IREB Foundation Level

LE 3.2 Natürlichsprachige Arbeitsprodukte (K2)

LE 3.3 Vorlagenbasierte Arbeitsprodukte (K3)

IREB-FL Struktur

Tag	Inhalt
Tag 1	LE 1, LE 2, LE 3.1, LE 3.5 bis LE 3.8, LE 4.1 bis LE 4.2
Tag 2	LE 3.2 bis LE 3.4
Tag 3	LE 4.3 bis LE 4.4, LE 5, LE 6, LE 7
Tag 4	Zertifizierungsprüfung

IREB-FL Tagesablauf

Tag	Inhalt
9:00 – 10:45	LE 3.2 Natürlichsprachige Arbeitsprodukte LE 3.3 Vorlagenbasierte Arbeitsprodukte LE 3.3 Übung Satzschablone
10:45 – 11:00	Pause
11:00 – 12:30	LE 3.4.1 Die Rolle von Modellen im Requirements Engineering LE 3.4.2 Modellierung des Systemkontexts LE 3.4.2 Übung Use-Case Diagramm
12:30 – 13:30	Mittagspause
13:30 – 15:00	Drei Perspektiven auf Anforderungen LE 3.4.3 Modellierung von Struktur und Daten LE 3.4.3 Übung UML-Klassendiagramm
15:00 – 15:15	Pause
15:15 – 17:00	LE 3.4.4 Modellierung von Funktion und Ablauf LE 3.4.5 Zustands- und Verhaltensmodellierung LE 3.4.6 Modellierung von Zielen LE 3.4.5 Übung UML-Zustandsdiagramm

Notizen:

LE 3.2 – LE 3.3 Agenda

LE 3.2 Natürlichsprachige Arbeitsprodukte

- Probleme – Vorteile – Nachteile
- Sprachliche Effekte – Fallstricke

LE 3.3 Vorlagenbasierte Arbeitsprodukte

- Satzschablonen – Formularvorlagen – Dokumentvorlagen

LE 3.2 Natürlichsprachige Arbeitsprodukte

Arbeitsprodukt (Artefakt):

- Definition:

  „Ein aufgezeichnetes Zwischen- oder Endergebnis, das in einem Arbeitsprozess generiert wird“

- = Anforderungsdokument unterschiedlicher Darstellung, Tiefe und Lebensdauer
  - Abstraktionsniveau (Geschäfts-, Domänen-, Stakeholder-, Benutzer-, Systemanforderung)
  - Lebensdauer (kurzlebig → sich weiterentwickelnd → langlebig)
  - Darstellung (Natürlichsprachig, Vorlagen bzw. Modell basiert)
- Natürlichsprachige Anforderungen:
  - Verschriftlichte Darstellung der gesprochenen Sprache
  - Prosatext ohne zwingende formgebende Regeln

LE 3.2
Natürlichsprachige Arbeitsprodukte

Vorteile – Nachteile – Probleme

Natürlichsprachige Arbeitsprodukte

Vorteile natürlichsprachiger Dokumentation:

- Domänenübergreifend
- Ausdrucksstark
- Flexibel
  - Beliebige Detaillierungs- bzw. Abstraktionsgrade möglich
- Leicht anwendbar
  - Kein Erlernen zusätzlicher Techniken oder Werkzeuge nötig

Natürlichsprachige Arbeitsprodukte

Nachteile natürlichsprachiger Dokumentation:

- Generalität und Vagheit mancher Begriffe
- Unterspezifikation durch Mehrdeutigkeiten in der Sprache
  - Lexikalische Mehrdeutigkeit
    - Synonym (z.B. Kundennummer – Kunden-ID)
    - Homonym (z.B. Bank, Schimmel, Steuer)
  - Syntaktische Mehrdeutigkeit

    „Die letzten fünf Zielorte und die Startpunkte werden im Anzeigefeld des Navigationssystems angezeigt.“
  - Semantische Mehrdeutigkeit

    „Wird eine Scheibe am Fahrzeug beschädigt und die Innenraumüberwachung des Fahrzeugs stellt einen Eindringling fest oder eine Tür des Fahrzeugs wird ohne Fahrzeugschlüssel geöffnet, löst das Sicherheitssystem einen Diebstahlalarm aus.“

Natürlichsprachige Arbeitsprodukte

Probleme der natürlichsprachigen Dokumentation:

- Sprache basiert auf gesprochener Kommunikation zwischen interagierenden Menschen (Kontextbasiert, jederzeit Rückfrage möglich)
- Daher ist die natürliche Sprache *nicht* für eine präzise, eindeutige und umfassende Kommunikation mittels schriftlicher Dokumente ausgelegt
- Mehrdeutigkeiten, Auslassungen und Unstimmigkeiten sind schwer in geschriebenen Texten zu identifizieren

Natürlichsprachige Arbeitsprodukte

Vorgehensweisen bei natürlichsprachiger Dokumentation:

