Die SOPHISTen

»Schablonen für alle Fälle«

SOPHIST

Die SOPHISTen

SOPHIST GmbH
Vordere Cramergasse 13
90478 Nürnberg
Deutschland
www.sophist.de

5. Auflage 2019

Copy Editing & Herstellung: Roland Kluge, SOPHIST GmbH
Umschlaggestaltung und Layout: Heike Baumgärtner, Büro Hochweiss
Druck: Flyeralarm

Copyright SOPHIST GmbH

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung der SOPHIST GmbH urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.
Es wird darauf hingewiesen, dass die in der Broschüre verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.
Alle Angaben und Programme in dieser Broschüre wurden mit größter Sorgfalt kontrolliert. Weder Autoren noch die Firma SOPHIST GmbH, etc. können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieser Broschüre stehen.

Die SOPHISTen

»Schablonen für alle Fälle«

www.sophist.de

Inhaltsverzeichnis

1. MASTeR – Schablonen für alle Fälle ..... Seite 6

1.1 Einleitung und Motivation ..... Seite 6
1.2 Schöner Schreiben mit Schablone ..... Seite 7
1.3 Inhalt ..... Seite 8

2. Funktionale Anforderungen ..... Seite 10

2.1 FunktionsMASTeR ..... Seite 11
2.2 Detaillierter FunktionsMASTeR ..... Seite 17
2.3 Englische Versionen ..... Seite 19

3. Nicht-funktionale Anforderungen ..... Seite 21

3.1 Überblick ..... Seite 21
3.2 Die Kategorien nicht-funktionaler Anforderungen im Überblick ..... Seite 23
3.3 EigenschaftsMASTeR ..... Seite 27
3.4 UmgebungsMASTeR ..... Seite 30
3.5 ProzessMASTeR ..... Seite 34
3.6 Englische Versionen ..... Seite 36

4. Bedingungen ..... Seite 38

4.1 Verwendung ..... Seite 38
4.2 Detaillierungsebenen ..... Seite 39
4.3 BedingungsMASTeR ..... Seite 40
4.3.1 LogikMASTeR ..... Seite 42
4.3.2 EreignisMASTeR ..... Seite 44
4.3.3 ZeitraumMASTeR ..... Seite 45
4.4 Englische Versionen ..... Seite 47

5. Quellenverzeichnis ..... Seite 49

1. MASTeR – Schablonen für alle Fälle

1.1 Einleitung und Motivation

Seit Mitte der 90er Jahre sind wir SOPHISTen als Berater und Trainer in den unterschiedlichsten Firmen und Domänen tätig. Schwerpunktmäßig unterstützen wir unsere Kunden beim Requirements-Engineering (RE).

Von den vier Haupttätigkeiten des RE – Ermitteln, Dokumentieren, Prüfen und Abstimmen sowie Verwalten von Anforderungen – stellt insbesondere die Dokumentation funktionaler und nicht-funktionaler Anforderungen eine Herausforderung dar.
Zur schriftlichen Dokumentation von funktionalen Anforderungen entwickelten und veröffentlichten die SOPHISTen Ende der 90er Jahre eine Satzschablone.
Getreu dem Motto unserer griechischen Namenspatronen

Die Infragestellung des Althergebrachten führt zur Suche nach etwas Besserem.

haben wir diese Satzschablone auf den Prüfstand gestellt.

Im Rahmen unseres Innovationsprojekts **MASTeR** (**M**ustergültige **A**nforderungen - die **S**OPHIST **Te**mplates für **R**equirements) haben wir uns der Herausforderung gestellt, die natürlichsprachliche Dokumentation von Anforderungen zu verbessern. Die Ergebnisse dieses Projekts stellen wir Ihnen in dieser Broschüre vor: Lernen Sie ein Werkzeug für die schriftliche Dokumentation von Anforderungen kennen und werden Sie „MASTeR of the Requirements Universe".

Dass wir Ihnen auf den folgenden Seiten die Ergebnisse unseres MASTeR-Projekts vorstellen können, ist vielen fleißigen Personen zu verdanken. Ein herzlicher Dank für viel Engagement an alle Mitglieder des MASTeR-Projekts sowie an alle weiteren SOPHISTen, die an der Erarbeitung der Ergebnisse beteiligt waren:

Rainer Joppich	Verantwortlich für die inhaltliche Überarbeitung der Schablonen für funktionale Anforderungen;
Dr. André Pflüger	Verantwortlich für Projektleitung, Projektkoordination und inhaltliche Mitarbeit;
Dr. Stefan Queins	Verantwortlich für die inhaltliche Erarbeitung der Schablonen für nicht-funktionale Anforderungen;
Chris Rupp	Verantwortlich für die strategische Ausrichtung des Projekts;
Kristina Schöne	Verantwortlich für Übersetzungen und Veröffentlichungen;
Achim Stuy	Inhaltliche Mitarbeit bei der Überarbeitung der Schablonen;
Alexander Vöge	Inhaltliche Mitarbeit bei der Überarbeitung der Schablonen.

Ein extra Dankeschön an alle SOPHISTen, die in sonstiger Weise zum Erfolg des Projekts bei-

getragen haben, sei es für den kritischen Blick in Reviews, für fachliche Diskussionen, für die Erstellung und das Layout von Veröffentlichungen oder für alle weiteren Tätigkeiten, die notwendig sind, um ein solches Projekt durchzuführen.

1.2 Schöner Schreiben mit Schablone

Die Arbeit mit einer Satzschablone für Anforderungen trägt ihren Teil dazu bei, dass Anforderungssätze einheitlich strukturiert sind und so gewisse Defekte (wie z. B. das Fehlen des Akteurs) von vornherein vermieden werden.

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

Durch die Vorgabe der Syntax können Sie Anforderungen in einem optimalen Kosten-Nutzen-Verhältnis formulieren. Dabei legen wir besonderen Wert auf eine intuitive und einfache (d. h. eine dem natürlichen Sprachgefühl entsprechende) Handhabung jeder Schablone.

Jedoch sind auch nach einer Schablone geschriebene Anforderungen nicht gänzlich von sprachlichen Effekten befreit (Tilgung, Verzerrung oder Generalisierung von Informationen, vgl. [Rupp14]). Mit den MASTeR-Schablonen arbeiten Sie allerdings mit einem Werkzeug, das Ihnen dabei hilft, einige sprachliche Effekte zu vermeiden. Implizit wenden Sie mit den MASTeR-Schablonen einige grundlegende Regeln für das Schreiben von Anforderungen sowie so manche Regel des SOPHIST-*RE*gelwerks an vgl. [Rupp14]:

- *Es werden stets Anforderungen im Aktiv geschrieben.*
- *Es werden stets Anforderungen als vollständige Sätze geschrieben.*
- *Die Anforderungen bilden Prozesse mit Hilfe von Vollverben ab.*
- *Für jedes Prozesswort wird genau ein Anforderungssatz geschrieben.*
- *Die Detaillierungsmöglichkeiten des FunktionsMASTeRs helfen bei der Analyse fehlender Informationen zum Prozesswort.*

Der erfahrene Requirements-Engineer mag nun die Frage stellen, ob es neben der SOPHIST-Satzschablone für funktionale Anforderungen auch Schablonen für nicht-funktionale Anforderungen und Bedingungen, die sowohl funktionale als auch nicht-funktionale Anforderungen erweitern können, gibt. Zunächst ist festzustellen, dass gegenwärtig auf dem RE-Markt nur wenige Satzsschablonen speziell für nicht-funktionale Anforderungen existieren [Botella98] [Mavin10]. Diese sind allerdings nicht für eine natürlichsprachliche Dokumentation geeignet [Botella98] oder vermischen funktionale und nicht-funktionale Inhalte miteinander [Mavin10]. Bedingungen werden in verschiedenen Satzschablonen für funktionale Anforderungen berücksichtigt. Dies reicht von der Formulierung auf Programmierebene durch if-Statements [Esser07] über die Berücksichtigung von zeitlichen Bedingungen [Mavin10]

bis hin zur klaren Unterscheidung zwischen logischen und zeitlichen Bedingungen [Holtmann10]. Über die Vorgabe von Konjunktionen gehen die Ansätze allerdings nicht hinaus. Für die Syntax des Bedingungssatzes gibt es keine Empfehlungen.
Vor diesem Hintergrund sind so neben den MASTER-Schablonen für funktionale Anforderungen auch MASTER-Schablonen für nicht-funktionale Anforderungen entstanden. Mit einem weiteren Schablonensatz, dem BedingungsMASTER, ist es nunmehr möglich, Bedingungen nach Schablone zu schreiben. Diese Schablonen werden Sie aber nicht zwingend verwenden, denn nicht jede funktionale oder nicht-funktionale Anforderung (egal, ob sie nach Schablone geschrieben ist oder nicht) ist an eine Bedingung gebunden.

1.3 Inhalt

In den folgenden drei Kapiteln dieser Broschüre stellen wir Ihnen die erwähnten MASTER-Schablonen vor.

- Kapitel 2 thematisiert den FunktionsMASTER sowie seine Detaillierungsmöglichkeiten. Mit dieser Schablone können funktionale Anforderungen durchdacht und ausführlich formuliert werden.

- Nicht-funktionale Anforderungen sind der Schwerpunkt des dritten Kapitels. Um die MASTER-Schablonen nutzen zu können, ist das Verständnis und die Abgrenzung der Kategorien der nicht-funktionalen Anforderungen notwendig. Aus diesem Grund ist den Schablonen eine Übersicht über die nicht-funktionalen Anforderungskategorien vorangestellt.

- Die Schablonen des vierten Kapitels unterscheiden sich syntaktisch von den Schablonen der Kapitel 2 und 3: Mit den drei BedingungsMASTER werden Schablonen für Nebensätze eingeführt. Sie können optional funktionale wie auch nicht-funktionale Anforderungen ergänzen. Als Ergänzungsmöglichkeit stellen wir Ihnen im vierten Kapitel drei Konjunktionen vor, die die häufig verwendete Konjunktion *WENN* ablösen und eine Bedingung einleiten können. Jede der drei BedingungsMASTER-Schablonen verwendet ein anderes Schlüsselwort (Konjunktion), um mehrdeutige Bedingungen zu verhindern. Den sich daran anschließenden Bedingungssatz können Sie frei formulieren oder auf die Formulierungsvorschläge der drei BedingungsMASTER-Schablonen zurückgreifen.

Um die Ausführungen zu den einzelnen MASTER-Schablonen zu verdeutlichen, werden in jedem Kapitel Beispielanforderungen verwendet. Dabei greifen wir auf verschiedene Beispielsysteme zurück.

Folgenden Hinweis möchten wir Ihnen mit auf den Weg geben:
Bei den Anforderungen in dieser Broschüre handelt es sich um Einzelbeispiele. Anforderungen, die für sich alleine stehen, sind nur selten eindeutig und vollständig. Mehrdeutigkeiten lösen sich häufig erst durch weitere Anforderungen oder Glossareinträge auf. Diese Broschüre thematisiert allerdings vordergründig die Einführung der neuen MASTeR-Schablonen und verdeutlicht den Inhalt ihrer Satzpositionen mit Beispielen. So sind die Einzelbeispiele zum Teil sehr generisch formuliert und würden in einer Spezifikation auf verschiedenen Ebenen weiter verfeinert werden.

2. Funktionale Anforderungen

Im letzten Jahrzehnt hat sich die SOPHIST-Satzschablone zum Schreiben funktionaler Anforderungen auf dem RE-Markt etabliert. Dank unserer vielseitigen Praxiserfahrung und des ständigen Austausches mit unseren Kunden haben wir diese Schablone bearbeiten und verbessern können. Welche Bestandteile der etablierten Schablone bei der Überarbeitung für uns von besonderem Interesse waren, zeigt die Abbildung 01.

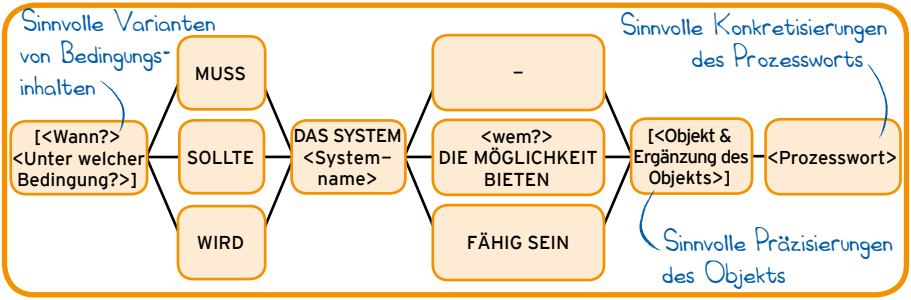


Abbildung 01: Gegenstand der Schablonen-Überarbeitung

▶▶ Als Richtlinie für die Überarbeitung dienten die folgenden Fragen:

- Welche Formulierungsmöglichkeiten für Informationen zu *Objekt* und *Prozesswort* bietet die deutsche Sprache?
- Welche inhaltliche Vertiefung im Kontext von Anforderungen ist sinnvoll?
- Welche Pfade bzw. Umformulierungen ergeben sich aus den Antworten auf diese zwei Fragen für die SOPHIST-Satzschablone?
- Welche Formulierungsmöglichkeiten für Bedingungen gibt es? Welche Rolle spielt dabei die einleitende Konjunktion?

Obwohl mit der Schablone der Satzbau einer Anforderung vorgegeben ist, gibt es verschiedene sprachliche Möglichkeiten, die Schablone zu „bestücken". Die Antworten auf die vier zentralen Fragen lassen Sie die semantische Vielfalt in Anforderungen erahnen.
Diese Erkenntnis ist jedoch kein Freifahrtschein dafür, schlichtweg jede Information, an die Sie kommen, in Form einer Anforderung zu dokumentieren. Hinterfragen Sie gesammelte Informationen, denn nicht jede wird für Ihre Spezifikation relevant sein.

