten können.

**Detaillierter Schritt 5:** Legen Sie die Bedingungen fest, unter denen die geforderte Funktionalität durchgeführt wird, oder unter denen der nicht-funktionale Aspekt gilt.

### 10.8.1 Syntax für und Semantik in Bedingungen

Da Bedingungen grundsätzlich optionale Bestandteile eines Anforderungssatzes sind, müssen diese nicht zwingend in jeder Anforderung beschrieben werden. In diesem Fall gilt der FunktionsMASTER ohne Bedingung (siehe Abbildung 10.2). Oft jedoch fordert der zu dokumentierende Inhalt einer Anforderung eine oder mehrere Bedingungen, unter der/denen die Anforderung ihre Gültigkeit erhält. Ohne diese Bedingungsnebensätze wäre die Anforderung unvollständig.

Der FunktionsMASTER ohne Bedingung muss demnach um den Bestandteil <Bedingung> erweitert werden und wir erhalten den FunktionsMASTER mit Bedingung, den wir hier nochmals in Erinnerung rufen.




Der Bestandteil <Bedingung> ist hier nicht [optional] gekennzeichnet, da es die beiden Varianten des FunktionsMASTER mit und ohne Bedingung gibt.

**Abbildung 10.20:** FunktionsMASTER mit Bedingung

Betrachtet man den Bestandteil **<Bedingung>** näher, so lässt sich schnell erkennen, dass es eine Vielzahl möglicher semantischer Belegungen dieses Bestandteils geben kann. Sprachliche Untersuchungen der verschiedensten inhaltlichen Varianten eines Bedingungsnebensatzes zeigen, dass zwischen drei inhaltlichen Kategorien unterschieden werden muss, die anhand definierter Konjunktionen expliziert werden. Als einleitende Konjunktionen für den Bedingungsnebensatz einer Anforderung gelten diese Signalwörter.



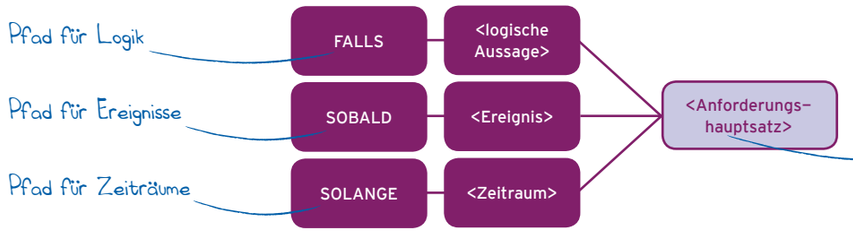


| Einleitende Konjunktion | Logische Bedingung | Zeitliche Bedingung | Bedeutung |
|---|---|---|---|
| **FALLS** | **X** | **–** | FALLS leitet eine logische Aussage ein. Umschreibungen für diese Konjunktion sind:<br>■ für den Fall<br>■ unter der Voraussetzung, dass<br>■ unter der Bedingung, dass |
| **SOBALD** | **–** | **X** | SOBALD rückt den Zeitpunkt, zu dem ein Ereignis eintritt, in den Vordergrund. Umschreibungen lauten:<br>■ in dem Augenblick<br>■ sofort<br>■ sowie |
| **SOLANGE** | **–** | **X** | SOLANGE leitet einen Bedingungssatz ein, der einen Zeitraum beschreibt. Im genannten Zeitraum laufen Bedingungsnebensatz und Anforderungshauptsatz gleichzeitig ab. SOLANGE wird umschrieben durch:<br>■ währenddessen |

Wir empfehlen, das Signalwort WENN nicht zu verwenden. Es macht den semantischen Unterschied der Bedingungsinhalte nicht deutlich.

**Abbildung 10.21:** Signalwörter für Bedingungen

Zunächst haben wir das Augenmerk auf die einleitenden Konjunktionen gelegt, um so die unterschiedliche Semantik der Bedingungen herauszuarbeiten. Diese Unterscheidung wirkt sich in dem BedingungsMASTER in drei Pfaden aus. Der BedingungsMASTER beleuchtet den einzelnen Bestandteil **<Bedingung>** des FunktionsMASTER mit Bedingung aus Abbildung 10.20 detailliert und zeigt die drei Pfade für logische Aussagen, Aussagen zu Zeitpunkten oder Ereignissen und Aussagen zu Zeiträumen.