- Schreiben kurzer und gut strukturierter Sätze
- Definition und konsequente Verwendung einer einheitlichen Terminologie (Glossar)
- Vermeidung ungenauer oder mehrdeutiger Begriffe und Phrasen
- Erstellung von gut strukturierten Arbeitsprodukten
- Vermeidung von Fallstricken des technischen Schreibens:
  - Unvollständige Beschreibungen
  - Unspezifische Substantive
  - Unvollständige Bedingungen
  - Unvollständige Vergleiche
- Zu beachten:
  - Passive Formulierungen
  - Universalquantoren
  - Nominalisierungen

Natürlichsprachige Arbeitsprodukte

Lösungsansätze für natürlichsprachige Dokumentation:

Vorlagenbasierte Arbeitsprodukte:

- *Satzschablonen*

  bieten vordefinierte syntaktische Struktur für einen Satz

- *Formularvorlagen*

  stellen eine Reihe vordefinierter Felder in einem Formular zur Verfügung

- *Dokumentvorlagen*

  bieten eine vordefinierte Struktur für Anforderungsdokumente.

LE 3.2
Natürlichsprachige Arbeitsprodukte

Sprachliche Effekte – Fallstricke

Sprachliche Effekte

- Natürliche Sprache kann mehrdeutig sein:

Das Auto soll das Hindernis umfahren

- Menschen interpretieren Gesagtes/Geschriebenes unterschiedlich

→ *Missverständnissen*

- Wahrnehmung und Darstellung führt zu *Transformationseffekten*
- Sprachliche Effekte aufgrund von Transformationsprozessen:
  - Unvollständige Beschreibungen
  - Unspezifische Substantive
  - Unvollständige Bedingungen
  - Unvollständige Vergleiche
  - Passive Formulierungen
  - Universalquantoren
  - Nominalisierungen
- Ziel ist daher → möglichst *Eindeutigkeit* zu erreichen

Fallstricke natürlichsprachiger Dokumentation

Unvollständige Beschreibungen:

- Verben in natürlicher Sprache bedingen Substantive (Hauptwörter) oder Pronomen (Fürwörter).
- Auf vollständige Beschreibung ist daher zu achten.
  (Z.b.: benötigt „geben" genaue Bezeichnung für „*wer, wem, was")*

- Beispiel (unvollständige Beschreibung):

Das System muss in der Lage sein Daten auszugeben.

- Beispiel (vollständige Beschreibung):

Fallstricke natürlichsprachiger Dokumentation

Unspezifische Substantive:

- Gefahr: Unspezifische Substantive können zu unterschiedlichen Interpretationen führen
- Auf möglichst spezifische Beschreibung ist daher zu achten
- Bzw. Klarheit durch Hinzufügen von Adjektiven (Eigenschaftswort) schaffen
- Beispiel (unspezifische Substantive):

Das System muss in der Lage sein Daten auszugeben.

- Beispiel (spezifische Substantive):

Fallstricke natürlichsprachiger Dokumentation

Unvollständige Bedingungen:

- Bedingungen, die nur einen Fall beschreiben (meist den Häufigsten)
- Gefahr: Indikator für einen möglichen Informationsverlust, da Ausnahmefälle nicht beschrieben sind (Z.B.: „wenn ... dann“, „falls“, „im Falle von“, „abhängig von“, etc.)
- Beispiel (unvollständige Bedingung):

Das Restaurantsystem soll einem registrierten Gast
bei einem Alter von über 18 Jahren
alle im Lokal angebotenen Getränke anzeigen.

- Beispiel (vollständige Bedingung):

Fallstricke natürlichsprachiger Dokumentation

Unvollständige Vergleiche:

- Vergleiche, die in der gesprochenen Kommunikation scheinbar aus dem Kontext erschließbar sind (besser als, schneller als, ..)
- Auf messbare Vergleiche ist daher zu achten

- Beispiel (unvollständiger Vergleich):

Das neue Restaurantsystem
muss schneller als das Alte sein.

- Beispiel (vollständiger Vergleich):

Fallstricke natürlichsprachiger Dokumentation

Passive Formulierungen:

- Sätze im Passiv fokussieren auf das Objekt und nicht das Subjekt (bzw. haben kein Subjekt)
- Gefahr: Kann dazu führen, dass nicht deutlich wird, wer verantwortlich ist
- Besser: Anforderungen als aktive Sätze formulieren

- Beispiel (passive Formulierung):

Zur Anmeldung
werden die Login-Daten eingegeben.