Ziel des Schreibens von Anforderungen nach Schablone sind sowohl syntaktisch als auch semantisch sinnvolle Anforderungen. Dieses Ziel verfolgt der Funktions-MASTeR für das Schreiben funktionaler Anforderungen. Kapitel 2 thematisiert seine Satzpositionen und zeigt darüber hinaus, wie funktionale Anforderungen detailliert ausgedrückt werden können. Schließlich endet Kapitel 2 mit der Einführung des englischen FunktionsMASTeRs und des detaillierten FunktionsMASTeRs.

2.1 FunktionsMASTeR

Der IREB e.V. (International Requirements Engineering Board) definiert eine funktionale Anforderung folgendermaßen:

„Eine funktionale Anforderung ist eine Anforderung bezüglich eines Ergebnisses oder eines Verhaltens, das von einer Funktion eines Systems bereitgestellt werden soll.“ [Pohl11]

Mit der Schablone FunktionsMASTeR können diese Anforderungen mit geringem Aufwand formuliert werden (vgl. Abb. 02).

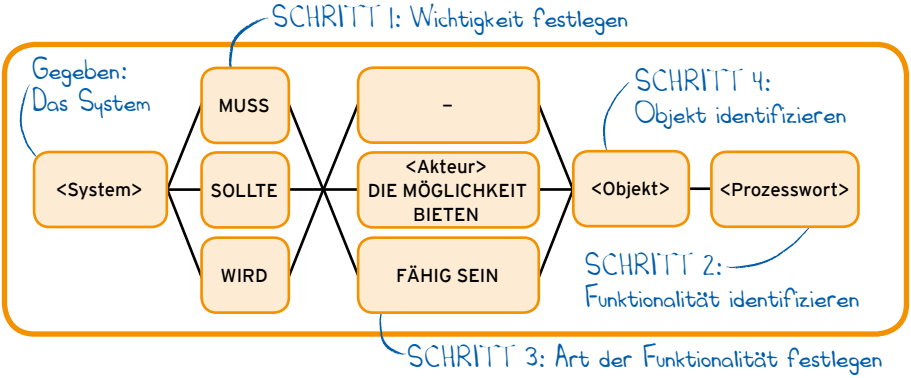


Abbildung 02: FunktionsMASTeR

Im Folgenden vermitteln wir Ihnen Hintergrundwissen zum Ausfüllen der einzelnen Positionen der Schablone. Anhand der Schritte 1-4 aus Abbildung 02 füllen wir nacheinander die einzelnen Bestandteile der Schablone mit Inhalt und gelangen somit zu einem vollständigen Anforderungssatz.

Gegeben: Das System

So einfach es auch scheinen mag – es ist entscheidend, dass Sie das zu betrachtende System beim Namen nennen. Sie geben dem Leser Ihrer Spezifikation auf diese einfach Art und Weise eine Lese- und Verständnishilfe mit auf den Weg.

Schritt 1: Die rechtliche Verbindlichkeit

In der RE-Praxis weisen unterschiedliche Stakeholder ein und derselben Anforderung eine unterschiedliche juristische Verbindlichkeit zu. Gerade im Zusammenhang mit Verträgen ist diese Wertung von entscheidender Bedeutung. Ist die rechtliche Verbindlichkeit festgelegt, kann eine Anforderung bei Nichtumsetzung auch eingeklagt werden.

Wir empfehlen daher, folgende drei Schlüsselwörter zu benutzen:

MUSS ***SOLLTE*** ***WIRD***

Diese Schlüsselwörter finden sich auch im FunktionsMASTeR zur Abbildung der rechtlichen Verbindlichkeit einer funktionalen Anforderung wieder.

- *MUSS:* Alle Anforderungen, die mit *MUSS* formuliert sind, sind verpflichtend in der Umsetzung. Die Abnahme eines Produkts kann verweigert werden, sollte das System einer *MUSS*-Anforderung nicht entsprechen.

- *SOLLTE:* Formulierungen mit *SOLLTE* stellen einen Wunsch eines Stakeholders dar. Sie sind nicht verpflichtend und müssen nicht erfüllt werden. Allerdings erhöht ihre Umsetzung die Zufriedenheit der Stakeholder und ihre Dokumentation verbessert die Zusammenarbeit und Kommunikation zwischen Stakeholdern und Entwicklern/Auftragnehmern.

- *WIRD:* Mit *WIRD* dokumentieren Sie die Absicht eines Stakeholders. Eine mit *WIRD* formulierte Anforderung dient als Vorbereitung für eine in der Zukunft liegende Integration einer Funktion. Sie ist verpflichtend in der Umsetzung zu berücksichtigen, auch wenn ihre Realisierung zunächst nicht getestet wird.

Halten Sie die Schlüsselwörter Ihrer Spezifikation schriftlich fest und klären Sie deren Bedeutung bzw. rechtliche Verbindlichkeit mit allen Projektbeteiligten. Ein sinnvoller Ort dafür ist das Glossar Ihrer Spezifikation. Stellen Sie sicher, dass es mit allen vereinbarten Definitionen

von jedem Leser und Bearbeiter der Spezifikation eingesehen werden kann. So wirken Sie Fehlinterpretationen der Anforderungen entgegen.

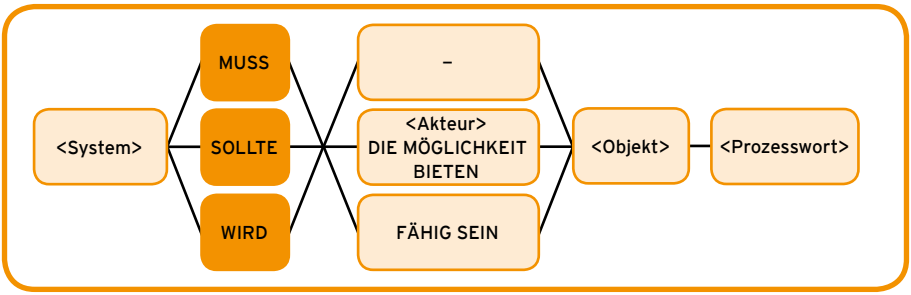


Abbildung 03: Die Schlüsselwörter der rechtlichen Verbindlichkeit im FunktionsMASTER

▶▶ Schritt 2: Das Prozesswort

Charakteristisch für die deutsche Syntax und somit auch für Anforderungen, die nach FunktionsMASTER geschrieben sind, ist die Klammer aus finitem Verb (bei uns *MUSS*, *SOLLTE* oder *WIRD*) und infinitem Verb (bei uns Prozesswort). So endet ein Anforderungssatz nach dem FunktionsMASTER mit einem *Prozesswort* im Infinitiv. Es ist der semantische Kern der Anforderung, denn es nennt die geforderte Funktion des Systems. Daher ist es wichtig, Prozesswörtern genügend Aufmerksamkeit zu schenken. Eine Möglichkeit dafür ist das Erstellen einer Prozesswortliste am Anfang eines Projekts. In dieser Liste sammeln Sie alle projektrelevanten Infinitive und definieren diese in deren Bedeutung. Auf diese Weise schaffen Sie bei allen Projektbeteiligten ein gemeinsames Verständnis der Anforderungen und der darin enthaltenen Prozesswörter, wie z. B. *archivieren*, *auswählen* oder *berechnen*.

Die Wetterstation muss Temperaturinformationen archivieren.

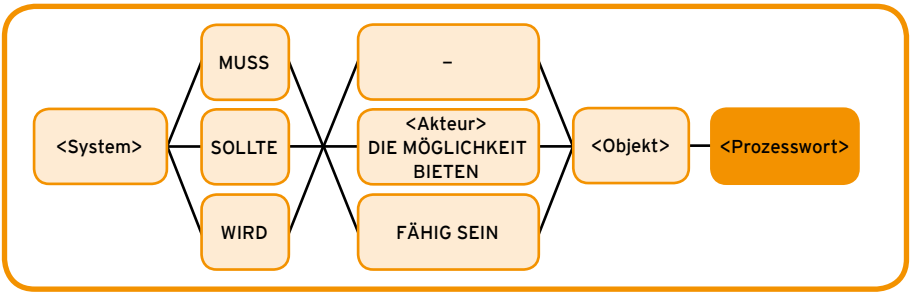


Abbildung 04: Das Prozesswort im FunktionsMASTER

Schritt 3: Die Art der Funktionalität

Mit dem FunktionsMASTeR können drei unterschiedliche Funktionsarten beschrieben werden:

- **Selbsttätige Systemaktivität:** Ein System startet eine Funktion automatisch und führt sie anschließend automatisch aus. Diese Anforderungen formulieren Sie mit der entsprechenden rechtlichen Verbindlichkeit (Modalverben *MUSS*, *SOLLTE* oder Hilfsverb *WIRD*) und einem Prozesswort im Infinitiv.
  Das Prozesswort bildet die Funktion des Systems ab, die es selbsttätig durchführt. Ein Benutzer tritt dabei nicht in Erscheinung.

- **Benutzerinteraktion:** Das System stellt einem Benutzer eine Funktionalität zur Verfügung oder es tritt mit ihm in Interaktion (das kann z. B. in Form einer Auswahlmaske geschehen). Für diese Interaktion zwischen System und Benutzer wählen Sie die Formulierung *<Akteur> DIE MÖGLICHKEIT BIETEN* und ein Prozesswort im Infinitiv mit *ZU*. Hinter der Rolle verbirgt sich ein mit dem System interagierender Benutzer.

- **Schnittstellenanforderung:** Mit Anforderungen an Schnittstellen wird der Fall abgedeckt, dass ein System eine Funktion nur in Anhängigkeit der Informationsübergabe durch einen Dritten, der nicht der Benutzer ist, ausführen kann. Das kann ein Nachbar- oder Fremdsystem sein. Die Übergabe der Informationen kann in unregelmäßigen Abständen erfolgen und unvorhersehbar sein. Formulieren Sie diese Anforderungen mit *FÄHIG SEIN* und einem Prozesswort im Infinitiv mit *ZU*.

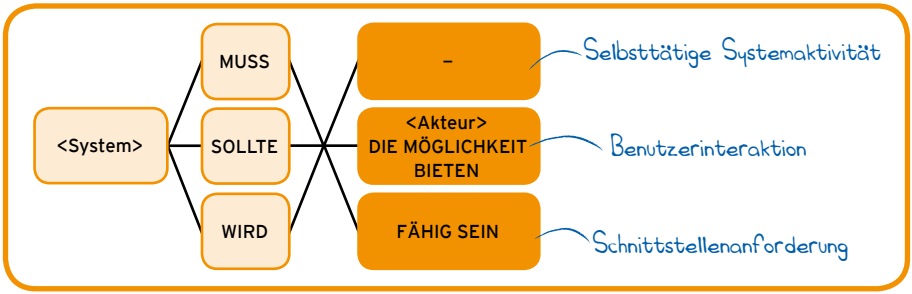


Abbildung 05: Die drei Arten der Systemfunktionalität im FunktionsMASTeR

Sie werden sicherlich auf Anforderungen stoßen, die Sie zwei oder gar allen drei Funktionsarten zuordnen würden. Sollte das der Fall sein, sollten Sie das Gespräch mit dem relevanten Stakeholder suchen und die Semantik der (noch) mehrdeutigen Systemaktivität klären. Ein besseres Systemverständnis ist die Folge und kann sich positiv auf weitere Anforderungen auswirken.

▶▶ Schritt 4: Das Objekt

Anschließend wird von der Satzschablone ein *Objekt* gefordert. Objekte sind fachlich wertvolle Begriffe im Kontext des betrachteten Gegenstandes.

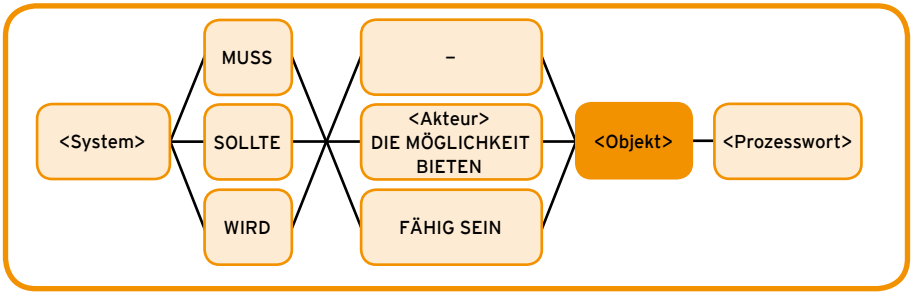


Abbildung 06: Das Objekt im FunktionsMASTERR

Unter solch ein Betrachtungsgegenstand kann Ihr System oder auch nur Teile Ihres Systems fallen. An dieser vierten Position des FunktionsMASTERs können die semantisch verschiedensten Substantive stehen. Die Semantik wird dabei stark vom Prozesswort beeinflusst, d. h. welches Objekt denn überhaupt Sinn in einer Anforderung ergibt, ist vom Prozesswort abhängig. In Tabelle 01 haben wir einige denkbare Objekte gesammelt.

OBJEKTKANDIDATEN	BEISPIELE
Greifbare Objekte	Fließband, Temperatursensor, Telefon
Rollen, die von Objekten gespielt werden	Pilot, Mitarbeiter, Käufer
Objekte, die andere Objekte zusammenfassen	Steuerprogramm, Montageanleitung, Rezept
Vorgänge, Geschehnisse	Anruf, Verkauf, Schaltvorgang

Tabelle 01: Objektkandidaten

Folgendes Beispiel soll die Verwendung der Objekte in der Tabelle 01 verdeutlichen.

Das System muss dem Benutzer die Möglichkeit bieten, Rezepte zu ordnen.

In diesem Beispiel wird eine Ordnungsfunktion vom System gefordert, die es seinem Benutzer zur Verfügung stellen muss. Das Objekt *Rezepte* ist ein Oberbegriff, d. h. unter ihm werden Anleitungen zusammengefasst, mit denen der Benutzer verschiedene Gerichte zubereiten kann.