Hier folgt der Hauptsatz - die eigentliche Anforderung - nach FunktionsMASTER.

**Abbildung 10.22:** BedingungsMASTER

Beispiele für Bedingungsnebensätze der einzelnen drei Pfade aus dem Bibliothekssystem könnten so lauten:

**Falls** das Alter eines Kunden kleiner als 18 Jahre ist, muss …

**Sobald** sich ein Kunde authentifiziert hat, muss …

**Solange** das Bibliothekssystem Bestandsdaten empfängt, muss ...

Bedingungsstrukturen in einem Anforderungssatz lassen sich beliebig kompliziert aufbauen.

Es ist nicht vorgegeben, dass es nur einen Bedingungsnebensatz pro Anforderungssatz geben darf. Sofern benötigt, können einzelne Bedingungen auch mittels der logischen Operatoren AND (UND) OR (ODER) und XOR (exklusives ODER) verbunden oder durch das Setzen von Klammern in jedweder Variante geschachtelt und logisch verknüpft werden. Achten Sie dabei nur darauf, dass die Anforderung als solche verständlich bleibt. Deutlich wird die Problematik anhand der folgenden Ausgangsanforderung.

Arbeiten Sie bei komplizierten Bedingungsstrukturen gerne mit Zeilenumbrüchen, Entscheidungstabellen oder anderen Strukturierungsmitteln, um die Lesbarkeit Ihrer Anforderung zu gewährleisten.

An jedem Dienstag um 8:00 Uhr oder an jedem Freitag um 9:00 Uhr und unter der Bedingung, dass mindestens zehn Neukunden seit der letzten Übertragung der Neukundenstatistik im Bibliothekssystem aufgenommen wurden, muss das Bibliothekssystem die Auswertungsdaten der Neukundenstatistik an den zentralen Administrationsrechner automatisch übertragen.

Ausgangsanforderung

Es ist in dieser Anforderung nicht klar, ob die Bedingung, dass zehn Neukunden aufgenommen wurden, sich nur auf die Übertragung am Freitag oder auch auf die Übertragung am Dienstag bezieht. Zudem ist nicht eindeutig beschrieben, ob das System die Daten sowohl am Dienstag als auch am Freitag übertragen kann (OR) oder ob die Daten nur an einem einzigen Tag der Woche übertragen werden – dann entweder am Dienstag oder am Freitag (XOR). Durch die Verwendung von mathematisch definierten logischen Operatoren wird die Anforderung bezüglich der Vorbedingung eindeutig.

Verbesserte Anforderung

(Sobald der Zeitpunkt [Wochentag = Dienstag UND Tageszeit = 08:00 MEZ] erreicht ist
ODER
sobald der Zeitpunkt [Wochentag = Freitag UND Tageszeit = 09:00 MEZ] erreicht ist)
UND
falls mindestens zehn neue Kunden seit der letzten Übertragung der Neukundenstatistik automatisch im Bibliothekssystem angelegt wurden,
muss das Bibliothekssystem die Auswertungsdaten der Neukundenstatistik automatisch an den zentralen Administrationsrechner übertragen.

Für die bessere Lesbarkeit haben wir Zeilenumbrüche integriert.

Die Bestandteile **<logische Aussage>**, **<Ereignis>** und **<Zeitraum>** betrachtet der dargestellte BedingungsMASTER nicht noch detaillierter. Es existieren hierzu dennoch weitere detaillierte Schablonen, die zusätzliche inhaltliche Pfade für diese Bestandteile definieren. Für den interessierten Leser sind diese unter www.sophist.de/re6/kapitel10 zu finden.

### 10.8.2 Konstruieren in englischer Sprache

Wie bei den bereits vorgestellten Anforderungsschablonen für funktionale Anforderungen und für nicht-funktionale Anforderungen gibt es den BedingungsMASTER in der englischen Übersetzung. Den ConditionMASTER zeigt folgende Abbildung 10.23.