- Beispiel (aktive Formulierung):

Fallstricke natürlichsprachiger Dokumentation

Universalquantoren:

- Universalquantoren sind Angaben über Häufigkeiten.
- Z.B.: „nie“, „immer“, „kein“, „jeder“, „alle“, etc.
- Gefahr: Das spezifizierte Verhalten trifft nicht auf alle so bezeichneten Objekte zu
- Besser: Jeden möglichen Fall taxativ aufzählen

- Beispiel (Universalquantor):

Das System soll in jedem Hauptmenü ein Untermenü anzeigen

- Beispiel (Universalquantor vermeiden):

Fallstricke natürlichsprachiger Dokumentation

Nominalisierung:

- Substantive (Hauptwörter) abgeleitet von Verben oder Adjektiven
- Z.B.: übermitteln → Übermittlung; einfach → das Einfache
- Gefahr: hinter Nominalisierungen können sich nicht spezifizierte Anforderungen verbergen

- Beispiel (Nominalisierung):

Bei einem Systemabsturz soll ein Neustart des Systems erfolgen

- Zu prüfen (Nominalisierung):

LE 3.2
Vorlagenbasierte Arbeitsprodukte

Satzschablonen – Formularvorlagen – Dokumentvorlagen

Vorlagenbasierte Arbeitsprodukte

Definition:

- Eine Satzschablone ist eine Vorlage (Template)
  für die syntaktische Struktur einer Anforderung
- Kann eine individuelle Anforderung im klassischen RE, oder
- eine User Story im agilen RE sein

Unterschiedliche Standards:

- ISO29148 „Systems and software engineering — Life cycle processes — Requirements engineering”
- EARS (Easy Approach to Requirements Syntax)
- Rupp & Co: „Satzschablonen in Requirements-Engineering und Management“
- Agile Anforderungen: User Storys

Vorlagenbasierte Arbeitsprodukte

Vorteile:

- Klare, wiederverwendbare Strukturen
- Hilfestellung bei der Erfassung der wichtigsten Informationen
- Einheitliche Struktur von Anforderungen und Arbeitsprodukten
- Verbesserung der Gesamtqualität von Anforderungen und Arbeitsprodukten

Nachteile:

- Gefahr einer ausschließlich formalisierten Herangehensweise
- Daher geringere Konzentration auf den Inhalt
- Aspekte, die nicht in der Vorlage enthalten sind, könnten vergessen werden

Satzschablone

Satzschablone nach ISO29148:

[<Bedingung>] <Subjekt> <Aktion> <Objekte> [<Einschränkung>]

Beispiel:

Wenn eine gültige Karte erkannt wird,	<Bedingung>
soll das System	<Subjekt>
die Meldung „Geben Sie Ihre PIN ein”	<Aktion>
auf dem Dialogbildschirm anzeigen,	<Objekte>
und zwar innerhalb von 200ms.	<Einschränkung>

Satzschablone

Satzschablone nach EARS (Easy Approach to Requirements Syntax):

Das <Systemname> soll <Systemantwort>
Das Bibliothekssystem soll Kunden erfassen können.

Ereignisgesteuerte Anforderungen nach EARS:

SOBALD|NACHDEM <optionale Vorbedingungen> <Auslöser> soll das <Systemname> <Systemantwort>
Nachdem der PC hochgefahren ist soll das Bibliotheksystem den Login Screen zeigen.

Unerwünschtes Verhalten (Beschreibung zu vermeidende Situationen):

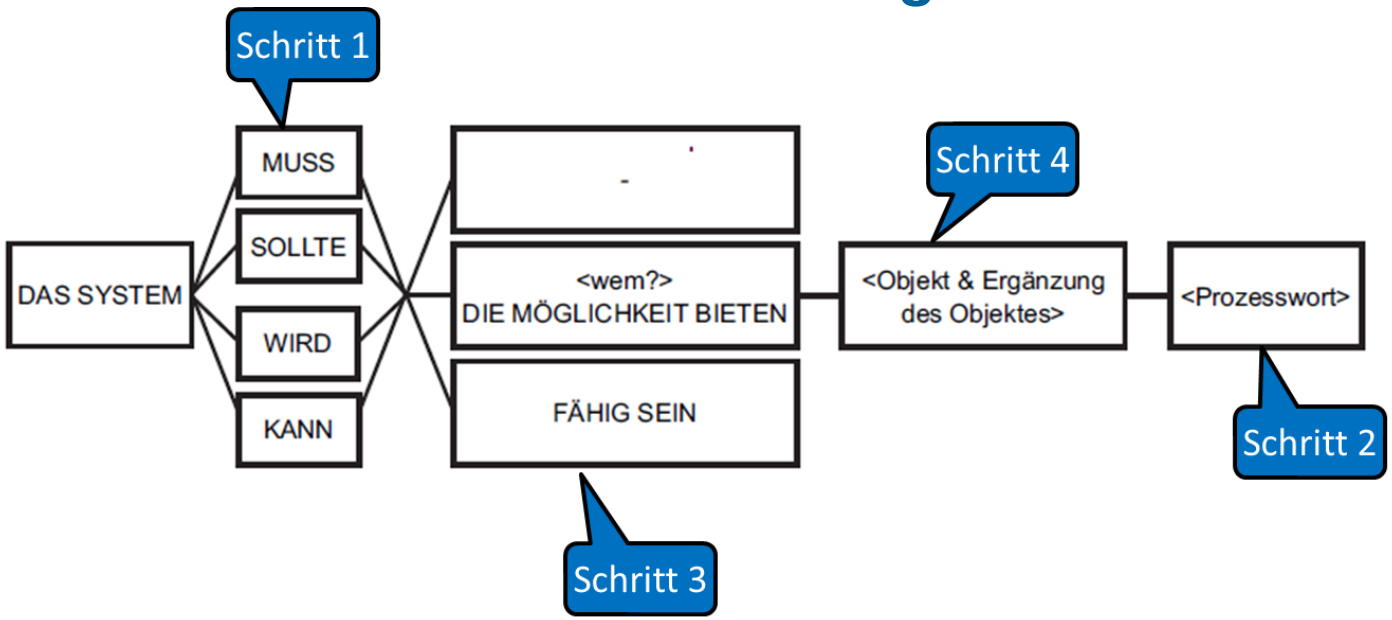
WENN <optionale Vorbedingungen> <Auslöser>, DANN soll das <Systemname><Systemreaktion>
Wenn der Kunde seine Login Daten 3mal falsch eingegeben hat,
dann soll das Bibliotheksystem einen Alarmton ausgeben.