▶▶ Schritt 5: Die Bedingungen

In der Praxis sind nicht nur einfache Anforderungssätze gebräuchlich. Funktionale Anforderungen sind häufig an Bedingungen geknüpft. Bestimmte Funktionen stellt ein System erst unter einer Reihe von Bedingungen zur Verfügung.

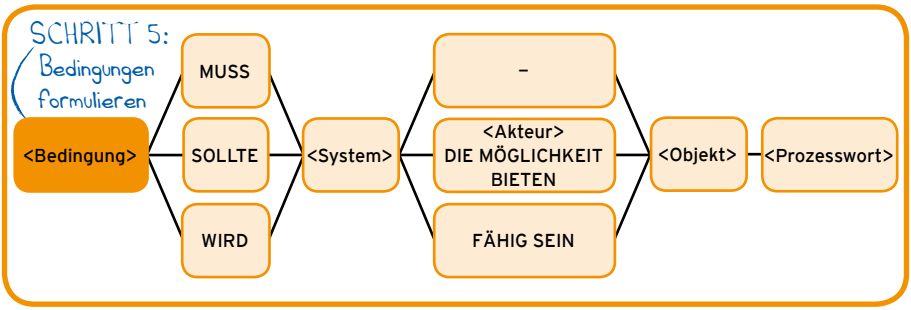


Abbildung 07: Die Bedingung im FunktionsMASTeR

Wie aber formulieren Sie eine gut verständliche Anforderung mit diesen komplexen Informationen? Der FunktionsMASTeR in Abbildung 07 gibt die Satzposition vor, auf der die Bedingung steht. Zum einen können Sie die Bedingung frei formulieren. Zum anderen finden Sie aber auch umfangreiche Formulierungsvorschläge in den Schablonen des BedingungsMASTeRs im vierten Kapitel.

An dieser Stelle sei lediglich auf die Syntax einer komplexen Anforderung bzw. Satzes hingewiesen. Die Bedingung in Form eines Nebensatzes eröffnet den Satz auf der ersten Position Nach einem Komma folgt das finite Verb bzw. die rechtliche Verbindlichkeit auf der zweiten Position. Das eigentliche System wird auf der dritten Position genannt. Die anderen Positionen des FunktionsMASTeRs verändern sich nicht.

In folgenden zwei Beispielen wird dies deutlich:

Sobald das Ereignis Geburtstag eintritt, muss das System dem Benutzer einen Kalendereintrag anzeigen.

Falls der Benutzer einen Termin eingibt, muss das System dem Benutzer die Möglichkeit bieten, den Termin zu speichern.

Die Beispiele sind wie im SOPHIST-*RE*gelwerk (siehe Kapitel 7 in [Rupp14]) empfohlen im Aktiv geschrieben und verwenden genau ein Prozesswort.

2.2 Detaillierter FunktionsMASTeR

Bei dem detaillierten FunktionsMASTeR handelt es sich nicht um eine eigenständige Schablone. Er verdeutlicht nur, wie sich Anforderungen semantisch ausführlicher beschreiben lassen. Die Detaillierungsmöglichkeiten beziehen sich auf die letzten beiden Positionen der Schablone. Somit beschäftigt sich Kapitel 2.2 schwerpunktmäßig mit der Frage, welche Formulierungsmöglichkeiten die deutsche Sprache für die Präzisierung des Objekts und für die Konkretisierung des Prozessworts bietet. In Abbildung 08 sind Fragewörter integriert, mit denen der Informationsgehalt einer Anforderung erhöht werden kann.

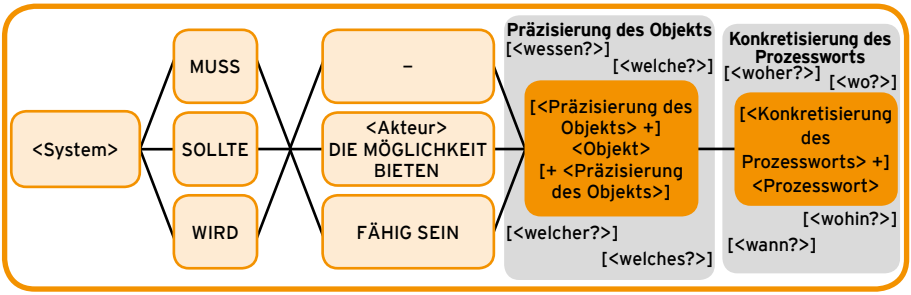


Abbildung 08: Detaillierter FunktionsMASTeR

▶▶ Objekt & Präzisierung des Objekts

Folgende, nach FunktionsMASTeR geschriebene Anforderung ist denkbar.

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

Das Objekt *Kundendaten* fasst andere Objekte zusammen. So kann sich hinter *Kundendaten* Vor- und Nachname, Geburtstagsdatum und die Adresse des Kunden verbergen.
In einer Spezifikation könnten folgende Fragen auftauchen:

- ***Von welchen Kundendaten ist hier eigentlich die Rede?***
- ***Wessen Kundendaten sind gemeint?***

In einem solchen Fall greift der detaillierte FunktionsMASTER. Das *Objekt* in einer Anforderung kann durch ein voran- und/oder ein nachgestelltes Attribut präzisiert werden.

Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten, die selektierten Kundendaten eines registrierten Kunden zu drucken.

Mit dem Fragewort *welche/r/s* kann ein *Objekt* um eine Eigenschaft oder einen Zustand ergänzt werden. Diese Konkretisierung ist dem Objekt vorangestellt. Eine weitere Konkretisierung des Objekts ist mit dem Fragewort *wessen* möglich. *Wessen* konkretisiert die Rolle oder den Akteur, für den das Objekt gilt. Diese Information ist dem Objekt nachgestellt.

Das *Objekt* in der Beispielanforderung ist *Kundendaten*. Das vorangestellte Attribut *selektierten* kann mit dem Fragewort *welche* erfragt werden. Die große Masse der Kundendaten wird gefiltert. Durch diese Einschränkung bekommen Sie die Angaben, die Sie brauchen: nicht alle *Kundendaten,* sondern nur die *selektierten*. Im Gegensatz dazu das nachgestellte Attribut: Mit *wessen* kann *eines registrierten Kunden* erfragt werden. Hier geht es um den Besitzer der *Kundendaten*. Für die Rolle der *registrierten Kunden* gelten die *selektierten Kundendaten*. Für einen Kunden, der nicht registriert ist, können keine Kundendaten selektiert werden.

▶▶ Prozesswort & Konkretisierung des Prozessworts

Das Prozesswort ist der semantische Kern einer Anforderung. Eine Konkretisierung des Prozessworts ist im Sinne einer vollständigen und eindeutigen Anforderung daher sinnvoll.

Diese Konkretisierung kann mit der Antwort auf jeweils vier Fragewörter geschehen.
Die Antworten auf diese Fragen bilden die Konkretisierung, die vor dem Prozesswort steht.
Das Prozesswort als infinites Verb steht immer auf der letzten Position einer Anforderung.

- ***Wann konkretisiert den Zeitpunkt, zu dem ein Prozess gestartet wird oder verfügbar sein soll.***
- ***Wo konkretisiert einen Betrachtungsgegenstand und eventuell den Standort, an dem ein Prozess durchgeführt wird.***

- ***Wohin** konkretisiert Betrachtungsgegenstände (Fremd- oder Nachbarsysteme), mit denen das zu spezifizierende System interagiert.*
- ***Woher** konkretisiert Betrachtungsgegenstände (Fremd- oder Nachbarsysteme), die mit dem zu spezifizierenden System agieren.*

Im folgenden Beispiel wurde die Wo-Frage beantwortet:

> ***Das Bibliothekssystem muss dem Bibliothekar die Möglichkeit bieten, die selektierten Kundendaten eines registrierten Kunden auf einem Drucker zu drucken.***

Hingewiesen sei an dieser Stelle auf das Fragewort *wie*. Es wurde bewusst nicht in die Liste der Fragewörter aufgenommen, da es meist nicht-funktionale Anteile in den FunktionsMASTeR integrieren würde. Beim Erstellen einer Spezifikation sollten Sie sich den Unterschied zwischen funktionalen und nicht-funktionalen Anforderungen bewusst machen. Auch wenn diese Trennung unter Umständen erst nach der Ermittlung von Anforderungen erfolgen kann, da Stakeholdern diese Unterscheidung manchmal schwer fällt, ist sie sinnvoll. Wie in dieser Broschüre können Sie dafür unterschiedliche Kapitel in Ihrem Anforderungsdokument anlegen. Durch Referenzen können Sie deutlich machen, dass nicht-funktionale Anforderungen die Funktionen eines Systems beeinflussen.

2.3 Englische Versionen

Die englische Sprache hat in den letzten Jahren in der Firmenkommunikation einen wichtigen Stellenwert eingenommen. Die Gründe dafür liegen auf der Hand: Unternehmen sind global aufgestellt und längst sind die Arbeitnehmer nicht mehr ausschließlich deutsche Muttersprachler. Projekte werden über Sprachen- und Ländergrenzen hinweg besetzt. Das heißt, dass Meetings und Vorträge auf Englisch gehalten werden. Aber auch der Schriftverkehr greift auf das Englische als Lingua Franca zurück: E-Mails und Spezifikationen werden auf Englisch geschrieben. Auch wenn Englisch für viele Projektbeteiligte nicht die Muttersprache ist, ermöglicht es die Kommunikation zwischen verschiedensprachigen Projektbeteiligten. Multilinguale Projekte sind so erst denkbar.

Auch RE kommt in einem multilingualen Umfeld zum Einsatz. Aus diesem Grund gibt es jede der MASTeR-Schablonen auch in englischer Sprache. Bei den Übersetzungen war nicht eine wortwörtliche Übersetzung das Ziel. Vielmehr war es uns wichtig, die Semantik der deutschen MASTeR-Schablonen in das Englische zu übertragen. Abbildung 09 zeigt den englischen FunktionsMASTeR (FunctionalMASTeR).

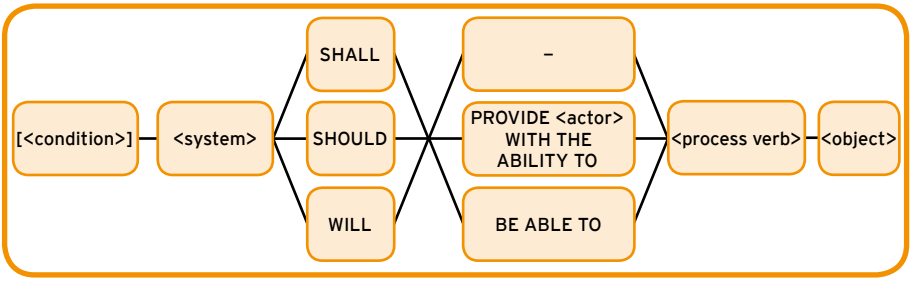


Abbildung 09: FunctionalMASTeR

Die Ausführungen zu den einzelnen Satzpositionen des FunktionsMASTeRs gelten auch für die Schablone in Abbildung 09. Der FunctionalMASTeR kann *optional* mit einer *condition* beginnen. Ist die Anforderung an keine Bedingung geknüpft, steht auf der ersten Position das *system*. Für die rechtliche Verbindlichkeit empfehlen wir, wie auch der IREB e.V., *SHALL, SHOULD* und *WILL*. Mit drei verschiedenen Pfaden kann die Art der Funktion eines Systems abgebildet werden. Die für das Deutsche typische Klammer aus finitem und infinitem Verb greift im Englischen nicht. Das *process verb* steht vor dem *object*.

Anforderungen nach dem FunctionalMASTeR können sein:

The system shall design postcards.

The system should provide the user with the ability to search items.

The system will provide the user with the ability to print selected colored photos at the net-work printer.

Analog den Fragestellungen und Ausführungen zur Deutschen Sprache (vgl. Kap. 2.2) existiert eine detaillierte Variante des FunctionalMASTeR im Englischen. Auch hier wurden die beiden Bestandteile <object> und <process verb> präzisiert bzw. konkretisiert.

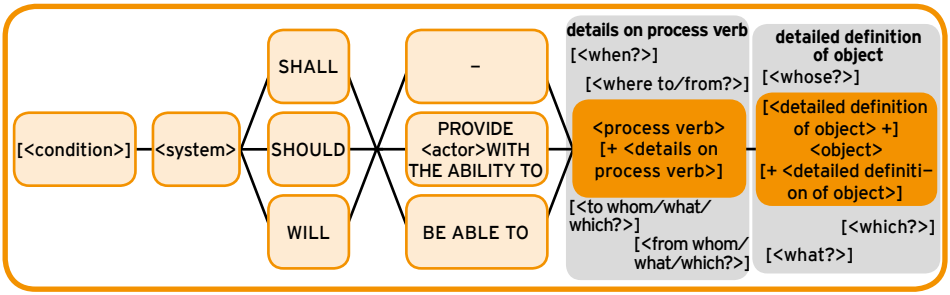


Abbildung 10: Detailed FunctionalMASTeR

3. Nicht-funktionale Anforderungen

3.1 Überblick

Die Funktionen eines Systems bilden häufig den Schwerpunkt in der Systementwicklung. Doch auch nicht-funktionale Anforderungen sollten in jedem Fall Teil Ihrer Spezifikation sein, da sie großen Einfluss auf die Realisierungsmöglichkeiten und die Akzeptanz eines Systems haben. Warum in der Ermittlung der Anforderungen eines Systems die Unterscheidung in funktionale und nicht-funktionale Anforderungen ratsam ist, illustrieren folgende, mögliche Szenarien im Arbeitsleben:

- Mitarbeitern sollte bei Bedienungsschwierigkeiten eines Systems ein Ansprechpartner oder ein Handbuch zur Verfügung stehen.
- Ersatzteile eines Systems zu bestellen darf sich nicht als Unmöglichkeit erweisen.

