Requirements-Engineering

SOPHIST

Anforderungsarten

Funktionale Anforderungen

Nicht-funktionale Anforderungen

- Technologische Anforderungen
- Anforderungen an die Benutzungsoberfläche
- Anforderungen an durch-zuführende Tätigkeiten
- Qualitäts-anforderungen
- Anforderungen an sonstige Lieferbestandteile
- Rechtlich-vertragliche Anforderungen

Anforderungen lassen sich nach ihrer Art einteilen. Die Art beschreibt eine Unterscheidung von Anforderungen nach einer inhaltlichen Kategorie/Gruppe. Der oft verwendete Begriff *nicht-funktionale Anforderung* spiegelt sich in obiger Aufstellung auch wider: Jede Anforderung, die keine funktionale Anforderung ist, ist nicht-funktional.

Qualitätskriterien

Für jede einzelne Anforderung		Für das Anforderungsdokument
Vollständig	Atomar	Vollständig
Technisch lösungsneutral		Konsistent
Konsistent	Prüfbar	Erschwinglich
Notwendig	Verfolgbar	Abgegrenzt
Realisierbar	Eindeutig	

Qualitätskriterien helfen zu verstehen, welche Kriterien für die einzelne Anforderung bzw. für das gesamte Anforderungsdokument besonders wichtig und zu beachten sind. Anhand dieser Kriterien kann beurteilt werden, wie qualitativ hochwertig eine Anforderung/ein Anforderungsdokument ist.

Rechtliche Verbindlichkeit

Verbindlichkeit	Deutsches Schlüsselwort	Englisches Schlüsselwort
Pflicht Klassifiziert eine rechtlich verbindliche, zwingend zu erfüllende Anforderung.	muss	shall
Wunsch Wird für Anforderungen verwendet, von deren Umsetzung unter bestimmten Umständen abgesehen werden kann.	sollte	should
Absicht Ermöglicht es, zukünftige Rahmenbedingungen und Anforderungen bereits anzukündigen.	wird	will

Die rechtliche Verbindlichkeit beschreibt den Grad der Bedeutung, den der Stakeholder den Angaben in der Anforderungsspezifikation beimisst. Üblicherweise wird die rechtliche Verbindlichkeit durch Verwendung bestimmter Modalverben (Schlüsselwörter) ausgedrückt.

Perspektiven auf Anforderungen

Strukturperspektive	Funktionsperspektive	Verhaltensperspektive
Inhalt Z. B. (Daten-)Strukturen eines technischen Systems oder Subsysteme.	**Inhalt** Z. B. Funktionalitäten eines technischen Systems, die dem Nutzer zur Verfügung stehen.	**Inhalt** Z. B. Verhalten eines technischen Systems aufgrund eingehender Ereignisse.
Dokumentation Z. B. mit UML-Klassendiagramm oder mittels natürlichsprachlicher Anforderung.	**Dokumentation** Z. B. mit UML-Aktivitätsdiagramm oder mittels natürlichsprachlicher Anforderung.	**Dokumentation** Z. B. mit UML-Zustandsdiagramm oder mittels natürlichsprachlicher Anforderung.

Anforderungen an einen Betrachtungsgegenstand können aus drei verschiedenen Perspektiven betrachtet werden. Unterschiedliche Dokumentationstechniken bieten die Möglichkeit die Anforderungen einer bestimmten Perspektive zu explizieren.