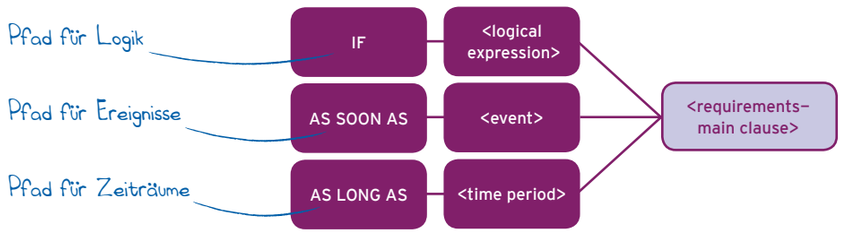



**Abbildung 10.23:** ConditionMASTER





Die einleitenden Konjunktionen heißen hier IF, AS SOON AS und AS LONG AS. Ebenso wie in der deutschen Sprache wurden die detaillierten Schablonen für die Bestandteile **<logical expression>**, **<event>** und **<time period>** genauer untersucht. Sie stehen unter www.sophist.de/re6/kapitel10 zur Verfügung.

## 10.9 Auf die Sätze, fertig, los!

Die Einführung funktioniert zu Beginn eines Projekts wie auch während der Projektlaufzeit.

Setzen Sie die Schablonentechnik in erster Linie dann ein, wenn sich in Ihren Projekten die Bereitschaft der Mitarbeiter herauskristallisiert, die Anforderungsschablonen anzuwenden. Neben der reinen Methodenkenntnis müssen Ihre Anforderungsautoren bereit sein, sich einer stark normierten Vorgehensweise zu unterwerfen.

Sie können die Schablonen aber auch im Verborgenen einführen, indem Sie „nebenbei" Anforderungen nach und nach umformulieren.

Ihre stilistischen Freiheitsgrade bei der Formulierung von Anforderungen werden dabei stark eingeschränkt. Den besten Erfolg erzielten wir, wenn wir die Anforderungsschablonen nicht als ein Muss vorgeschrieben haben, sondern die Methode geschult und die Schablone als Hilfsmittel dargestellt haben. Müssen sie eine Fremdsprache verwenden, sind die meisten Projektmitarbeiter für Templates, Glossare und möglichst klare Vorgaben dankbar.

Falls Ihnen an höchster Qualität der Anforderungen gelegen ist, führen Sie die Anforderungsschablonen ein!

Die Schablonen funktionieren nicht nur für IT oder Software. Sie können damit auch im Bereich Hardware, Elektronik oder Mechanik Anforderungen dokumentieren.

Für die Einführung der Schablonen integrieren Sie Arbeitsanweisungen und Erklärungen zum Umgang mit den Schablonen in Ihrem RE-Leitfaden. Dort können Ihre Projektmitarbeiter nachlesen oder Selbststudium betreiben, um den Umgang selbst zu erlernen. Führen Sie Workshops durch, in denen Sie das Schreiben nach Schablone üben lassen, etc. Mehr zur Einführung und zum Leitfaden in Kapitel 22 „Einführungsstrategien".

Gute Erfahrungen mit den Schablonen machen wir bei Personen, die Neulinge im RE sind.

Sie können hier ohne Weiteres 20 bis 50 Anforderungen schreiben lassen.

Bei der Qualitätssicherung der Anforderungen sollten Sie Folgendes bedenken: Das Lesen von Dutzenden gleich strukturierten Anforderungen stellt eine gewisse Hürde dar, ist ermüdend und erfordert eine Menge Konzentration. Erwarten Sie nicht, dass jemand das gesamte Anforderungsdokument liest. Strukturieren Sie Ihr Dokument so, dass unterschiedliche Leser ohne großen Suchaufwand an genau die Stellen gelangen, die für sie von Interesse sind.

Das Lesen schablonenbasierter Anforderungen ist nach Schäfchenzählen die beste Einschlafmethode.

Bauen Sie Ihre Anforderungen und Definitionen ab heute einfach nach wenigen einfachen Regeln zusammen. Es ist kinderleicht!

Schablonenbasierte Anforderungen nutzen Sie am besten ab Detaillierungsebene 2 und darunter. Falls Sie jedoch nur bis Level 1 beschreiben, dann auch auf Level 1.