Diese zwei Szenarien lassen erkennen, dass die Ermittlung und Dokumentation nicht-funktionaler Anforderungen neben der eigentlichen Systementwicklung auch auf nachgelagerte Prozesse wie z. B. den Support oder die Wartung eines Systems Einfluss haben. Andere nicht-funktionale Anforderungen wiederum sind in der Realisierung des Systems nicht wegzudenken. Technologische Anforderungen z. B. können die Entwicklungsmöglichkeiten Ihres Systems einschränken.

In der Vielfalt nicht-funktionaler Anforderungen sowie in ihrer Bedeutung für eine gelungene Systementwicklung liegt die Forderung nach ihrer durchdachten Dokumentation begründet. Die Vielfalt ist auch der Grund, warum wir nicht nur eine, sondern drei Satzschablonen für nicht-funktionale Anforderungen entwickelt haben.
Bei der Entwicklung der Schablonen haben wir eine große Menge gesammelter nicht-funktionaler Anforderungen (anonymisiert) aus verschiedensten Projekten untersucht. Ziel dabei war, die essentiellen Informationen, die mit nicht-funktionalen Anforderungen beschrieben werden, herauszufinden.

Welche Satzschablone zur Formulierung von Anforderungen aus den verschiedenen Kategorien der nicht-funktionalen Anforderungen geeignet ist, fasst Tabelle 02 zusammen. Der Quasi-Alles-Könner ist der EigenschaftsMASTeR. Mit ihm können Sie Anforderungen aus allen Kategorien formulieren. Für die Kategorien *Durchzuführende Tätigkeiten* und *Rechtlich-vertragliche Anforderungen* haben Sie die freie Wahl zwischen EigenschaftsMASTeR und ProzessMASTeR.
Wählen Sie hier einfach die Satzschablone, deren Formulierung Ihnen am ehesten zusagt. Ähnlich ist es bei der Kategorie *Technologische Anforderungen*, allerdings eignet sich der Umgebungs-MASTeR hier nur für Anforderungen der Subkategorien *Umgebung* und *Mengengerüst*.

Die restlichen Subkategorien deckt der EigenschaftsMASTeR mit ab.

Vielleicht fragen Sie sich gerade, warum es den UmgebungsMASTeR gibt, wenn Sie mit dem EigenschaftsMASTeR doch alle technologischen Anforderungen formulieren können.
Die Antwort ist recht einfach: Die Anforderungen für diese beiden Subkategorien klingen mit dem UmgebungsMASTeR einfach besser.

	EigenschaftsMASTeR	UmgebungsMASTeR	ProzessMASTeR
Qualitäts-anforderungen	X		
Technologische Anforderungen	X	Umgebung, Mengengerüst	
Benutzungs-oberfläche	X		
Sonstige Liefer-bestandteile	X		
Durchzuführende Tätigkeiten	X		X
Rechtlich-vertragliche Anforderungen	X		X

Tabelle 02: MASTER-Schablonen für nicht-funktionale Anforderungen

Im nachfolgenden Kapitel 3.2 werden wir die zum Verständnis der MASTeR-Schablonen für nicht-funktionale Anforderungen benötigten Kategorien erläutern und anschließend in Kapitel 3.3 die Satzschablonen im Detail vorstellen.

3.2 Die Kategorien nicht-funktionaler Anforderungen im Überblick

Mit der Abbildung 11 erhalten Sie einen Überblick über ausgewählte Kategorien der nicht-funktionalen Anforderungen.

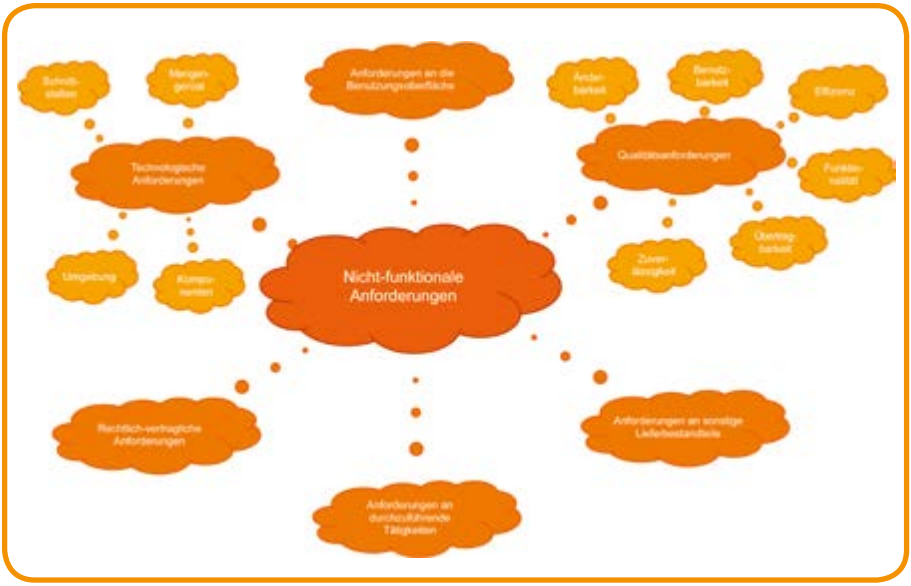


Abbildung 11: Nicht-funktionale Anforderungen im Überblick

Auf oberster Ebene lassen sich die nicht-funktionalen Anforderungen in sechs verschiedene Kategorien einteilen. Eine weitere Unterteilung ist möglich und für die Vorstellung der drei Satzschablonen für nicht-funktionale Anforderungen zum Teil auch nötig. Deshalb erläutern wir Ihnen im Folgenden die sechs Hauptkategorien und für die technologischen Anforderungen auch die dazugehörigen Subkategorien.

▶▶ Technologische Anforderungen

Unter Technologische Anforderungen verstehen wir Anforderungen, die direkten Einfluss auf die in der Entwicklung zu verwendenden Technologien haben. Auch wenn zur Realisierung eines Systems verschiedene technologische Gegebenheiten zur Verfügung stehen, sind sie vielleicht nicht im Rahmen jedes Projekts zugänglich und nutzbar. Technologische Anforderungen können in entscheidender Weise die Art der Realisierung des Systems be-

einflussen oder gar vorgeben. Sie können in vier verschieden Gruppen eingeteilt werden:

▶ Anforderungen, die sich aus der Systemumwelt ergeben

Umwelteinflüsse können das physikalische Umfeld (Temperatur, Feuchtigkeit, Druck, Bewegung, chemische Beanspruchung etc.) und das technologische Umfeld eines Systems (Stromversorgung, Ausrüstung und das Netzwerk etc.) sein. Sie beschreiben hier, welchen klimatischen, elektronischen oder elektrischen Bedingungen ein System in seinem Umfeld ausgesetzt wird.

> *Der Fahrradcomputer muss so gestaltet sein, dass er bei Temperaturen von -25°C bis +50°C betrieben werden kann.*

▶ Mengengerüstanforderungen

Diese Anforderungen geben Mengenangaben wie Speicherplatzkapazität, Stromstärken, Benutzeranzahlen oder auch Stückzahlen vor.

> *Die Datenbank des Systems muss so gestaltet sein, dass das System während der Bürozeiten von 8-17 Uhr von 100 Nutzern gleichzeititg betrieben werden kann.*

▶ Anforderungen an Schnittstellen

Schnittstellen können mechanisch oder softwarespezifisch sein. Mechanische Schnittstellen beschreiben die mechanischen Verbindungen zwischen dem zu entwickelnden System und seiner Umwelt. Um mechanische Anforderungen sinnvoll zu formulieren, ist es ratsam, auf einzelne Komponenten des Systems Bezug zu nehmen. Softwareschnittstellen umfassen bekannte, standardisierte Protokolle, die referenziert werden können oder proprietäre Protokolle, die neu definiert werden können.

> *Das Bibliothekssystem muss so gestaltet sein, dass es bei einer Spannung zwischen 210 Volt und 230 Volt betrieben werden kann.*

▶ Vorgaben an Komponenten

Diese Kategorie umfasst Vorgaben an Hardware- oder Softwarekomponenten, aus denen sich das System zusammensetzen soll. Es kann sich dabei auch um Vorgaben handeln, die bei der Entwicklung der Komponenten zu beachten sind.

> *Die Erwärmung der Systemkomponenten muss kleiner gleich 50°C sein.*

Qualitätsanforderungen

Qualitätsanforderungen machen Angaben über die Güte des Produkts, des Prozesses oder der am Prozess beteiligten Personen. Sie definieren damit für ein zu entwickelndes System die qualitativen Eigenschaften, die das betrachtete System bzw. einzelne Funktionen aufweisen müssen. Sie lassen sich in sechs verschiedene Gruppen unterteilen:

- **Änderbarkeit**
- **Benutzbarkeit**
- **Effizienz**
- **Funktionalität**
- **Übertragbarkeit**
- **Zuverlässigkeit**

Der Aufwand für die Integration einer neuen Funktionalität muss kleiner gleich 3 Tage sein.

Die Einarbeitungszeit für einen Benutzer muss kleiner gleich 6 Tage sein.

Die Zeitdauer für die Übertragung auf einen anderen Server muss kleiner gleich 30 Minuten sein.

Die Erreichbarkeit des Systems muss größer als 98 % pro Jahr sein.

Anforderungen an die Benutzungsoberfläche

Mit Anforderungen an die Benutzungsoberfläche Ihres Systems wird eine Menge von Faktoren beschrieben, die sich mit dem visuellen, akustischen oder auch haptischen Erscheinen und der Bedienung des Systems befassen. Sie ergänzen die funktionalen Anforderungen, die sich an der Oberfläche zeigen und spezifizieren, wie diese funktionalen Anforderungen zu erscheinen haben.
Wir unterteilen sie in die Kategorien:

- **Bedienkonzepte,**
- **Rahmenbedingungen für die Gestaltung der Benutzungsoberfläche sowie**
- **Bedienelemente.**

Die Farbe aller Buttons auf der Benutzungsoberfläche muss schwarz sein.

Anforderungen an sonstige Lieferbestandteile

In diese Kategorie fallen alle Lieferbestandteile, die während der Entwicklung mit erstellt werden müssen, wie z. B. Benutzerhandbücher oder ähnliches. Hier wird auch beschrieben, was die zu erstellenden Unterlagen beinhalten sollen.
Einige typische Lieferbestandteile sind:

- **Dokumentationen zu einzelnen Tätigkeiten,**
- **Hardware- und Softwaredokumentationen,**
- **Benutzer-, Installations- und Wartungshandbücher,**
- **Schulungsunterlagen,**
- **Installationssoftware und**
- **Werkzeuge zur Montage von Komponenten.**

Das CI-Layout der Schulungsunterlagen muss gleich dem CI-Layout vom 19.08.2013 sein.

Anforderungen an durchzuführende Tätigkeiten

Manchmal sollen auch Projektprozesse beschrieben werden, z. B. wie das Produkt zu entwickeln ist, wie das Produkt eingeführt oder betrieben werden soll oder wie bestimmte Meetings abzuhalten sind. Diese werden meist nicht in der Anforderungsspezifikation beschrieben, sondern in einem Projekthandbuch oder in Vereinbarungen zwischen Auftraggeber und Auftragnehmer.

Der Auftragnehmer muss die Risikoliste wöchentlich aktualisieren.

Rechtlich-vertragliche Anforderungen

Unter rechtlich-vertraglichen Anforderungen werden alle Anforderungen zu vereinbarten Rechten und Pflichten in Bezug auf Entwicklung und Verwendung des zu erstellenden Produkts verstanden. Wegen des eingeschränkten Leserkreises und der wenig direkten Verbindungen zu anderen Anforderungen finden Sie diese Anforderungen häufig in einem eigenen Spezifikationskapitel oder sogar in einem separaten Dokument.
Sie lassen sich in die folgenden Gruppen unterteilen:

- **Anforderungen an den Auftragnehmer,**
- **Kosten,**
- **Anforderungen hinsichtlich des Angebotsprozesses und des Angebots,**

- **rechtliche Anforderungen und**
- **Übereinstimmungsanforderungen.**

Der Auftragnehmer muss mindestens 60 % der Leistung mit eigenen Mitarbeitern erbringen.

Im nächsten Schritt erläutern wir die Schablonen, die für die nicht-funktionalen Kategorien Anwendung finden.

3.3 EigenschaftsMASTeR

Sie können mit dem EigenschaftsMASTeR einen Großteil der nicht-funktionalen Anforderungen formulieren:

- **Qualitätsanforderungen,**
- **technologische Anforderungen,**
- **Anforderungen an die Benutzungsoberfläche,**
- **Anforderung an die sonstigen Lieferbestandteile sowie**
- **rechtlich-vertragliche Anforderungen.**

Mit folgender Schablone formulieren Sie Anforderungen der obigen Kategorien (Abb. 12):

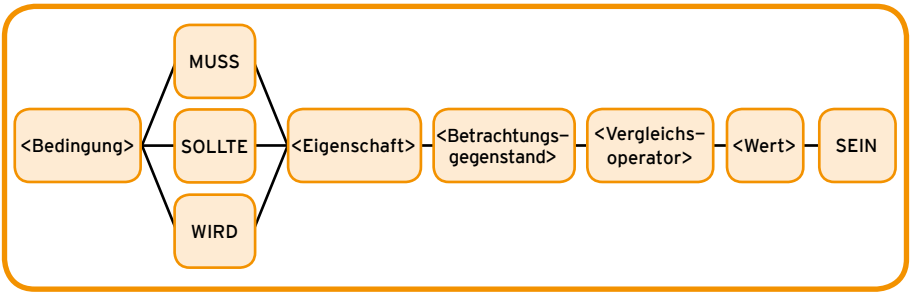


Abbildung 12: EigenschaftsMASTeR

Anhand des folgenden Beispiels werden die Besonderheiten dieser Schablone verdeutlicht.