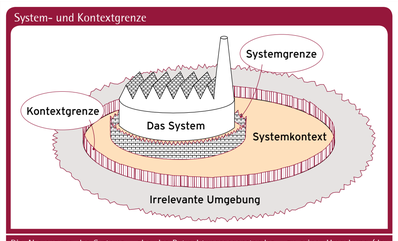
Detaillierungsebenen

Ebenen	Zugeordnete Begrifflichkeiten: aus herkömmlichen Vorgehen, wie z. B. RUP oder V-Model	aus agilen Vorgehen: Scrum	Adaptive Software Development	Lean Software Development	Crystal
Spezifikations-level 0	Grobe Systembeschreibung, Systemziele, Systemüberblick, Vision, Introduction, Mission Statement	Sprint Goal, Product Vision	Project Vision, Project Data Sheet	Release Plan	Mission Statement
Spezifikations-level 1	Anwendungsfall (Use-Case), User Story, (Anwendungs-) Szenario, Fachkonzept, Funktionsbeschreibung, Funktionsgliederung, fachliche Anforderung, Organisational Requirement, Featureliste, Kontextabgrenzung	Epic, Theme	Product Specification Outline	User Story, Backlog Item, Use-Case	Actor Goal List, User Role Model, Use-Case
Spezifikations-level 2	Anwenderforderung, Nutzeranforderung, Operational Concept Description, Operational Requirements Description, Interface Requirements Specification, Lastenheft, Sollkonzept, Grobspezifikation, Operational Requirement, betriebliche Anforderung, Testfälle, Featureliste	User Story, Technical User Story, Backlog Item, Acceptance Criteria, Definition of Done	Product Specification Outline	User Story, Backlog Item, Use-Case, Iteration Plan, Qualifier, Test Case	Common Domain Model, Test Case, Iteration Plan
Spezifikations-level 3	Detaillierte Anwenderforderungen, Technische Anforderung, Schnittstellenübersicht, Schnittstellenbeschreibung, System Requirements Specification, Interface Requirements Specification, Systemanforderung, Feinspezifikation, Testfälle, Featureliste	Technical User Story, Backlog Item, Acceptance Criteria, Definition of Done	Technical User Story, Backlog Item, Acceptance Criteria, Definition of Done	Qualifier, Test Case	Architecture Description, Test Case
Spezifikations-level 4	Komponentenanforderungen, Technische Anforderung, Schnittstellenübersicht, Schnittstellenbeschreibung, Software Requirement Specification, Interface Design Description, Pflichtenheft, Feinspezifikation, Modulanforderung, Testfälle	Technical User Story, Backlog Item, Acceptance Criteria	Technical User Story, Backlog Item, Acceptance Criteria	Test Case	Architecture Description, Test Case

Anforderungen können in einer Verfeinerungshierarchie zueinander stehen und so einer von fünf verschiedenen Detaillierungsebenen/Spezifikationslevel zugeordnet werden. Die Anforderungen auf jeder Ebene beschreiben dabei den Betrachtungsgegenstand vollständig, je nach Ebene sehr abstrakt oder verfeinert.

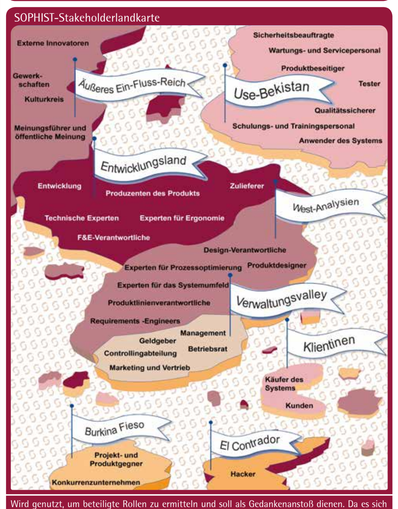
Ermitteln

System- und Kontextgrenze



Die Abgrenzung des Systems – also des Betrachtungsgegenstandes – zu seiner Umgebung (der Kontext) wird genutzt, um klar zu definieren, an welchen Betrachtungsgegenstand Anforderungen gestellt werden müssen und dürfen und an welche Gegenstände in der Umgebung nicht. Durch die dadurch entstehende Systemgrenze werden hier bereits Schnittstellen zwischen dem System und den externen Partnern sichtbar.

SOPHIST-Stakeholderlandkarte



Wird genutzt, um beteiligte Rollen zu ermitteln und soll als Gedankenanstoß dienen. Da es sich um eine Sammlung aus vielen Jahren der Stakeholderanalyse handelt, sollten viele Rollen bereits auf dieser Landkarte verzeichnet sein. Da es jedoch schwer ist, die ganze Welt zu überblicken, sind Erweiterungen erlaubt und jederzeit willkommen.

SOPHIST-Ermittlungstechnikenmatrix

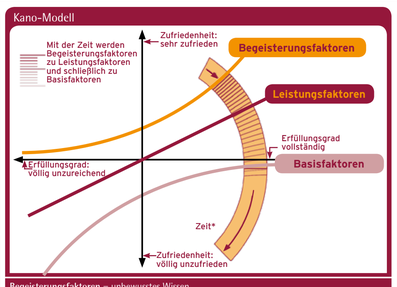
Legende:

–	nicht empfohlen
0	kein Einfluss=> ist anwendbar
+	empfohlen
++	sehr empfohlen

	Brainstorming	Methode 6-3-5	Wechsel der Perspektive	Feldbeobachtung	Apprenticing	Fragebogen	Interview	Systemarchäologie	Reuse
Menschliche Einflussfaktoren									
Geringe Motivation der Stakeholder (aktiv mitzuwirken)	–	–	–	+	–	0	+	++	++
Schlechte kommunikative Fähigkeiten	–	–	–	++	++	–	+	++	++
Geringes Abstraktionsvermögen	–	–	–	++	++	0	+	++	++
Viele verschiedene Meinungen	+	++	+	++	++	+	0	0	0
Machtgefälle zwischen beteiligten Personen	–	+	–	0	0	0	0	0	0
Problematische Gruppendynamik	–	+	+	0	0	0	0	0	0
Organisatorische Einflussfaktoren									
Entwicklung für den komplexen Markt	++	+	+	–	–	++	0	+	0
Fixiertes, knappes Projektbudget	++	++	+	+	+	–	+	–	++
Hohe örtliche Verteilung der Stakeholder	–	0	–	0	0	++	0	0	0
Schlechte zeitliche Verfügbarkeit der Stakeholder	+	+	–	+	–	+	++	++	++
Hohe Anzahl der Stakeholder	+	–	+	0	–	++	0	0	0
Fachliche/Inhaltliche Einflussfaktoren									
Hohe Kritikalität des Sachverhalts	0	0	+	++	–	+	+	++	+
Großer Systemumfang	0	0	0	+	–	+	–	++	+
Keine Erfahrung im Fachgebiet	0	0	0	–	+	–	+	++	+
Grobe Anforderungen gesucht	++	++	+	+	0	+	++	–	0
Detaillierte Anforderungen gesucht	+	+	+	+	++	–	+	++	+
Nicht funktionale Anforderungen gesucht	0	0	0	0	+	–	+	+	+
Hohe Komplexität des Sachverhalts	0	0	0	+	–	–	+	+	+

Sie wird eingesetzt, um anhand bewerteter Einflussfaktoren und Randbedingungen die am besten geeigneten Ermittlungstechniken der jeweiligen Situation zu identifizieren.
Tipp: Zwei bis drei verschiedene Ermittlungstechniken aus unterschiedlichen Bereichen verwenden, um das beste Ergebnis zu erzielen. Ebenso darauf achten, dass unbewusstes, bewusstes und unterbewusstes Wissen abgedeckt sind.

Kano-Modell



Begeisterungsfaktoren = unbewusstes Wissen
>> Müssen als neue Ideen/Innovationen erst neu erarbeitet/erfunden werden.
Leistungsfaktoren = bewusstes Wissen
>> Werden explizit gefordert und genannt.
Basisfaktoren = unterbewusstes Wissen
>> Werden als selbstverständlich erachtet und oft nicht (mehr) explizit gefordert.