Auch bei der Ermittlung von Anforderungen leisten die Anforderungsschablonen bzw. das Wissen darüber sehr gute Dienste. Denn das Wissen über die einzelnen notwendigen Bestandteile eines Anforderungssatzes hilft enorm dabei, die richtigen Fragen in Ermittlungsworkshops bzw. beim Einsatz von Befragungstechniken zu stellen. Sie können somit ganz zielgerichtet Ihre Fragen formulieren und erhalten die wichtigen Informationen, um eine gute Anforderung zu erzeugen.

Eine weitere wichtige Erkenntnis aus unserer Projekterfahrung: Nutzen Sie technische Unterstützung, das erleichtert die Arbeit. Die Verwendung von Anforderungsschablonen und eines begrenzten, definierten Wortschatzes lässt sich in Requirements-Management-Tools durch Assistenten/Wizards realisieren. Dadurch, dass sich die meisten Werkzeuge, die für das Dokumentieren und Verwalten von Anforderungen genutzt werden, anpassen lassen, können Sie das „Zusammenbauen" von Anforderungen in einem gewissen Maß automatisieren.

Als abschließenden Tipp zu diesem Thema möchten wir Ihnen noch Folgendes mitgeben: Ob in klassischen oder agilen Vorgehensweisen, in Mischformen oder in „wie-auch-immergenannten"-Vorgehen/Prozessen/Methoden – nutzen Sie die Grundidee der Anforderungsschablonen für die Dokumentation von „Anforderungen" jeglichen Inhalts und jeglicher Ebene oder Art, sofern Sie in Textform dokumentieren.

Die Grundidee ist: Formalisieren Sie Sprache bis zu einem gewissen Grad, machen Sie sich Gedanken über wichtige Bestandteile und wie diese zusammenhängen. Dies führt immer zu einer Art von Syntaxbauplan. Ob für Anforderung, User-Story, Testfall, Use-Case oder, oder, oder. Es funktioniert überall und Sie profitieren von den in diesem Kapitel besprochenen Vorteilen.

Für alle Leserinnen und Leser, die nun noch nicht genug von den Anforderungsschablonen haben, stellen wir weiterführende Literatur zum Thema in Form einer Broschüre mit dem Titel „MASTER – Schablonen für alle Fälle" bereit. Diese finden Sie unter www.sophist.de/re6/kapitel10.

## Applying Requirements Templates in Practice: Lessons Learned

*Alistair Mavin, Rolls-Royce PLC*
*Alistair Mavin is a Requirements Specialist at Rolls-Royce PLC and is the lead author of the Easy Approach to Requirements Syntax (EARS) natural language requirement template [1, 2, 3]. EARS and MASTER were generated independently, but both build on similar work and share the same basic principles. EARS is a single template, the application of which produces requirements in a small number of patterns. In this Expert Box, Alistair shares some insights and lessons learned from applying this type of natural language template in practice.*

The beauty of a simple natural language (NL) template is that it can be applied by anyone with no tool support. Whilst many requirement notations rely on the use of expensive specialist tools, a simple NL template can be used on the back of a cigarette packet, in a simple application such Word or Excel, or with dedicated tool support. This makes templates a very cost-effective improvement to requirements practice.