Das Gewicht des Smartphone muss kleiner gleich 100g sein.

Betrachten wir die Positionen im Einzelnen, um die verwendeten Begriffe besser zu verstehen.

Eigenschaft und Betrachtungsgegenstand

Eine nach dem EigenschaftsMASTER formulierte Anforderung beginnt mit einer *Eigenschaft* eines *Betrachtungsgegenstandes*. Ein bestimmtes Merkmal, das zum Betrachtungsgegenstand gehört, wird hier als Eigenschaft bezeichnet. Der Betrachtungsgegenstand, hinter dem sich der zu entwickelnde Liefergegenstand verbirgt, *MUSS, SOLLTE* oder *WIRD* diese Eigentümlichkeit aufweisen.

Ob sich hinter Betrachtungsgegenstand ein ganzes System oder nur ein Teil bzw. eine Komponente eines Systems oder auch eine Funktion verbirgt, hängt von der gewählten Spezifikationsebene ab. Sie können mit dieser Schablone Anforderungen an Systemteile auf verschiedenen Ebenen stellen. Folgende drei Fragen können helfen, den Betrachtungsgegenstand, der eine gewünschte Eigenschaft haben soll, einzugrenzen:

- **Welcher Teil des Systems ist gerade der Fokus meiner Arbeit?**
- **An welche Eigenschaft meines Betrachtungsgegenstandes habe ich Erwartungen?**
- **In welcher Qualität soll eine Funktion realisiert weden?**

In unserem Beispiel steht eine Eigenschaft (*Gewicht*) des Betrachtungsgegenstands (*Smartphone*) im Vordergrund.

Rechtliche Verbindlichkeit

Auch bei nicht-funktionalen Anforderungen muss die rechtliche Verbindlichkeit geklärt sein. Analog zum FunktionsMASTER für funktionale Anforderungen empfehlen wir die bekannten Schlüsselwörter:

- **Alle Anforderungen, die mit *MUSS* formuliert sind, sind verpflichtend in der Umsetzung.**
- **Formulierungen mit *SOLLTE* stellen einen Wunsch eines Stakeholders dar.**
- **Mit *WIRD* dokumentieren Sie die Absicht eines Stakeholders.**

Aus diesen drei Schlüsselwörtern ist der Grad der Bedeutung, die ein Stakeholder einer Anforderung beimisst oder die sich aus Gesetzen und anderen Randbedingungen ergibt, zu erkennen (vgl. Kap. 2.1).

Vergleichsoperator, Wert und SEIN

Die Schablone verlangt nun einen konkreten *Wert*, der sich aus der Eigenschaft ergibt. Das kann ein Zahlenwert und die semantisch passende Einheit sein, z. B. Zeit, Gewicht, Längenmaß etc. Genauso kann an dieser Stelle aber auch eine Wortgruppe stehen, die einen Wert repräsentiert, der sich aus der in der Anforderung genannten Eigenschaft ergibt:

> ***Die Farbe des Gehäuses des Smartphones muss weiß sein.***
>
> ***Das Dateiformat des Monatsreports muss PDF in der Version 1.5 sein.***

Die Angabe des Werts ist obligatorisch. Die vorangehende Angabe eines *Vergleichsoperators*, wie z. B. *mindestens, maximal* oder *größer (als)* ist fakultativ. Die eingangs genannte Beispielanforderung nutzt den Vergleichsoperator *kleiner gleich*. Der Zahlenwert wird von der Mengeneinheit Gramm komplettiert (*100g*). Die zwei weiteren Beispiele haben keinen Vergleichsoperator. Die Werte *weiß* und *PDF in der Version 1.5* ergeben sich aus der Eigenschaft *Farbe* bzw. *Dateiformat*. Der Schreiber einer Anforderung nach dem EigenschaftsMASTeR wird durch die vorgegeben Positionen der Schablone zur Genauigkeit angehalten. Schnell könnte man einfach schreiben *Das Smartphone muss weiß sein*. Die Information, dass es sich hier um eine Eigenschaft des Gehäuses handelt, scheint offensichtlich und wird vielleicht gerade deshalb oft nicht dokumentiert. Sehen Sie den EigenschaftsMASTeR als ein Hilfsmittel, mit dem Sie implizierte Annahmen bewusst machen können.

Eine nach dieser Schablone formulierte Anforderung endet mit dem Vollverb *SEIN*. Andere Vollverben, die Funktionen beschreiben (z. B. *drucken, rechnen*), sind an dieser Stelle unangebracht, denn der EigenschaftsMASTeR beschreibt keine Funktionen. Das geschieht mit dem FunktionsMASTeR (vgl. Kap 2).

Gegenüber frei geschriebenen Anforderungen haben Anforderungen, die nach dem EigenschaftsMASTeR geschrieben sind, zwei Vorteile: Die Systembeschreibung wird genauer und die Anforderungen haben eine erhöhte Testbarkeit. Der genannte Wert gibt dem Tester konkrete Vorgaben für seinen Test. Mit Vergleichsoperator und Wert beschreiben Sie die Eigenschaft eines Betrachtungsgegenstandes genauer. Die Eigenschaft wird mess- bzw. testbar. Sie schreiben mit dem EigenschaftsMASTeR also automatisch Anforderungen, die einem wichtigen Qualitätskriterium für Anforderungen (testbar) entsprechen. Im Gegensatz dazu sind nämlich (gesteigerte) Adjektive wie schnell, effektiv, besser oder leichter ohne jedwede Eingrenzung oder Referenz nicht testbar.
Weitere Anforderungen gemäß dieser Ausführungen könnten wie folgt lauten:

> ***Die Dauer des Einschaltvorgangs des Fernsehers muss in 40 % der Fälle maximal fünf Sekunden sein.***
>
> ***Das Dateiformat des Pflichtenhefts muss gleich PDF sein.***

3.4 UmgebungsMASTeR

Mit der nächsten Schablone, dem UmgebungsMASTeR, stellen wir Ihnen eine Schablone für Anforderungen aus der Kategorie der technologischen Anforderungen, nämlich Anforderungen aus der Umgebung und Mengengerüstanforderungen, zur Verfügung (Abb. 13).

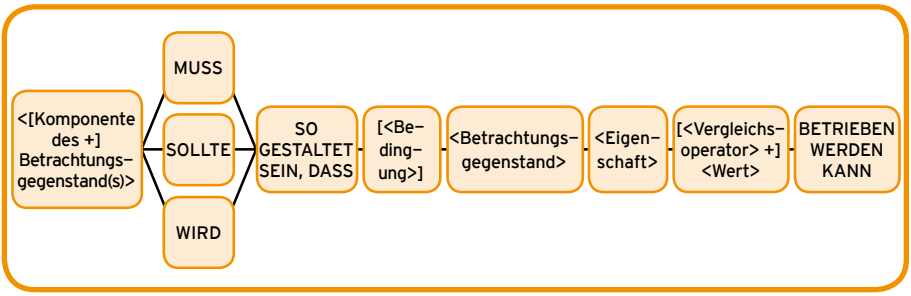


Abbildung 13: UmgebungsMASTeR

Diese Schablone hält Sie dazu an, über die technologischen Randbedingungen Ihres Systems nachzudenken. Systeme müssen heutzutage eine stattliche Menge Anforderungen aus der Umgebung erfüllen. Sie müssen z. B. bei Extremtemperaturen und am besten unter Wasser funktionieren.
In der realen Projektwelt liest sich eine Anforderung aus der Umgebung des Systems bisher oft so: *Die Umgebungstemperatur des Smartphones muss -20 bis 60°C sein.* Hier wurde der Fokus der Anforderung falsch gesetzt. Die Anforderung wurde an die Umgebungstemperatur gestellt und nicht an das System. Der Unterschied wird mit einem Vergleich sehr gut deutlich. Dazu formulieren wir das eben genannte Beispiel nach dem UmgebungsMASTeR:

> ***Die Außenhülle des Smartphones muss so gestaltet sein, dass das Smartphone bei einer Umgebungstemperatur von -20°C bis 60°C betrieben werden kann.***

Bei unserem Beispiel handelt es sich um eine Anforderung, die sich aus einer möglichen Umgebung des Betrachtungsgegenstandes heraus ergibt. Denn wird das Smartphone bei einer Außentemperatur von -17°C benutzt, wäre sein Besitzer sehr enttäuscht, könnte er allein aufgrund der extremen Umgebungstemperatur die Funktionen nicht verwenden.
Aus welchen Bestandteilen setzt sich nun die Schablone hinter der genannten Anforderung zusammen (vgl. Abb. 13)?

Hauptsatz

Es fällt auf, dass Sie – im Gegensatz zu den bereits eingeführten Schablonen – mit dieser Schablone komplizierte Anforderungen schreiben. Anforderungen nach dem UmgebungsMASTER bestehen aus einem Haupt- und einem untergeordneten Nebensatz.

Betrachtungsgegenstand

Wie schon mit dem EigenschaftsMASTER können Sie auch mit dem UmgebungsMASTER Anforderungen an einzelne Teile eines *Betrachtungsgegenstandes* (in der Schablone Komponente) Ihres Systems schreiben. So wird die Anforderung genauer und die Entwicklung und der Test haben klare Anhaltspunkte für ihre Arbeit.

Im Kapitel 2.1 wurde erklärt, dass sich der Begriff *Betrachtungsgegenstand* sowohl auf ein ganzes System als auch einzelne Funktionen oder Komponenten des Systems beziehen kann. Durch die Nennung von Komponenten kann ein System auf immer tiefer gehenden Ebenen beschrieben werden (z. B. Smartphone – Akku, Smartphone – Display – LCD, Handy – Bedienfeld – Taste). Im Hauptsatz des UmgebungsMASTERs führen wir daher *Komponente* als einen fakultativen Bestandteil der Schablone auf. Der im Nebensatz geforderte Betrachtungsgegenstand ist der gleiche wie im Hauptsatz. Die Unterscheidung in Komponente und Betrachtungsgegenstand im UmgebungsMASTER macht eine Anforderung genauer und erleichtert das Verständnis der Anforderung.

In unserer Beispielanforderung wird eine Anforderung an eine Komponente (Außenhülle) des Betrachtungsgegenstandes (Smartphone) gestellt.

Rechtliche Verbindlichkeit

Auch der UmgebungsMASTER greift auf die bereits bekannten Schlüsselwörter für die rechtliche Verbindlichkeit zurück (vgl. Kap. 2.1).

- **Alle Anforderungen, die mit *MUSS* formuliert sind, sind verpflichtend in der Umsetzung.**
- **Formulierungen mit *SOLLTE* stellen einen Wunsch eines Stakeholders dar.**
- **Mit *WIRD* dokumentieren Sie die Absicht eines Stakeholders.**

▶▶ SO GESTALTET SEIN, DASS

SO GESTALTET SEIN, DASS ist ein obligatorischer Teil des UmgebungsMASTERs. *Gestalten* verdeutlicht die Bedeutung des UmgebungsMASTERs – bedeutet es doch, dass dem genannten Betrachtungsgegenstand oder seiner Komponente eine bestimmte Form gegeben werden *MUSS/SOLLTE/WIRD* oder er/sie in einer bestimmten Art entwickelt werden *MUSS/SOLLTE/WIRD*.

▶▶ Nebensatz

Die Konjunktion *DASS* leitet einen untergeordneten Nebensatz ein.

▶▶ Betrachtungsgegenstand

Subjekt des Anforderungsnebensatzes ist Ihr Betrachtungsgegenstand aus dem Anforderungshauptsatz. Seine Nennung ist obligatorisch. In unserem Fall: *das Smartphone*.

▶▶ Eigenschaft

Nennen Sie anschließend die Eigenschaft, die Ihr Betrachtungsgegenstand besitzen soll. Die Eigenheit bzw. Beschaffenheit des Betrachtungsgegenstandes kann sehr unterschiedlich sein (z. B. Gewicht, Zustand). Die Angabe der Eigenschaft ist obligatorisch.

In unserem Beispiel ist die Position der Eigenschaft mit einer Präposition, Artikel und Substantiv *(bei einer Umgebungstemperatur)* besetzt.

▶▶ Vergleichsoperator, Wert und SEIN

Wie im EigenschaftsMASTER ist auch im UmgebungsMASTER die Angabe eines Vergleichsoperators fakultativ (vgl. Kap. 3.2). Es handelt sich in unserem Beispiel um die Wörter *von* und *bis*. Bei genauerer Betrachtung wird deutlich, dass hier eigentlich zwei Vergleichsoperatoren und damit streng genommen zwei Anforderungen zusammengefasst sind: *größer gleich* und *kleiner gleich*. Die Position des obligatorischen Werts ist so besetzt, dass die Umgebungstemperatur durch Zahlenwerte präzisiert ist: -20°C und 60°C (vgl. Ausführungen zu *Wert* Kap. 3.2). Diese Angabe ist ein eindeutiger Hinweis für Tester.

Eine Anforderung wie:

Das Smartphone muss bei einer niedrigen Umgebungstemperatur funktionsfähig sein.

ist zu ungenau und nicht testbar, da nicht bekannt ist, wie niedrig die Umgebungstemperatur für das Smartphone genau sein darf. Diese Information wird mit dem UmgebungsMASTeR geliefert. Hier handelt es sich also um eine Schablone, mit der Sie Anforderungen schreiben, die dem Qualitätskriterium testbar entsprechen. Zusätzlich verweisen wir auf das **SOPHIST-*RE*gelwerk**: Bezugslose Eigenschaftswörter sollten hinterfragt werden. Formulieren Sie Vergleiche und Steigerungen so, dass sie mess- bzw. testbar sind.