SOPHIST-REgelwerk Kurzübersicht

	Regel zur Anforderungsformulierung	Defektbafte Signalwörter
Prozesse prüfen	1 Formulieren Sie jede Anforderung im Aktiv und identifizieren Sie den Akteur der Anforderung	werden; wurden; sind; ...
	2 Drücken Sie Prozesse durch eindeutige/definierte Vollverben aus.	angeben; bereitstellen; verhindern; ...
	3 Lösen Sie Nominalisierungen auf, die nicht exakt definiert sind, und schreiben Sie pro Nominalisierung eine oder mehrere neue Anforderungen mit gutem Vollverb.	Rückgabe; Eingabe; Speicherung; Archivierung: ...
	4 Lösen Sie Funktionsverbgefüge auf und bestimmen Sie den konkret geforderten Prozess, der das Systemverhalten mit einem „guten" Vollverb darstellt.	zur Verfügung stellen; zu Ende bringen; eine Veränderung erfahren; ...
	5 Identifizieren Sie die Vollverben und schreiben Sie für jedes Vollverb genau einen Anforderungssatz.	anzeigen und anschließend ausdrucken; ...angeben und speichern; ...
	6 Analysieren Sie fehlende Informationen zum Vollverb und ergänzen Sie diese falls notwendig. Stellen Sie W-Fragen: Was? Wem? Wie? Wann? Wie oft?	<wer>, <was>, <wann> anzeigen; <wer>, <was>, <wohin> übermitteln; ...
Eigenschaften prüfen	7 Analysieren Sie fehlende Informationen zum Eigenschaftswort, welches sich aus einem Vollverb ableitet, und ergänzen Sie sie notwendigenfalls.	erzeugte; versandte; abgezeichnete; ...
	8 Formulieren Sie Eigenschaftswörter mess- und testbar. Achten Sie dabei auf Adjektive, Adverbien und insbesondere Vergleiche und Steigerungen.	schnell; effektiv; performant; besser; intuitiv; leichter; ...
	9 Formulieren Sie eigene Anforderungen für nicht-funktionale Aspekte, wenn diese eigenständig behandelt werden sollen oder als übergeordnet gelten.	*Qualitätsaspekte, wie z. B. Zeitverhalten, Verfügbarkeit, ...*
Mengen prüfen	10 Hinterfragen Sie verwendete Zahl- und Mengenwörter und prüfen Sie, ob mit ihnen die zu bezeichnende Menge der Objekte auch genau erfasst wurde.	alle; jeder/jeden; immer; kein; 7; ...
	11 Klären Sie fehlende Zahl- und Mengenwörter. Prüfen Sie, ob Zahl- oder Mengenwörter hinzugefügt werden können. Ist dies der Fall, bestimmen Sie die genau gültige Menge an Objekten.	soll <welchen> Benutzern ermöglichen; <welche> Daten genau; ...
	12 Hinterfragen Sie schwammig formulierte Substantive und stellen Sie fest, welche Objekte oder Akteure genau gemeint sind.	der Anwender; die Meldung; die Funktion; ...
Redundante Infos prüfen	13 Klären Sie Mögliches und Unmögliches und ersetzen Sie deren beschreibende Formulierungen. Hinterfragen Sie, was das geforderte Verhalten oder die Eigenschaft möglich oder unmöglich macht, und bestimmen Sie die identifizierte Logik.	muss möglich sein; darf nicht möglich sein; ...
	14 Extrahieren Sie Nebensätze, die für die Anforderung nicht notwendige Informationen enthalten.	weil; damit; um; so dass; ...
	15 Kürzen oder entfernen Sie floskelhafte Wörter oder Redewendungen, die für Ihre Anforderungen irrelevant sind und präzisieren sie. Vereinfachen Sie komplizierte Redewendungen und unnötig verschachtelte Sätze.	für den Fall; gelb in der Farbe; klein in der Größe; ...
Gesamtbild prüfen	16 Klären Sie Ausnahmen vom Normalverhalten des Systems und erweitern Sie die Anforderung bzw. formulieren Sie ggf. eine neue.	sollen; sollte; müssen; es ist notwendig; ...
	17 Anforderungen mit unvollständigen Bedingungsstrukturen sollten überprüft und ausformuliert oder durch eine weitere Anforderung beschrieben werden. Bestimmen Sie das Systemverhalten für jede noch nicht beschriebene Bedingung.	wenn ... dann; falls .. dann; für den Fall dass; sollte; ...
	18 Für jede nicht beschriebene implizite Annahme müssen eine oder mehrere zusätzliche Anforderungen geschrieben werden.	falls; nachdem; angezeigte View; Kundendaten; ...

Priorität: = hoch = mittel = niedrig

Vor allem bei der Ermittlung von Anforderungen genutzt, ist das SOPHIST-*REgel*werk jedoch fast ein Alleskönner. Es kann auch als Prüftechnik für bereits bestehende Anforderungen eingesetzt werden. Oder es hilft bei der Dokumentation von Anforderungen.
Eigentlich immer einsetzbar, sobald es um Texte oder Sprache geht – egal ob geschrieben oder gesprochen. Ein tolles Mittel also für die natürlichsprachliche Anforderungsanalyse.

Stakeholder-Tabelle

Funktion	Name	Kontakt	Verfügbarkeit
Bibliothekar Leiter der Bibliothek	Herr Bauer	Tel. 409000	Von 9–19 Uhr telefonisch erreichbar, Mitarbeit zu 30% möglich, Nürnberg
Administrator (Wartungs- und Schulungspersonal)	Herr Heiner	Heiner@gmx.net	Per Mail, immer erreichbar, Verfügbarkeit 50%, Nürnberg
Product-Owner	Paul Ottmer	po@ottmer.de	Per Mail und tel. tagsüber, Verfügbarkeit 100%, Nürnberg

Wissen	Interesse & Ziele	Relevanz
Kennt das Altsystem aus der Anwendersicht, soll mit dem System arbeiten	Vereinfachung der Ausleihprozesse	Fachlicher Entscheider
Vertraut mit vergleichbarer Verwaltungssoftware	Stabiles System, geringer Wartungsaufwand	Informationslieferant bzgl. Wartungsanforderungen
Koordinator für die Inputs der Stakeholder	ROI des Systems sicherstellen	Entscheider – als Koordinator der Stakeholderanforderungen