Zustandsorientierte Anforderungen (gelten nur in bestimmten Zuständen):

SOLANGE <in einem bestimmten Zustand> soll das <Systemname> <Systemantwort>
Solange der Kunde eingeloggt ist, soll das Bibliotheksystem den Screensaver deaktiviert halten.

Satzschablone

Satzschablone für funktionale Anforderungen nach RUPP & Co:



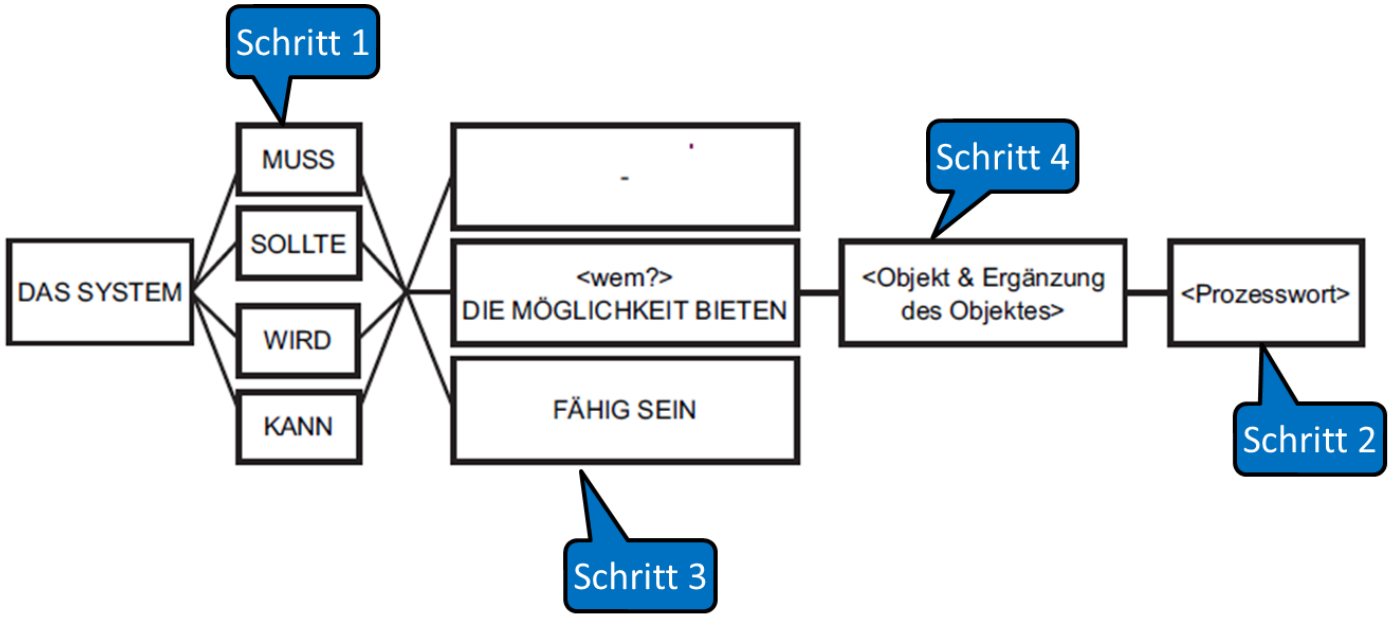
Schritt 1: Festlegen der rechtlichen Verbindlichkeit (Wichtigkeit/Priorität)

Schritt 2: Kern der Anforderung (Funktionalität) festlegen (Prozess)