▶▶ BETRIEBEN WERDEN KANN

UmgebungsMASTeR-Anforderungen enden immer mit der Verbgruppe *BETRIEBEN WERDEN KANN*. Durch die Verwendung dieser Verbgruppe liegt der inhaltliche Fokus immer auf der Beschreibung der Umgebung, in der der Betrachtungsgegenstand verwendet werden soll. Außerdem verdeutlicht das Modalverb *KANN*, dass die gewünschte Eigenschaft zwar vorhanden sein *MUSS/SOLLTE/WIRD*, wie oft sie aber dann in der Realität vom Kunden auch genutzt wird, ist kundenspezifisch. Nicht jeder Smartphone-Besitzer wird sein Smartphone bei bitteren -20°C benutzen.

Zwei weitere, nach dem UmgebungsMASTeR formulierte Anforderungen sind:

> ***Das System muss so gestaltet sein, dass das System mit einer Anzahl gleichzeitig surfender Gäste von mindestens 100 betrieben werden kann.***
>
> ***Das System muss so gestaltet sein, dass das System bei einer Spannung von 220 Volt +- 10 Volt betrieben werden kann.***

3.5 ProzessMASTeR

Die dritte Schablone für nicht-funktionale Anforderungen ist der ProzessMASTeR – eine Schablone für Anforderungen an durchzuführende Tätigkeiten und rechtlich-vertragliche Anforderungen (Abb. 14).

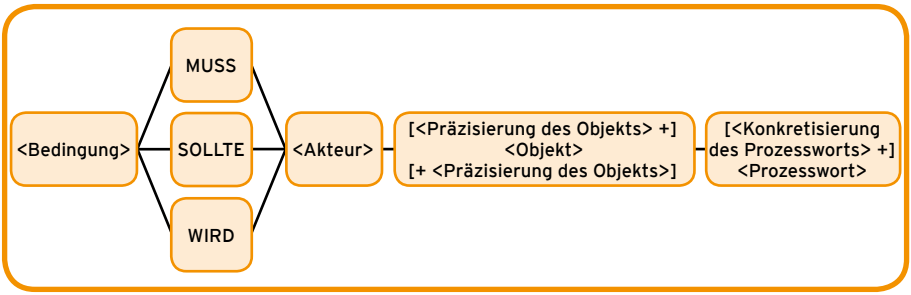


Abbildung 14: ProzessMASTeR

In diese Kategorie der nicht-funktionalen Anforderungen fallen neben Gesetzen oder Normen auch Anforderungen an den Auftragnehmer oder andere Personen. Der Faktor Mensch spielt nach wie vor eine entscheidende Rolle in der Systementwicklung. Die folgende Beispielanforderung verdeutlicht dies:

Der Auftragnehmer muss ein Betriebshandbuch für den MP3-Player erstellen.

Hinter dieser Anforderung verbirgt sich ein Prozess (hier: Wissen sammeln, gliedern, niederschreiben etc.). Die Anforderung ist nicht an das betrachtete System gestellt, sondern muss neben der Entwicklung des Systems vom Auftragnehmer realisiert werden.

Wie spiegelt sich dieser Fakt im ProzessMASTeR wider?

▶▶ Akteur

Der ProzessMASTeR greift auf einen Zweig des FunktionsMASTeRs 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. Sie geht mit der eigentlichen Entwicklung des Systems einher. 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*.

Rechtliche Verbindlichkeit

Hier ist der ProzessMASTeR keine Ausnahme. Die bereits vertrauten Modalverben und das Hilfsverb für die rechtliche Verbindlichkeit finden Anwendung:

- **Alle Anforderungen, die mit *MUSS* formuliert sind, sind verpflichtend in der Umsetzung.**
- **Formulierungen mit *SOLLTE* stellen einen Wunsch eines Stakeholders dar.**
- **Mit *WIRD* dokumentieren Sie die Absicht eines Stakeholders.**

Objekt und Präzisierungsmöglichkeiten

Auf der dritten Position der Schablone steht ein *Objekt*. Die Ausführungen zu Objekt im Kapitel 2.1 gelten ebenso für den Objektbegriff in der ProzessMASTeR-Schablone.

Daher nur eine kurze Gedächtnisauffrischung: Mit Objekt sind fachlich wertvolle Begriffe im Kontext des Betrachtungsgegenstandes gemeint. Die Präzisierung des Objekts kann auf zwei Arten erfolgen:

- **Die Präzisierung steht nach dem Objekt mit Hilfe einer Ergänzung mit Präposition (für den MP3-Player).**
- **Die Präzisierung steht vor dem Objekt mit Hilfe von Attributen (z. B. englisches).**

Prozesswort und Konkretisierungsmöglichkeiten

Auch im ProzessMASTeR besteht die Möglichkeit, die letzte Position der Schablone, das Prozesswort, zu konkretisieren. So kann ein tiefergehendes Verständnis der Anforderung beim Leser erzeugt werden. Prozesswörter wie *erstellen* oder *prüfen* sind Träger der Bedeutung der Anforderung und ihre Konkretisierung wirkt – wie auch eine Prozesswortliste – einer Fehlinterpretation entgegen.

> ***Der Auftragnehmer muss für die Wartung des Systems die Mitarbeiter des Auftraggebers einmal im Quartal schulen.***

Die bisherigen Beispielanforderungen werden um weitere ergänzt, um Ihnen ein besseres Gefühl für Anforderungen nach dem ProzessMASTeR zu vermitteln:

> ***Der Lieferant sollte gelagerte Ersatzteile jährlich im Juni prüfen.***

Die Marketingabteilung muss einen Produktkatalog in Farbe erstellen.

Der Auftragnehmer muss die für die Schulungen benötigten Computer-Systeme inklusive der benötigten Software zur Verfügung stellen.

3.6 Englische Versionen

Wie schon den FunktionsMASTER haben wir auch die nicht-funktionalen MASTER-Schablonen ins Englische übersetzt. Die Abbildungen 15 bis 17 zeigen diese drei Schablonen.

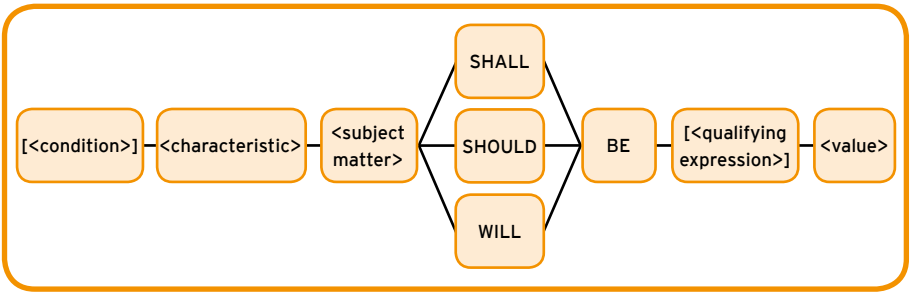


Abbildung 15: PropertyMASTER (EigenschaftsMASTER)

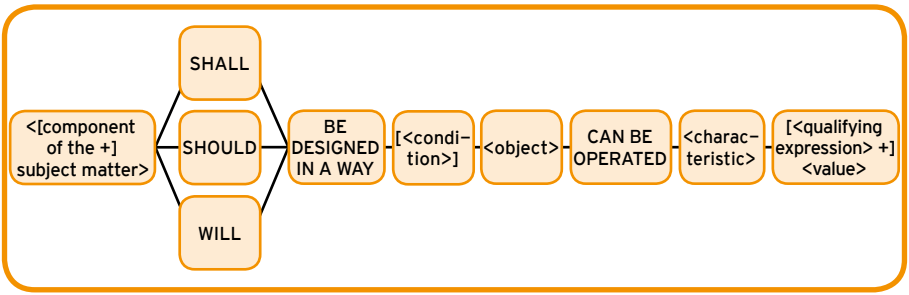


Abbildung 16: EnvironmentMASTER (UmgebungsMASTER)

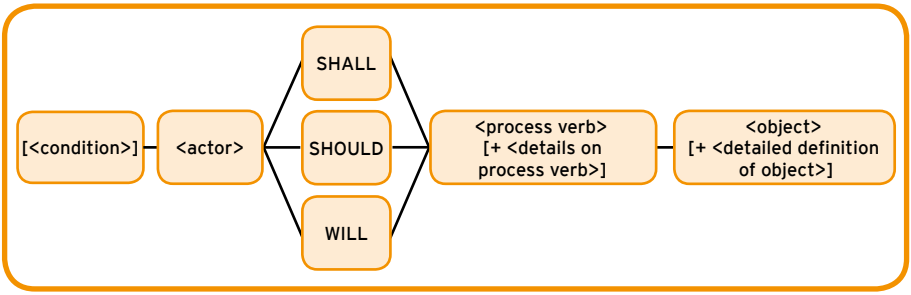


Abbildung 17: ProcessMASTeR (ProzessMASTeR)

Auf der ersten Position des PropertyMASTeRs sowie des ProcessMASTeRs kann optional eine Bedingung stehen. Es fällt auf, dass die zweite Position (ohne Bedingung ist es die erste Position) der drei Schablonen im Gegensatz zum FunctionalMASTeR mit unterschiedlichem Sprachmaterial besetzt ist. Diese Möglichkeiten spiegeln die Informationsvielfalt wider, die mit nicht-funktionalen Anforderungen ausgedrückt werden kann.

In den bisherigen Kapiteln wurde wiederholt eine Bedingung und ihre Formulierungsmöglichkeiten erwähnt. Das nächste Kapitel widmet sich nun genau diesen Möglichkeiten und den semantischen Unterschieden, die mit unterschiedlichen Formulierungen einhergehen.

4. Bedingungen

4.1 Verwendung

Systeme haben in den letzten Jahren dank neuer technischer Möglichkeiten und gestiegenen Kundenerwartungen an Komplexität gewonnen. Diese Komplexität fordert höchste Präzision in der Anforderungsermittlung. Dazu gehört auch das Ermitteln von Bedingungen, an die bestimmte Funktionen eines Systems gebunden sind. In einer Spezifikation sollten Sie eindeutig dokumentieren, unter welchen Voraussetzungen eine vom System geforderte Funktion überhaupt bereitgestellt wird und unter welchen Ausgangspunkten nicht-funktionale Anforderungen z. B. an die Benutzeroberfläche oder Qualität des Systems zum Tragen kommen.

Für die Formulierung von Bedingungen stehen drei MASTER-Satzschablonen zur Verfügung. Folgendes Beispiel verdeutlicht, dass die Notwendigkeit für Bedingungsschablonen in mehrdeutigen Bedingungen bzw. Anforderungen begründet liegt:

> ***Wenn das Smartphone inaktiv ist, muss der Stromverbrauch des Smartphones kleiner als 50 % sein.***

Die Bedingung ist der Zustand des Smartphones (*inaktiv*). Soweit kann ein Tester diese Anforderung nachvollziehen. Schwierig wird es jedoch mit der „Übersetzung" der in der Projektwelt sehr gern verwendeten Konjunktion *WENN*.
Laut Duden hat sie fünf verschiedene Bedeutungen:

1. *unter der Voraussetzung, Bedingung, dass …; für den Fall, dass …; falls*
2. *a, sobald*
   b, drückt mehrfache [regelmäßige] Wiederholung aus; sooft
3. *obwohl, obgleich*
4. *leitet einen Wunschsatz ein*
5. *leitet eine irreale vergleichende Aussage ein*

Mehrdeutigkeit ist eine häufig unterschätzte Ursache für Fehlentwicklungen.
Aber auch für einen Tester ist sie ein großes Hindernis und die Quelle für lästigen Interpretationsspielraum.

Welche Bedeutung von *wenn* ist in dem obigen Beispiel gemeint?
Was soll getestet werden?
Muss berücksichtig werden, wann das Smartphone in den Zustand inaktiv wechselt oder nicht?
Ist der Zeitraum, in dem das Smartphone im Zustand inaktiv ist, von Bedeutung?

Anhand dieser Fragen lassen sich die Detaillierungsebenen des BedingungsMASTERs ableiten. Anstatt einer mehrdeutigen Konjunktion verwendet der BedingungsMASTER drei verschiedene Konjunktionen. Der semantische Unterschied, der mit dem mehrdeutigen *WENN* völlig untergeht, kommt so zum Vorschein. Vor dem eigentlichen Schreiben einer Bedingung stehen Sie vor der Frage, welche der drei Bedingungsschablonen Sie benutzen sollen. Um diese Frage beantworten zu können, müssen Sie die Bedeutung der ermittelten Bedingung hinterfragen und Sie müssen sich intensiv mit der Bedeutung der gesamten Anforderung auseinander setzen. Und das ist essentiell für eine widerspruchsfreie Spezifikation, die einen geringen Interpretationsspielraum hat.

4.2 Detaillierungsebenen

In den Kapiteln 2 und 3 wurden die MASTER-Schablonen für funktionale und nicht-funktionale Anforderungen vorgestellt. Dabei handelt es sich um Anforderungshauptsätze und um einen komplexen Anforderungssatz. Der BedingungsMASTER stellt eine fakultative Erweiterung für die bereits eingeführten funktionalen und nicht-funktionalen MASTER-Schablonen dar. Es handelt sich hierbei um eine Schablone für Nebensätze, die alle MASTER-Schablonen – bis auf den UmgebungsMASTER – mit einem Bedingungsnebensatz einleitet. Beim UmgebungsMASTER steht die Bedingung im Satzmittelfeld.