Ist eine Sammlung aller Beteiligten am Vorhaben/Projekt mit deren wichtigsten Kontaktdaten, etc. Sie fungiert als (Projekt-)Adressbuch und ist für alle Beteiligten öffentlich zugänglich. Man findet darin jede/jeden, die/den man irgendwann im Projekt benötigt. Deshalb muss sie auch stetig aktuell gehalten werden. Die SOPHIST-Stakeholderlandkarte gibt Aufschluss über mögliche Rollen.

Transformationsprozesse/-effekte

Die eingeschränkte persönliche Wahrnehmung führt zur Wahrnehmungstransformation.

Der sprachliche Ausdruck des persönlichen Wissens führt zur Darstellungstransformation.

Die Transformationseffekte lassen sich in drei Kategorien einteilen - egal ob sie bei der Wahrnehmung oder bei der Wissensdarstellung auftreten:
Tilgung ist ein Indikator für unvollständige Sätze.
Generalisierung ist ein Indikator für fehlerhafte Verallgemeinerungen.
Verzerrung ist ein Indikator für realitätsverfälschende Aussagen.
Das SOPHIST-*REgel*werk geht auf die konkreten Effekte dieser drei Kategorien mit seinen Regeln näher ein. Mit diesen werden die konkreten Effekte untersucht.

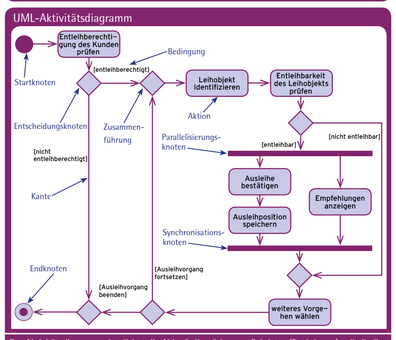
Dokumentieren

Entscheidungstabelle

Leihobjekt.Musterexemplar == ja	j	j	n	n
Leihobjekt.Status == reserviert	j	n	j	n
entleihbar	–	–	–	x
nicht entleihbar	x	x	x	–

Zusammenfassung von Bedingungskombinationen zu einem fachlichen Zustand z. B.
Falls [Leihobjekt entleihbar], ...
vgl. auch DMN (decision model and notation) - bei der OMG (Object Management Group) ist der Standard unter http://www.omg.org verfügbar.

UML-Aktivitätsdiagramm



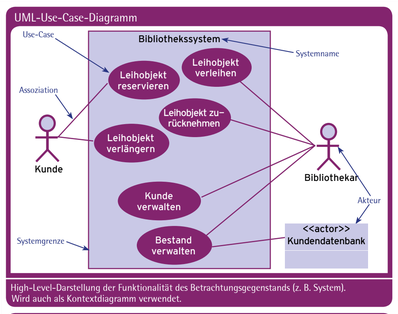
Das Aktivitätsdiagramm visualisiert die Ablaufreihenfolge von Schritten (Funktionen) vollständig, d.h. inklusive Alternativen, Fehler, Ausnahmen, ...
Das Einsatzspektrum reicht von Geschäftsprozessen, über die Verfeinerung von Use Cases oder Aktionen in anderen Aktivitätsdiagrammen, bis zur Darstellung von Abläufen in Source Code.

SysML-Diagramme

- Blockdefinitionsdiagramm
- Anforderungsdiagramm
- und weitere

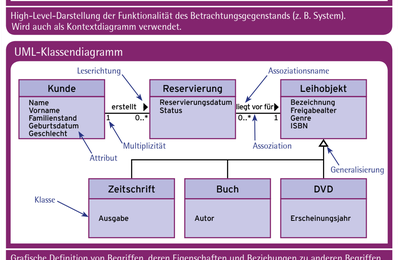
Sprache für die Modellierung von Systemen, nicht nur Software. Die SysML (Systems Modeling Language) als Erweiterung zur UML 2 ergänzt diese um weitere Diagramme bzw. passt einige vorhandene Diagramme an die speziellen Bedürfnisse der Systemmodellierung an.
Der Standard ist bei der OMG (Object Management Group) verfügbar unter www.omg.org

UML-Use-Case-Diagramm



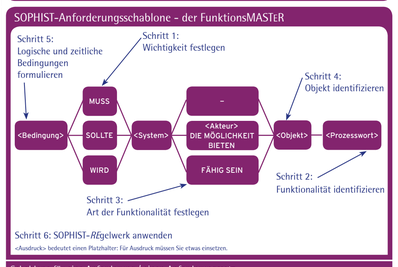
High-Level-Darstellung der Funktionalität des Betrachtungsgegenstands (z. B. System).
Wird auch als Kontextdiagramm verwendet.