Schritt 3: Art der Funktionalität definieren (Benutzeraktivität, Selbstständige Systemaktivität)

Schritt 4: Objekt definieren

Satzschablone

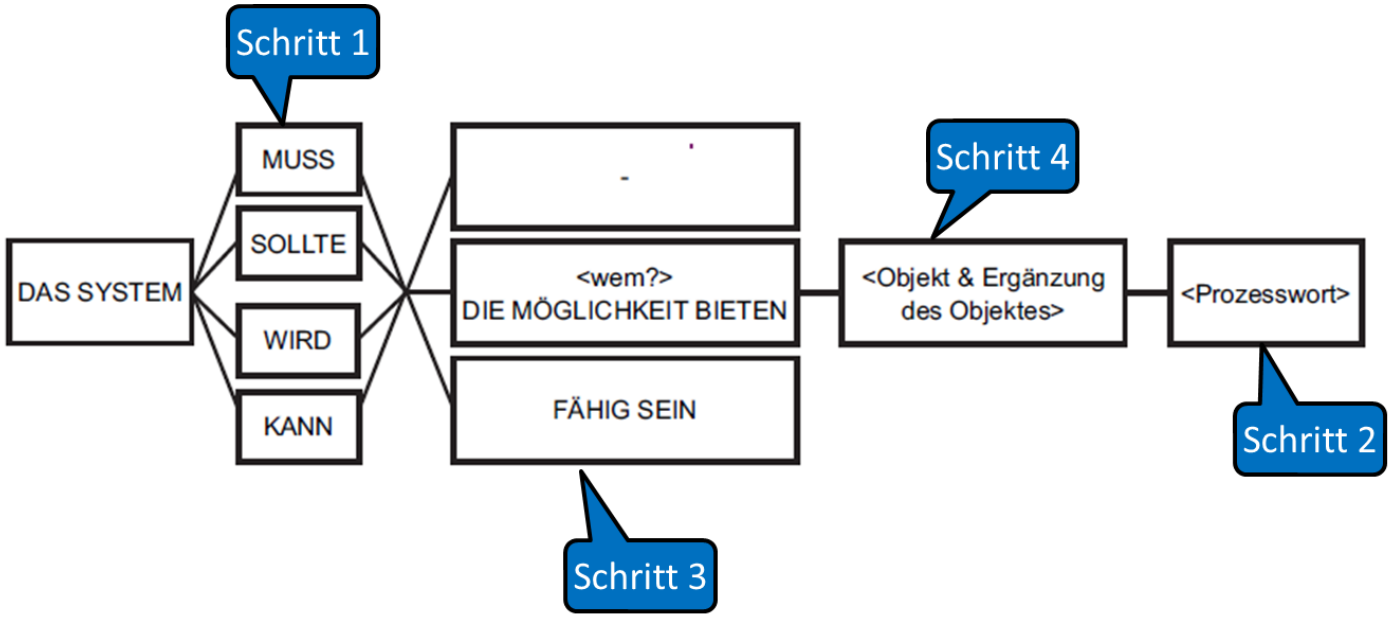


Beispiel:

Start: System konkret benennen → *Bibliotheksystem*
Schritt 1: Priorität festlegen → *muss*