Schreiben Sie eine komplexe funktionale oder nicht-funktionale Anforderung nach den MASTER-Schablonen, ändert sich bei der Nennung einer Bedingung die Position der rechtlichen Verbindlichkeit *MUSS, SOLLTE* und *WIRD* in der deutschen Sprache. Auf den Bedingungsnebensatz folgt das Modal- oder Hilfsverb der rechtlichen Verbindlichkeit des Anforderungshauptsatzes. Alle anderen Positionen im Anforderungshauptsatz bleiben gleich. Die Positionen des UmgebungsMASTERs ändern sich mit einer Bedingung nicht, da die Bedingung eingeschoben wird. Semantisch sind für das Formulieren einer Bedingung nach dem BedingungsMASTER drei Bedeutungen für die Konjunktion *WENN* entscheidend. Folgende drei Formulierungsmöglichkeiten stehen Ihnen zur Verfügung (Abb. 18):

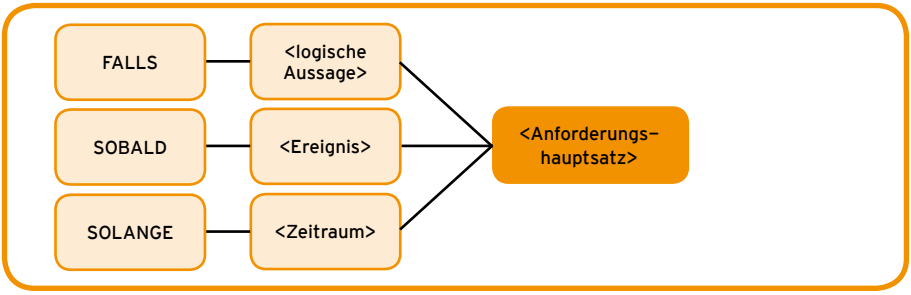


Abbildung 18: BedingungsMASTER

Zunächst legen wir das Hauptaugenmerk auf die einleitenden Konjunktionen, um so die unterschiedliche Semantik der drei BedingungsMASTeR herauszuarbeiten. In Kapitel 4.3 soll es dann um die einzelnen Positionen des jeweiligen Bedingungsnebensatzes gehen.

Der BedingungsMASTeR unterscheidet drei Bedingungsarten:

LogikMASTeR

FALLS leitet eine logische Aussage ein. Umschreibungen für diese Konjunktion sind laut Duden *für den Fall* oder *unter der Voraussetzung, dass* oder *unter der Bedingung, dass.*

EreignisMASTeR

Wie der Name schon vermuten lässt, ist ein Ereignis im EreignisMASTeR die Bedingung, an die der Anforderungshauptsatz geknüpft ist. *SOBALD* ist synonym zu *in dem Augenblick, sofort* oder *sowie* und rückt den Zeitpunkt, zu dem ein Ereignis eintritt, in den Vordergrund.

ZeitraumMASTeR

Der dritte Bedingungspfad beginnt mit *SOLANGE*. Dass hier auf einen Zeitraum referiert wird, wird mit dem Synonym *währenddessen* sehr gut deutlich. In diesem Zeitraum laufen Bedingungsnebensatz und Anforderungshauptsatz gleichzeitig ab.

Auf diese drei Bedingungsnebensätze kann ein Anforderungshauptsatz mit einer funktionalen oder nicht-funktionalen Anforderung folgen.

4.3 BedingungsMASTeR

In den drei Bedingungsschablonen tauchen folgende, bereits eingeführte Schablonen-Positionen auf:

- *Akteur*,
- *System*,
- *Vergleichsoperator* und *Wert*,
- *Objekt* sowie
- *Prozesswort*.

An dieser Stelle soll kurz zusammengefasst werden, welches Sprachmaterial auf diesen Positionen steht.

Akteur

Akteur steht als Überbegriff für die potentiellen Benutzer eines Systems, die in der Interaktion mit dem System verschiedene Rollen einnehmen können.

System

Der Begriff *System* meint den Betrachtungsgegenstand, an den Anforderungen gestellt werden, sodass er realisiert werden kann.

Vergleichsoperator und Wert

Um die Bedeutung einer Anforderung konkreter zu machen (d. h. ein Wegkommen von schwammigen Adjektiven *wie schnell* oder *intuitiv*), nennen Sie einen Vergleichsoperator wie z. B. *minimal* und einen Wert, der sich aus der Semantik der Anforderung (genauer aus dem Vergleichsgegenstand) ergibt. Neben Zeit- oder Zähleinheiten kann es sich auch um Werte handeln, die natürlichsprachlich beschrieben werden.

Objekt

Objekte sind fachlich wertvolle Begriffe im Kontext des betrachteten Gegenstandes.

Prozesswort

Als wichtigste semantische Einheit der Bedingung benennt das *Prozesswort* die funktionale oder nicht-funktionale Anforderung, die der Bedingung zu Grunde liegt.

Im nächsten Schritt werden die drei Schablonen einzeln eingeführt. Neue Positionen werden kurz erläutert und Beispiele sollen den Praxisbezug herstellen.

4.3.1 LogikMASTeR

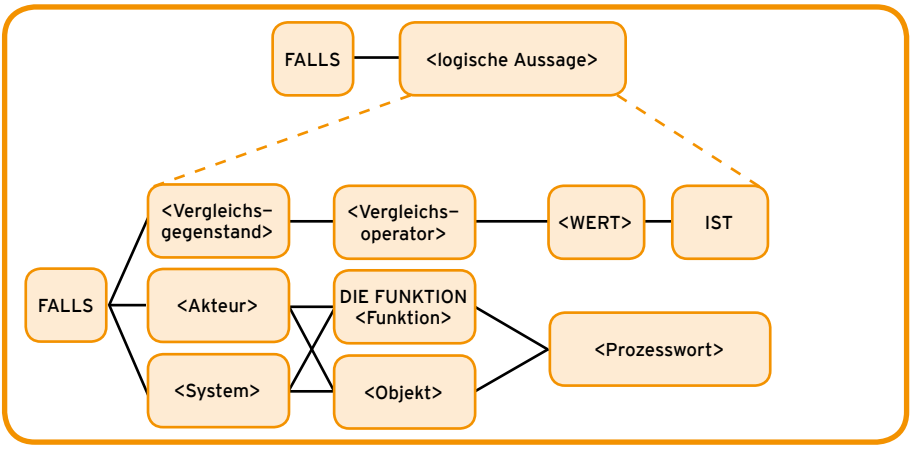


Abbildung 19: LogikMASTeR

Wie komplex die Formulierungsmöglichkeiten nach der Konjunktion *FALLS* sind, wird in Abbildung 19 deutlich. Mit allen Pfaden können Sie logische Bedingungen formulieren, an die Ihr System gebunden ist.

Folgende Beispiele sollen helfen, die komplexen Formulierungsmöglichkeiten des LogikMASTeR besser zu erfassen:

> ***Falls das Bibliothekssystem die Funktion Registrierungsdaten verifizieren nicht erfolgreich beendet, …***
>
> ***Falls das Bibliothekssystem das Leihobjekt nicht identifiziert, …***
>
> ***Falls der Bibliothekskunde das Leihobjekt reserviert, …***

Erst wenn die durch den LogikMASTeR formulierte Bedingung erfüllt ist, tritt der Anforderungshauptsatz ein. Aus diesem Grund kann die logische Aussage, die auf die Konjunktion *FALLS* folgt, nicht in der Vergangenheit stehen. Als Zeitform für den LogikMASTeR empfehlen wir Präsens. Sollten Sie auf eine Bedingung mit der Konjunktion *FALLS* in einer anderen Zeitform stoßen, hinterfragen Sie sie, um eine Bedeutungsverschiebung zu vermeiden. In den Ausführungen zum EreignisMASTeR wird die Verwendung des Perfekts empfohlen (vgl. Kap. 4.3.2). Der Bedeutungsunterschied in den genannten MASTeR-Schablonen wird also auch durch die verschiedenen Zeitformen unterstrichen.

Folgende weitere Positionen des LogikMASTERs werden nun betrachtet:

▶▶ Vergleichsgegenstand, Akteur, System

Die logische Aussage kann über einen Vergleichsgegenstand, einen Akteur oder ein System getroffen werden. Diese Satzposition ist im dazugehörigen Pfad obligatorisch zu besetzen.

▶▶ Vergleichsoperator und Wert

Beinhaltet die Anforderung einen Vergleichsgegenstand, sind die Nennung von Vergleichsoperator und Wert obligatorisch. Hinter einem Vergleichsgegenstand verbirgt sich eine Eigenschaft oder eine Rolle. Hinter Vergleichsoperator verbergen sich wie im Eigenschafts- und UmgebungsMASTER Adverbien wie z. B. *maximal* oder *mindestens*. Welcher Wert in der Anforderung steht, wird vom Vergleichsgegenstand festgelegt (natürlichsprachliche Wertbeschreibung oder Maßeinheiten).

▶▶ DIE FUNKTION <Funktion>

Die Arbeit mit diesem Pfad des LogikMASTERs setzt voraus, dass die Funktionen des Systems bereits ermittelt sind. Sie können eine Funktion, die Sie im Anforderungshauptsatz nennen, an eine Bedingung knüpfen, die möglicherweise auf einer anderen Funktion basiert. Diese Ausführungen gelten ebenso für nicht-funktionale Anforderungen.

Gängige Prozesswörter, die auf die Position DIE FUNKTION <Funktion> folgen, sind *starten*, *aktivieren*, *ausführen* oder *beenden*.

> ***Falls der Bibliothekskunde die Funktion Erleichterte Bedienung aktiviert, …***

> ***Falls der Benutzer die Funktion Hilfe beendet, …***

Folgende Beispiele sollen auch die anderen Pfade des LogikMASTERs verdeutlichen:

> ***Falls der Rezeptionist eine Buchung löscht, …***

> ***Falls der Bibliothekar die Funktion Leihobjekt suchen verwendet, …***

> ***Falls die Anzahl der entliehenen Leihobjekte größer 10 ist, …***

4.3.2 EreignisMASTeR

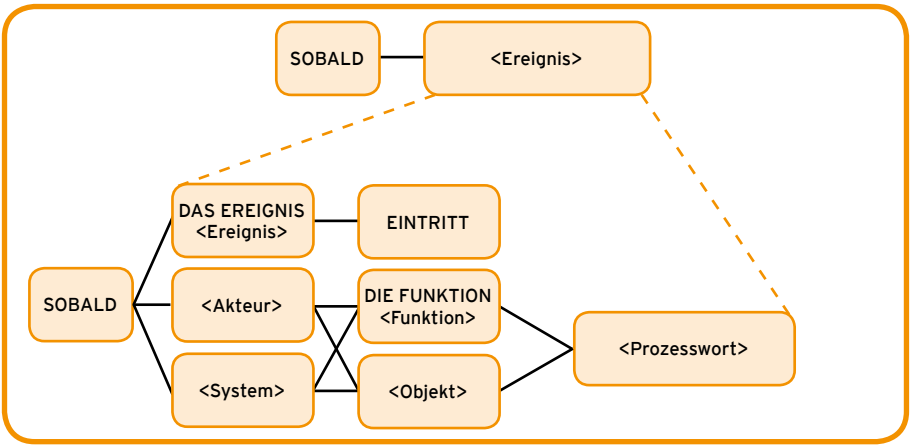


Abbildung 20: EreignisMASTeR

Abbildung 20 zeigt, wie die Position <Ereignis> verfeinert werden kann. Die folgenden Beispiele greifen auf jeweils einen Pfad des EreignisMASTeRs zurück.

> *Sobald im Bibliothekssystem das Ereignis Logindaten prüfen eintritt, …*
>
> *Sobald der Bibliothekar die Funktion Kunde registrieren aufgerufen hat, …*
>
> *Sobald der Bibliothekar alle Kundendaten eingegeben hat, …*
>
> *Sobald das Bibliothekssystem die Funktion Leihobjekt suchen abgeschlossen hat, ...*
>
> *Sobald das Bibliothekssystem die Kundendaten vom benachbarten Bibliotheks-system empfangen hat, …*

Allen Beispielen ist folgende Semantik gemeinsam: Mit der Konjunktion *SOBALD* wird eine Ungleichzeitigkeit der Bedingung und des Anforderungshauptsatzes ausgedrückt. Erst wenn die Bedingung beendet ist, beginnt bzw. gilt das funktionale oder nicht-funktionale Verhalten im Hauptsatz. In den bisherigen Bedingungsschablonen stand das Prozesswort immer im Präsens. An dieser Stelle empfehlen wir, das Prozesswort im Perfekt zu schreiben, um die zeitliche Ungleichheit von Neben- und Hauptsatz zu verdeutlichen.
Betrachten wir nun den neuen Pfad, der im EreignisMASTeR vorkommt:

▶▶ DAS EREIGNIS <Ereignis> EINTRITT

Dieser Pfad ist der Namensgeber der nun vorgestellten MASTeR-Schablone.
Der Begriff Ereignis fasst die möglichen Ereignisse, also systemrelevante Vorgänge oder Geschehnisse zusammen, die Einfluss auf das System haben können. Ein Ereignis wie ein Update oder eine Prüfung *MUSS*, *SOLLTE* oder *WIRD* Einfluss auf den Anforderungshauptsatz haben. Weitere Ereignisse können *Einschalten, Ausschalten* oder *Erschütterung* sein.

Die anderen Positionen sind mit bereits eingeführten Begriffen besetzt und folgende Beispiele sollen die möglichen Formulierungen nach dem EreignisMASTeR verdeutlichen:

Sobald das Ereignis Einschalten eintritt, …

Sobald der Benutzer die Funktion Sprachsteuerung aktiviert hat, ...

4.3.3 ZeitraumMASTeR

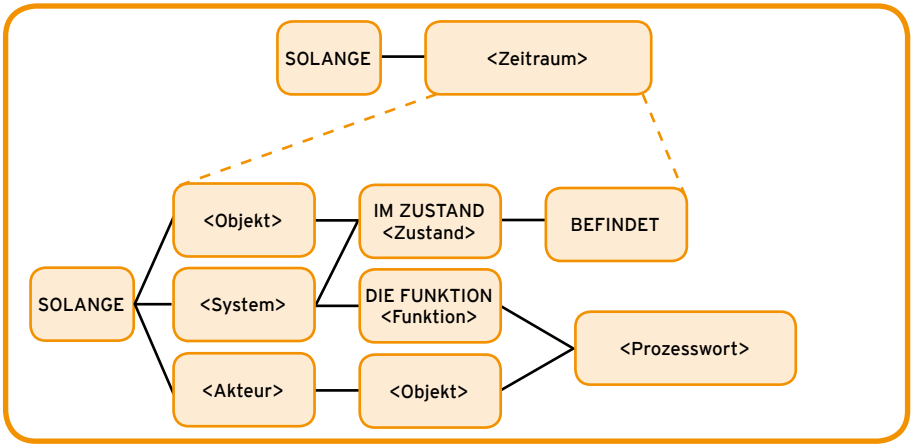


Abbildung 21: ZeitraumMASTeR

Auch der ZeitraumMASTeR bietet mehrere Möglichkeiten, eine Bedingung zu formulieren.