UML-Klassendiagramm



Grafische Definition von Begriffen, deren Eigenschaften und Beziehungen zu anderen Begriffen. Auch wenn Klassendiagramme in der Implementierung eine große Rolle spielen - hier nur als Begriffsmodell/Informationsmodell auf fachlicher/logischer Ebene.

SOPHIST-Anforderungsschablone – der FunktionsMASTeR



Schritt 6: SOPHIST-*REgel*werk anwenden
«Ausdruck» bedeutet einen Platzhalter: Für Ausdruck müssen Sie etwas einsetzen.

Schablone für eine Anforderung / einen Anforderungssatz.
Hier der FunktionsMASTeR mit Bedingung für eine funktionale Anforderung.
Weitere Schablonen finden Sie in der Broschüre „MASTeR - Schablonen für alle Fälle".

Use-Case-Beschreibung

Name des Use-Case	Eindeutige Benennung des Use-Cases. Der Name muss mit dem Namen des Use-Cases im Use-Case-Diagramm übereinstimmen.
Beschreibung	Kurze Beschreibung des Use-Cases.
Akteure	Aufzählung aller am Use-Case beteiligten Rollen.
Auslöser	Ereignis, welches dazu führt, dass ein Use-Case abläuft.
Vorbedingungen	Aufzählung aller Inputs, sowie aller logischen und zeitlichen Bedingungen, die zu Beginn des Use-Cases erfüllt sein müssen.
Normalablauf	Aufzählung der Schritte des Use-Cases, um das Ergebnis von fachlichem Wert zu erzeugen.
Alternativer Ablauf	Abweichende Schritte des Use-Cases, um das Ergebnis von fachlichem Wert zu erzeugen.
Ablauf mit Fehlern	Abweichende Schritte des Use-Cases, bei denen das Ergebnis von fachlichem Wert nicht erzeugt wird.
Ergebnis	Beschreibt, welches Ergebnis von fachlichem Wert erzeugt wird.

Vorlage für Use-Cases. Wird meist auf die individuellen Bedürfnisse des Projekts, des Unternehmens oder des Vorgehens angepasst. Z. B. Qualitätsaspekte (Qualitätsanforderungen), Ergebnis, Alternativen, Ausnahmen, Fehler, ...

User-Story-Schablone

Als <Benutzerrolle>

möchte ich <Funktionalität/Systemverhalten>,

so dass <fachlicher Wert für den Benutzer/Kunden bzw. wirtschaftlicher Nutzen>.

Als Nutzer der Bibliothek
möchte ich den Bestand nach Büchern eines bestimmten Autors durchsuchen können,
so dass ich alle Bücher meines Lieblingsautors finde.

Eine User-Story beschreibt eine Funktionalität, die für den Kunden oder Benutzer eines Produkts oder Systems von Wert ist. Sie besteht aus der schriftlichen Beschreibung der Funktionalität, Gesprächen über die Funktionalität und Akzeptanzkriterien, die Details vermitteln und festlegen, wann eine User-Story vollständig umgesetzt ist.

BPMN-Prozessdiagramm

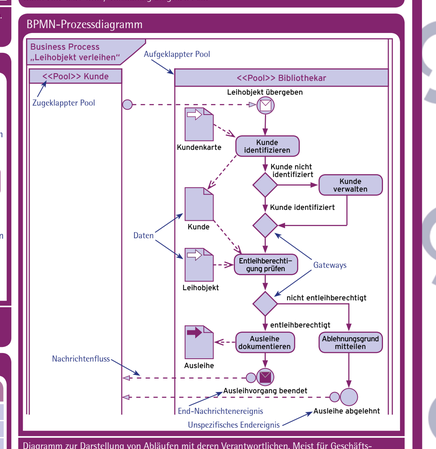


Diagramm zur Darstellung von Abläufen mit deren Verantwortlichen. Meist für Geschäftsprozesse verwendet. Weitere Diagrammarten neben dem Prozess-/Kollaborationsdiagramm: Konversationsdiagramm und Choreographiediagramm.
Der Standard ist bei der OMG (Object Management Group) verfügbar unter www.omg.org