Das **Bibliothekssystem muss …**

Satzschablone

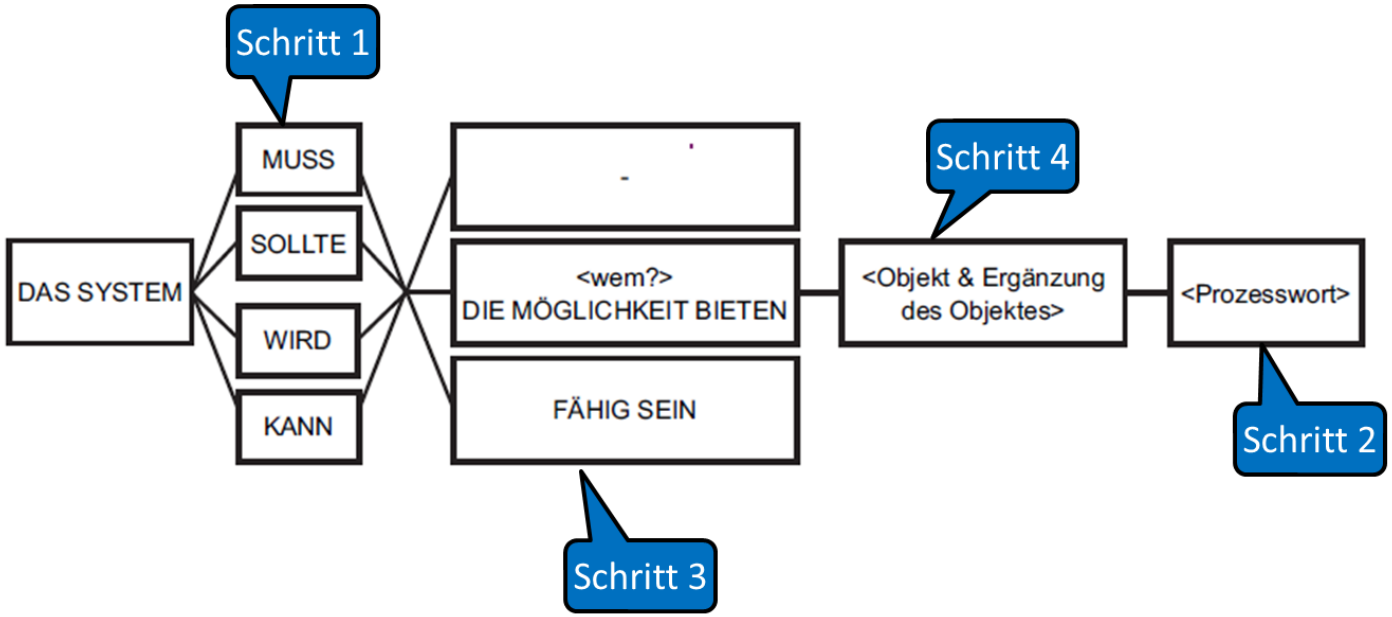


Beispiel:

Schritt 2: Funktionalität festlegen → *drucken*

Das Bibliothekssystem muss **… drucken.**

Satzschablone

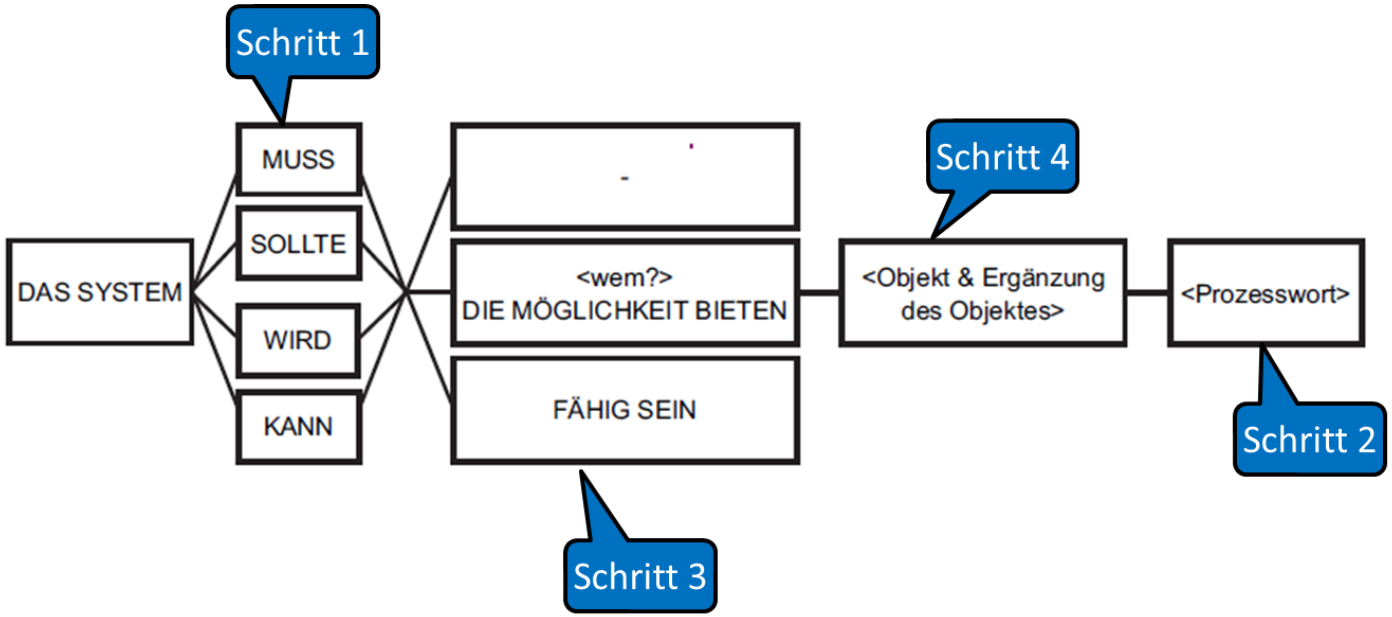


Beispiel:

Fortsetzung: <Akteur> konkret benennen → *Bibliothekar*
Schritt 3: Art der Funktionalität definieren → *die Möglichkeit bieten*

Das Bibliothekssystem muss **dem Bibliothekar die Möglichkeit bieten** … zu drucken.