Solange sich ein Leihobjekt im Zustand Reserviert befindet, …

Solange sich das Bibliothekssystem im Zustand Wartung befindet, …

Solange der Administrator den Namen eines neuen Mitarbeiters aufruft, …

Solange sich das Handy im Zustand Flugmodus befindet, …

In allen Beispielanforderungen wird mit der Konjunktion *SOLANGE* die Gleichzeitigkeit des Neben- und Hauptsatzes ausgedrückt. Das bedeutet, dass sowohl die Bedingung als auch der Anforderungshauptsatz gleichzeitig enden.

▶▶ IM ZUSTAND <ZUSTAND> BEFINDET

Systeme und Objekte können verschiedene Zustände einnehmen. Sie befinden sich dann für einen bestimmten Zeitabschnitt in einer bestimmten Verfassung. Mit diesem Pfad des ZeitraumMASTERs können also an zeitlich begrenzte Verfassungen, in denen sich ein System oder Objekt befinden kann, Bedingungen gestellt werden. Verschiedenes Verhalten von Systemen oder Objekten in verschiedenen Zuständen ist beispielsweise für die Energieeffizienz entscheidend. Ruhezustand, Energie sparen oder Flugmodus sind nur einige Beispiele dafür. Mit dem ZeitraumMASTER gewinnen die unterschiedlichen Anforderungen an einen Zustand an Bedeutung und können eindeutig formuliert werden.

Die weiteren Positionen sind die gleichen wie im EreignisMASTER und LogikMASTER.

Abschließend geben wir noch zwei Beispiele, die Ihnen beim Verinnerlichen der Konjunktion und der einzelnen Satzpositionen des ZeitraumMASTERs helfen sollen:

> ***Solange das Bibliothekssystem die Funktion Verbindung zum benachbarten Bibliothekssystem herstellen ausführt, …***
>
> ***Solange sich das Smartphone im Zustand Inaktiv befindet, …***

In diesem Kapitel wurde gezeigt, dass die Konjunktionen entscheidend für die Bedeutung der jeweiligen Bedingung sind. Alle Positionen können gleich sein und die Bedingung kann sich lediglich in der Konjunktion unterscheiden. Das Schreiben von Anforderungen nach Schablonen ist also kein stumpfsinniges Bestücken einer Schablone. Die gewissenhafte Auseinandersetzung mit den Inhalten einer Anforderung ist unerlässlich.

Abschließend ein Hinweis: Sollten die nach Schablonen formulierten Anforderungen und Bedingungen nicht Ihrem Sprachgefühl entsprechen, empfehlen wir Ihnen, zumindest einzelne Bestandteile der Schablonen beim Schreiben Ihrer Anforderungen zu berücksichtigen. Setzen Sie sich bewusst mit den einzelnen Positionen einer Schablone auseinander. Sie reflektieren so wichtige, systemrelevante Fragen, die Sie nicht zwingend nach einer Schablone schreiben müssen.

4.4 Englische Versionen

Die Abbildung 22 zeigt die Formulierungsvorschläge für Bedingungen. Besonderes Augenmerk soll wie auch im Deutschen auf den einleitenden Konjunktionen liegen.

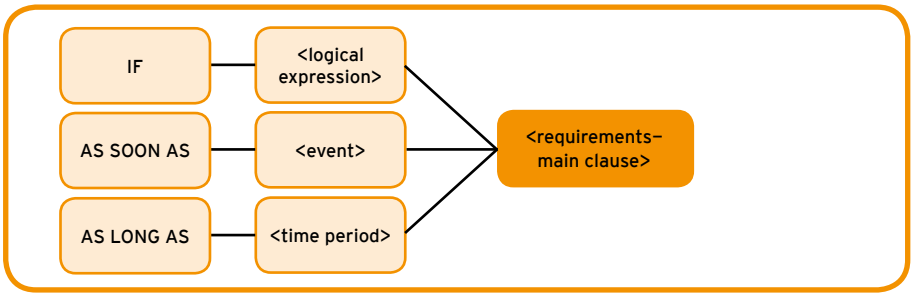


Abbildung 22: ConditionMASTER

Die drei unterschiedlichen Konjunktionen sollen mehrdeutige Anforderungen, die lediglich immer wieder auf eine Konjunktion zurückgreifen, vermeiden. Die detaillierten Pfade in den Abbildungen 23 bis 25 sind Vorschläge für die Formulierung von Bedingungsnebensätzen auf Englisch.

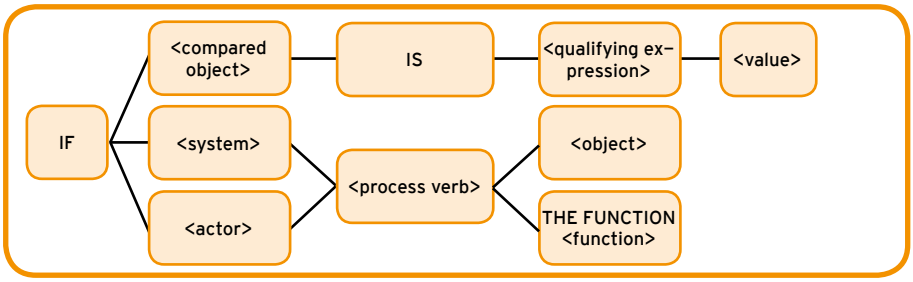


Abbildung 23: LogicMASTER

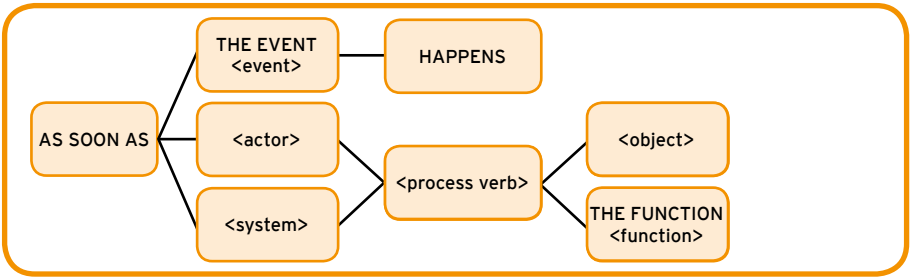


Abbildung 24: EventMASTER

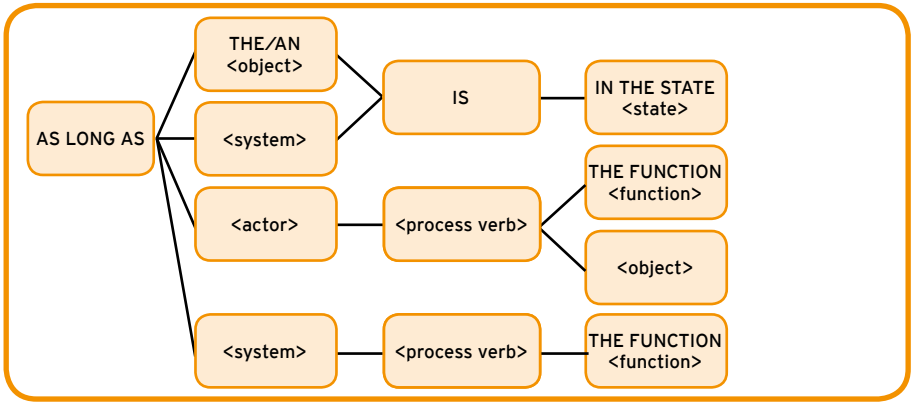


Abbildung 25: TimeMASTeR

5. Quellenverzeichnis

[Botella98] Botella, P., Burgues, X., Franch, X., Huerta, M., & Salazar, G.: Modeling non-functional requirements. Proceedings of Jornadas de Ingenieria de Requisitos Aplicada JIRA, 2001.

[Mavin10] Mavin, Alistair; Wilkinson, Philip: Big Ears (The Return of „Easy Approach to Requirements Engineering“), Requirements Engineering Conference RE), 2010

[Esser07] Esser, Michael; Struss, Peter: Automated Test Generation from Models Based on Functional Software Specifications, IICAI, 2007.

[Holtmann10] Holtmann, Jörg: Mit Satzmustern von textuellen Anforderungen zu Modellen, OBJEKTspektrum, 2010

[Rupp14] Rupp, Chris: Requirements-Engineering und –Management, Aus der Praxis von klassisch bis agil. 6. Auflage. Hanser, München, 2014.

[Pohl11] Pohl, Klaus; Rupp, Chris: Basiswissen Requirements Engineering, dpunkt Verlag, 2011

SOPHIST

» Grau is alle Theorie, maßgebend is auffen Platz. «

Adi Preißler

Genau. Deshalb müssen sich Mannschaften auch auf den Platz optimal vorbereiten. Am Besten geht das mit Profitrainern wie den SOPHISTen.

Wir coachen Kunden wie Siemens, Daimler, die Bundesagentur für Arbeit, Roche, BASF, Volkswagen, Robert Bosch, Lufthansa Systems, EnBW, VHV Versicherung und viele andere

- Anforderungen an Systeme zu hinterfragen, zu verstehen und zu dokumentieren,
- Widersprüche und Redundanzen in Modellen zu erkennen,
- Notationen richtig anzuwenden und perfekt zu modellieren,
- eine Architektur zu wählen, die länger als bis zur Abnahme aktuell ist,
- agil und angepasst zu arbeiten.

Das Ganze bieten wir in Form von Beratung, Coaching, Trainings und Vorträgen.

Wir bewahren Sie vor der Abseitsfalle.
Sprechen Sie mit uns über Ihren Trainingsplan unter:
+49 (0) 911 40 900 - 0

Oder schreiben Sie eine E-Mail an **heureka@sophist.de**

www.sophist.de

SOPHIST

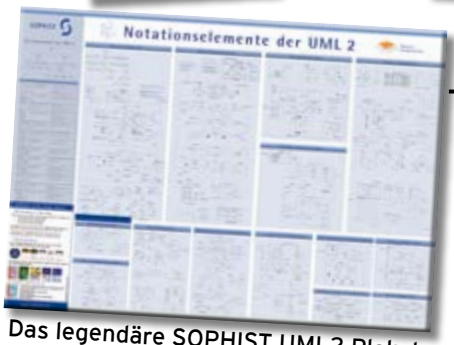
WISSEN for free

Geballtes Wissen von den SOPHISTen
Kostenfrei zu bestellen unter:
www.sophist.de/wissen-for-free

[Broschüren]

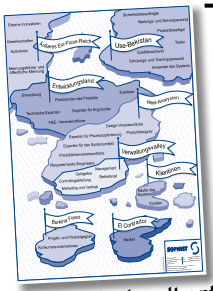
(Englisch)

[Plakate]

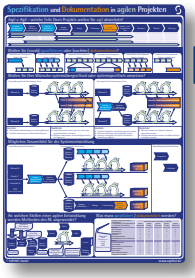


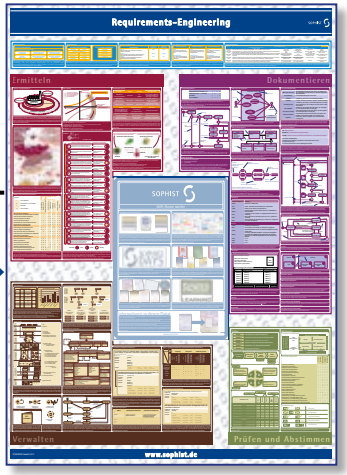
Das legendäre SOPHIST UML2 Plakat

Das neue modulare SOPHIST Requirements-Engineering Plakat...



Stakeholderlandkarte (Deutsch & Englisch)





...mit verschiedenen „Kernen“ zum modifizieren!

ANFORDERUNGEN SCHREIBEN LEICHT GEMACHT

Die Ermittlung, Dokumentation, Prüfung und Verwaltung von Anforderungen sind die Schwerpunkte der Arbeit eines Requirements-Engineers. Für die Dokumentation von Anforderungen stellen wir Ihnen in dieser Broschüre Werkzeuge vor, mit denen Sie funktionale und nicht-funktionale Anforderungen sowie mögliche Bedingungen an diese Anforderungen formulieren können. Auf dem Requirements-Engineering-Markt ist bereits ein solches Werkzeug, nämlich die SOPHIST-Satzschablone für funktionale Anforderungen, etabliert. Unter dem Begriff MASTeR führen wir nun eine Satzschablonenfamilie ein, die auch die nicht-funktionalen Anforderungen und Bedingungen abdeckt. Hinter diesem Begriff verbergen sich die folgenden Schablonen:

- eine Schablone für funktionale Anforderungen und Detaillierungsmöglichkeiten,
- drei Schablonen für nicht-funktionale Anforderungen sowie
- drei Schablonen für Bedingungsnebensätze, die sowohl funktionale als auch nicht-funktionale Anforderungen ergänzen können.

Die Lektüre der Broschüre bringt Ihnen die einzelnen Bestandteile der Schablonen näher und sensibilisiert Sie für die Dokumentation von Anforderungen. Ob Sie auf die Schablonen zurückgreifen oder sie als Anreiz für eigene Formulierungsmöglichkeiten Ihrer Anforderungen sehen, ist Ihre Entscheidung. MASTeR ist ein Angebot, das die langjährige Erfahrung der SOPHISTen zugänglich macht.