Planquage

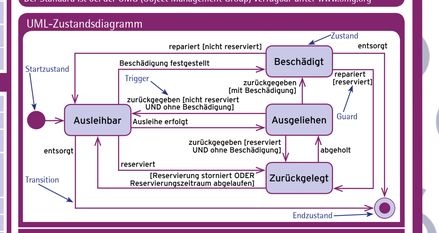
Konzept/Parameter	Erklärung
Name	Name der Anforderung
Gist	Grobe, informelle Umschreibung der Anforderung
Scale	Definition der Metrik der Anforderung
Meter	Operationalisierung der Metrik durch ein handhabbares Messverfahren
Past	Vergangener oder aktueller Ausgangszustand für die Anforderung
Must	Minimalziel für die Anforderung im Sinn von „soeben hinreichend"
Plan	Optimalziel für die Anforderung, im Sinne von „erfolgreich und zufriedenstellend"

Beispiel:

Name	Reaktion auf Tastendruck
Gist	Das System muss schnell auf Tastendruck reagieren
Scale	Durchschnittliche Zeit bis zur sichtbaren Reaktion auf den Tastendruck
Meter	Use-Case „Leihobjekt ausleihen" wird 25 mal durchgeführt. Es werden alle Reaktionszeiten des Systems mit Stoppuhr gemessen. Aus den Messergebnissen wird der Durchschnitt gebildet.
Past	1,3 s
Must	1,0 s
Plan	0,2 s

Von Tom Gilb entwickelte Möglichkeit, Qualitätsanforderungen zu quantifizieren. Beliebt in der Anwendung, da die Schablone leicht anzuwenden und das Ergebnis leicht zu verstehen ist.

UML-Zustandsdiagramm

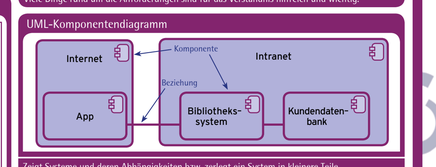


Zeigt das Verhalten eines Systems oder den Lebenszyklus / Workflow eines Objekts.

Freitext

Beispiele, Hinweise, Begründungen, Anmerkung, Annahmen, ...
Viele Dinge rund um die Anforderungen sind für das Verständnis hilfreich und wichtig.

UML-Komponentendiagramm



Zeigt Systeme und deren Abhängigkeiten bzw. zerlegt ein System in kleinere Teile.

Und-Oder-Bäume

Hierarchische Zerlegung eines Betrachtungsgegenstands, z. B. als Feature Tree oder als Zielbaum.

UML-Sequenzdiagramm

Exemplarische Darstellung (Szenario) der Interaktion mehrere Beteiligter.

Datenflussdiagramm

Zerlegt ein System aus Prozesssicht Top Down. Je Prozess (-teil) wird dargestellt, welche Daten der Prozess benötigt bzw. erzeugt. Auf oberster Ebene als Kontextdiagramm verwendbar.

Akzeptanzkriterien-Schablone

Schablone zur natürlichsprachlichen Beschreibung von Testfällen bestehend aus: Ausgangssituation, Ereignis und erwartetes Ergebnis.

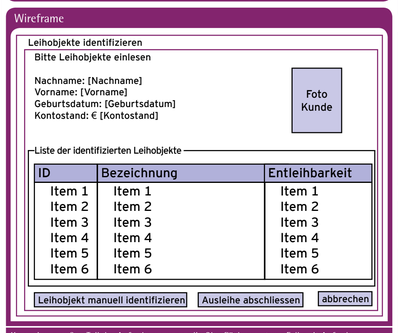
Entity-Relationship-Diagramm

Alternative zum UML-Klassendiagramm. Wie dieses wird das Entity-Relationship-Diagramm zur visuellen Definition der Begriffe eingesetzt.

Ereignisgesteuerte Prozesskette

Diagramm zur Ablaufdarstellung von Geschäftsprozessen.

Wireframe



Kann einen großen Teil der Anforderungen an die Oberfläche ersetzen. Falls mit Anforderungen gleichgesetzt (z. B. Vertragsgegenstand), muss definiert werden, welcher Teil der Darstellung verbindlich ist (z. B. RGB-Werte der Farbe der Buttons).