Satzschablone

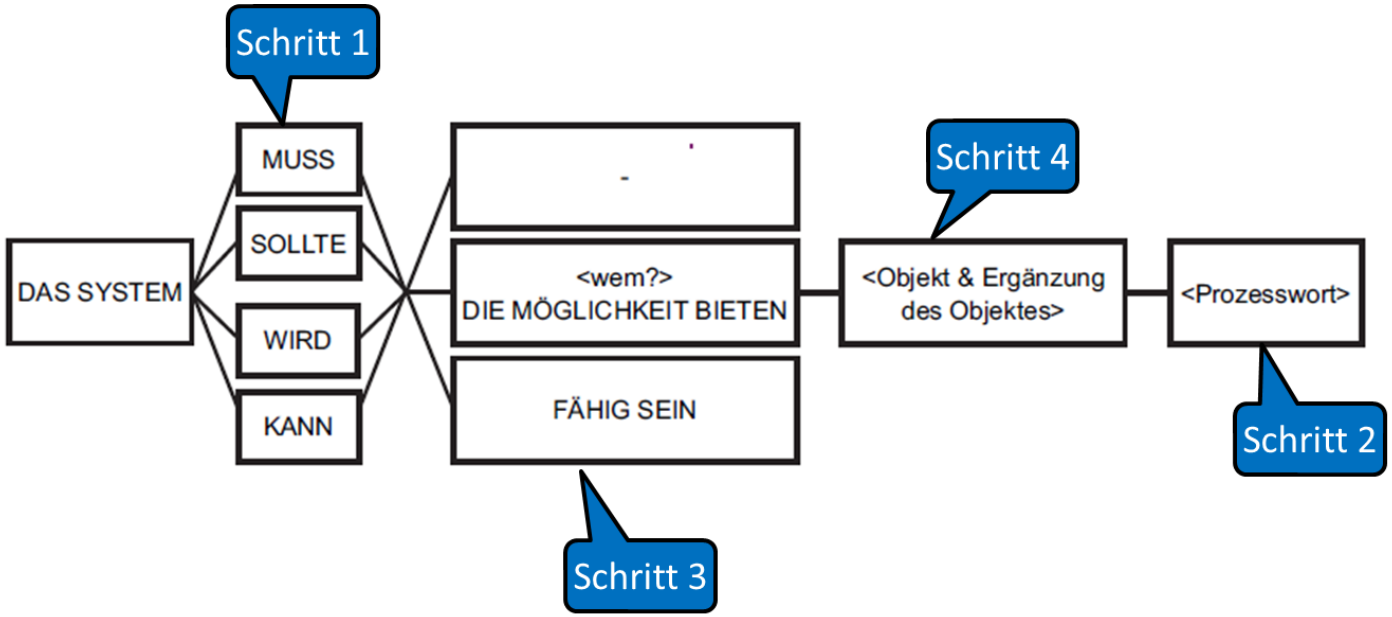


Beispiel:

Schritt 4: Objekt definieren → *Kundendaten*

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

Satzschablone



Beispiel:

Abschluss: Formulierung konkretisieren

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

Satzschablone

Agile User Storys:

- sind kurze, einfache Beschreibungen eines Bedürfnisses (Need)
  aus der Perspektive der Person, die die neue Fähigkeit wünscht
- sind Teil des agilen Ansatzes, bei dem der Fokus vom Schreiben von Anforderungen auf das Reden über sie verlagert wird
- bestehen aus ein oder zwei geschriebenen Sätzen und, was noch wichtiger ist, eine Reihe von Gesprächen über die gewünschte Funktionalität.

As a <type of user>,
I want <some goal>
[so that <some reason>].

Als <Rolle>
möchte ich <Ziel/Wunsch>,
[damit ich <Nutzen>].

Satzschablone

Agile User Storys Beispiele:

Als Kunde
möchte ich
über neue Produkte auf der Startseite informiert werden,
damit ich
kein interessantes Angebot versäume.

Als Vertragshändler
möchte ich
alle Gebrauchtwagen auflisten können,
damit ich
diese meinen Kunden anbieten kann.

Als Lektor
möchte ich
neue Kurse anlegen können,
damit ich
für jeden Kurs getrennt Informationen bereitstellen kann.

Als Student
möchte ich
Informationen über meinen Studienerfolg
damit ich
mich schnell orientieren kann was fehlt.