The simplicity of these templates means that they can be learned in an hour and people will immediately write better requirements. This seems to be true for both novices and experienced requirements authors. Indeed, I have received a lot of positive feedback from experienced authors, who will often revisit and improve an existing set of requirements once they start using the template.
To help embed the learning, some coaching and support is beneficial. I encourage first-time users of the template to write 10 requirements and then let me review them. With the review comments in mind, I ask them to write another 10. Using this type of approach, any systematic errors can easily be avoided so that when a full document is produced it will contain less requirements errors. This also makes document review easier and reduces the need for rework.
In practice, requirements are almost always written iteratively. Authors will usually start with a subject about which a requirement may be needed. For example, for an aero engine, subjects could include Noise, Vibration, Temperature, Weight and Reliability. Authors will use these seeds to draft and then improve requirements, potentially several times as development proceeds. It is important that stakeholders understand this need for iteration and do not expect to get every requirement right first time.
Once using a template, authors will begin to use the style and language of the template, automatically fitting emerging requirements into the template structure without even realising it. Surprisingly quickly, using a template will improve the quality of first draft requirements. A well-formulated template therefore serves to enhance the capability of requirement authors.
The structure of a requirement written with a template tends to help with requirements decomposition. This is because the clauses of a system requirement will often suggest the need for an element of functionality that has to be provided at a sub-system level. For example, where a system level requirement is triggered by a change in temperature, there must be some means of detecting that temperature change. This will yield a derived requirement at sub-system or component level to measure temperature.
NL templates provide guidance for writing each individual requirement more clearly, but they can also help with requirements coverage. One example is in requirements pairing, such as where a requirement for wanted behaviour will naturally imply the need for some requirements to cover associated unwanted behaviour. Similarly when a requirement is written for normal operation, authors will be likely to consider the behaviour required in back-up operation. Examples for an aero engine would be a requirement for engine dedicated generator power supply and a corresponding requirement for aircraft back-up power supply.
Scenarios are a useful mechanism for grouping related requirements into logical chunks of functionality. NL templates work particularly well with scenarios, which also helps to ensure requirements coverage. Scenarios can be used during requirements discovery, as the basis for structuring a requirements document and during system test [4].
Simple NL templates obey one of my key rules for stakeholder engagement; to keep things as simple as possible. Whenever I introduce an approach I do so in small steps, without explicitly labelling it as a new technique. For example, I would avoid the use of terms such as syntax or swimlane diagram, which may not be commonly understood. Instead I would use more everyday language like style or flow chart. Over time, more terms can be introduced, adding rigour to techniques and models, which will incrementally build stakeholders' knowledge and capability.
Human beings are naturally resistant to change which can hinder new initiatives.

Therefore in my experience it can be best to introduce an NL template by stealth. A simple NL template is a lightweight improvement that can be introduced without an official launch. When introduced quietly and allowed to grow and spread by word of mouth, an NL template can significantly improve requirements practices within an organisation.
The purpose of a requirements document is not to be an entertaining read. The purpose is to make it as easy as possible for the reader to understand what the system must do. If a requirement must be read several times and the meaning is still unclear, then it is a poorly written requirement. One key benefit of NL templates is to drive consistency into requirements. This makes each requirement more alike and therefore easier to understand, since humans are very good at pattern recognition. However, this consistency makes reading a requirements document rather repetitive. I do not see this as a problem, since I doubt that anyone has ever read a requirements document for pleasure.
We developed our template at Rolls-Royce because the majority of people write most of their requirements in NL. However, some requirements are better expressed in other formats, such as truth tables, graphics, mathematical formulas or Boolean algebra. It can actually be counter-productive to write these requirements using NL, even with the aid of a template. So, my final piece of advice is to accept and embrace some diversity of requirements expression. A document may include 95% NL requirements, but the other 5% will be consciously documented using other more suitable notations.

*1 "Easy Approach to Requirements Syntax (EARS)", Mavin, A., Wilkinson, P., Harwood, A. and Novak, M., Proceedings of RE09, IEEE, August 2009.*
*2 "BIG EARS (The Return of Easy Approach to Requirements Syntax)", Mavin, A. and Wilkinson, P., Proceedings of RE10, IEEE, September 2010.*
*3 "Listen, then use EARS", Mavin, A., IEEE Software, IEEE, March 2012.*
*4 "Scenarios, Stories, Use Cases - Through the Systems Development Life-Cycle", Alexander, I & Maiden, N (eds), Wiley, 2004.*

*Herr Büchle und Katharina sind begeistert. Nachdem Ramona und Ralph ihnen erklärt hatten, wie das mit den Schablonen funktioniert, haben sie in einem Workshop gemeinsam ihre ersten Anforderungen in natürlicher Sprache geschrieben. „Eigentlich ist das ja wirklich ganz einfach, wenn man's mal ein paarmal gemacht hat", schwärmt Katharina. Mittlerweile notieren Herr Büchle und Katharina bereits selbstständig Anforderungen, wenn ihnen so unter der Zeit etwas Neues einfällt, was sie noch vergessen hatten. Sie sind also nicht mehr ständig auf Ramona und Ralph angewiesen. „Auch wenn man sich mit diesen Schablonen etwas mehr Gedanken machen muss, um den Satz vollständig aufschreiben zu können – ich habe mich schon daran gewöhnt, so zu denken. Und Ramona und Ralph gefällt es sehr, wenn ich ihnen so „schöne" Anforderungen bringe", meint Herr Büchle.*