Tabellen

Stakeholdertabellen, Systemlisten, Dokumentlisten, ...
Oft bei der Verwaltung schwierig. Einerseits werden Tabellen von Requirements-Management-Tools nicht immer optimal unterstützt, andererseits leidet oft die Identifizierbarkeit (eine ID pro Anforderung) bei einer Tabelle, da diese oft **mehrere** Anforderungen enthält.

Prototypen

Ob Skizzen, Wireframes, Muster aus 3D-Drucker, ... Prototypen aller Art machen Anforderungen plastischer und ergänzen diese.

Konsolidierungsmatrix

	Einigung	Kompromiss	Variantenbildung	Abstimmung	Über sticht Unter
Hohe Anzahl von Stakeholdern	–	–	+	+	+
Hohe Kritikalität des Sachverhalts	+	–	–	–	–
Weiträumige Verteilung der Stakeholder	–	–	+	+	+
Hoher Zeitdruck für Konfliktauflösung	–	–	–	+	+
Eindeutigkeit des Ergebnisses wichtig	+	+	–	+	+
Geringe Sozialkompetenz der Stakeholder	–	–	+	+	+
Komplizierter Sachverhalt	–	+	–	–	–
Lange Lebensdauer der Ergebnisse	+	–	+	–	–
Geringe Motivation der Stakeholder (aktiv mitzuwirken)	–	–	+	+	+
Machtgefälle zwischen beteiligten Personen	–	–	0	+	+
Problematische Gruppendynamik	–	–	+	+	+
Viele verschiedene Meinungen	–	–	+	+	+
Geringe kommunikative Fähigkeiten der Stakeholder	–	–	+	+	+
Schlechte zeitliche Verfügbarkeit der Stakeholder	–	–	0	+	+
Konflikt betrifft Sachebene	+	+	+	+	–
Konflikt betrifft Beziehungsebene	–	–	–	–	–
Konflikt betrifft Wertebene	–	–	+	–	–
Konflikt betrifft Strukturebene	–	0	0	+	+
Konflikt betrifft Interessenebene	–	–	+	0	+
Bestimmender Stil in der Konfliktauflösung „Vermeidung"	–	–	+	+	+
Bestimmender Stil in der Konfliktauflösung „zwang"	–	–	+	–	–
Bestimmender Stil in der Konfliktauflösung „Zusammenarbeit"	+	+	–	–	–
Bestimmender Stil in der Konfliktauflösung „Entgegenkommen"	+	+	–	+	+
Bestimmender Stil in der Konfliktauflösung „Kompromissbereitschaft"	+	+	0	0	0

Konfliktlösungen

Umgang mit unterschiedlichen oder sich widersprechenden Anforderungen.

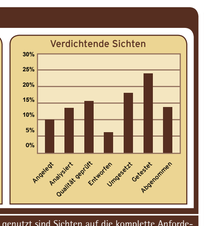
Verwalten

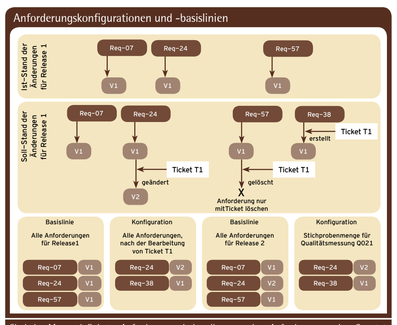
Sichtenbildung

Selektive Sichten

Objekt-ID	Zustand	Anforderungssatz	Version
LH-Bib 152	Angelegt	Das Bibliothekssystem ...	1
LH-Bib 186	Angelegt	Das Bibliothekssystem ...	2
LH-Bib 241	Angelegt	Das Bibliothekssystem ...	3
LH-Bib 639	Angelegt	Das Bibliothekssystem ...	4

Verdichtende Sichten



Im Arbeitsalltag zwingend erforderlich und oft genutzt sind Sichten auf die komplette Anforderungsbasis. Unterschieden werden selektive Sichten und verdichtende Sichten:
Selektive Sicht: beinhaltet einen Teil der verfügbaren Anforderungsinformationen (z. B. alle Anforderungen im Zustand x mit einer bestimmten Person als aktueller Bearbeiter).
Verdichtende Sicht: enthält generierte oder verdichtete Daten, die nicht unmittelbar in der Anforderungsbasis enthalten sind (z. B. eine Fortschrittsauswertung).

Anforderungskonfigurationen und -basislinien