Satzschablone

Agile Akzeptanz Kriterien:

- Agile **User Storys** werden durch Akzeptanzkriterien verfeinert
- Müssen spätesten vor Entwicklungsbeginn abgestimmt sein
- **Format** Acceptance Kriterien:

Given *<some precondition>*
When *<I do some action>*
Then *<I expect some result>*

Gegeben *<Vorbedingung>*
Wenn *<Aktion>*
Dann *<Erwartetes Ergebnis>*

Satzschablone

Beispiel Agiles Akzeptanz Kriterium:

Als Bibliothek Kustos
möchte ich
Bücher verwalten können,
damit ich
Jederzeit einen Überblick über die Bücher der Bibliothek habe

Akzeptanzkriterium: 1
Bücher hinzufügen

Gegeben
- Kustos ist eingeloggt
- Und im Menü Bücher hinzufügen

Wenn
- Bücherdaten eingegeben wurden
- Und Buch speichern gedrückt wurde

Dann
- Ist das neue Buch im System gespeichert
- Und eine Liste der zuletzt hinzugefügten Bücher wird angezeigt

Formularvorlage

Formularvorlagen:

- werden zur Strukturierung von Arbeitsprodukten verwendet
- Haben einen statischen Aufbau mit vordefinierten Eingabefeldern
- Anwendung z.B. bei Use-Case Beschreibungen:

Thema	Beschreibung
Identifikation	UC_47-11
Status:	Review
Name:	Bücher oder Medien ausleihen
Akteure:	Kunde, BibliothekarIn, Kundendatenbank
Beschreibung:	Kunde leiht Bücher oder Medien in einer Bibliothek aus BibliothekarIn verwaltet Kunden und Leihobjekte
Vorbedingung:	BibliothekarIn ist am System angemeldet
Auslöser:	BibliothekarIn nach Kundenwunsch
Ablauf:	• Leihobjekt reservieren, wenn verfügbar • Leihobjekt verlängern, wenn nicht bereits vorbestellt • Leihobjekt in Kundendatenbank vermerken und verleihen • Leihobjekt zurücknehmen und in Kundendatenbank vermerken • BibliothekarIn verwaltet Kunden und Leihobjekte
Ergebnis:	• Leihobjekt verliehen • Leihobjekt verlängert • Leihobjekt zurückgenommen • Kunden und Leihobjekte administriert

Dokumentvorlage

Dokumentvorlagen

- helfen bei der systematischen Strukturierung von Anforderungsdokumenten
- Können für gleichartige Projekte gut wiederverwendet werden
- **ISO 29148** definiert eine Standardvorlage für System-Anforderungsspezifikationen

Teil I: Einführung

1. Zweck des Systems
2. Umfang der Systementwicklung
3. Stakeholder

Teil II: Systemübersicht

4. System Vision und Ziele
5. Systemkontext und -grenze
6. Gesamtstruktur des Systems
7. Merkmale der Benutzer

Teil III: Systemanforderungen

Hierarchisch nach der Systemstruktur organisiert, unter Verwendung eines hierarchischen Nummerierungsschemas für Anforderungen

Pro Subsystem/Komponente:

- Funktionale Anforderungen (Struktur und Daten, Funktion und Ablauf, Zustand und Verhalten)
- Qualitätsanforderungen
- Randbedingungen
- Schnittstellen

Referenzen

Anhänge

Glossar (falls nicht als eigenständiges Arbeitsprodukt verwaltet)

Annahmen und Abhängigkeiten

Übung

Übung Satzschablone:

- **Kundenbedürfnis:** ***Bibliothekssystem***
  - Verwaltet Bücher
  - Verleiht Bücher an Kunden

- **Aufgabe** (Gruppenarbeit max. 3 Teilnehmer)
  - Erstellen Sie mindestens 2 ***Anforderungen*** auf Basis
    - klassischer Satzschablone (Rupp & Co)
    - Agiler Userstorys mit Akzeptanzkriterium
  - Zur Vergleichbarkeit und Diskussion
    soll die klassische und agile Anforderung paarweise erstellt werden
  - Präsentation und Diskussion der Ergebnisse

Literatur

- Basiswissen Requirements Engineering:
  Aus- und Weiterbildung nach IREB-Standard zum
  Certified Professional for Requirements Engineering Foundation Level
  Chris Rupp & Klaus Pohl; dpunkt.verlag GmbH; 2015

- Handbuch für das CPRE Foundation Level nach dem IREB-Standard
  Aus- und Weiterbildung zum Certified Professional for Requirements Engineering (CPRE) Foundation Level
  Martin Glinz, Hans van Loenhoud, Stefan Staal, Stan Bühne
  https://www.ireb.org/content/downloads/4-cpre-foundation-level-syllabus-3-0/cpre_foundationlevel_syllabus_de_v.3.0.1.pdf

- Requirements-Engineering und -Management:
  Aus der Praxis von klassisch bis agil
  Christine Rupp, die SOPHISTen
  Carl Hanser Verlag GmbH & Co. KG; 2020

- Schablonen für alle Fälle
  die SOPHISTen
  https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/Wissen_for_free/MASTeR_Broschuere_3-Auflage_interaktiv.pdf

Urheberrechtshinweis

Die Urheberrechte dieser Unterlage liegen beim Autor dieses Werkes.

Die Nutzung der bereitgestellten Unterlagen ist ausschließlich zur eigenen Verwendung im Rahmen des Kurses gestattet.
Eine Weitergabe an andere, nicht für den Kurs eingetragene Personen, wird ausdrücklich untersagt.

© Robert Jonas