



Requirements Engineering: Neun Prinzipien und ihre praktische Anwendung

Veröffentlicht am 17. Juni 2025

Ihre Ansprechpersonen: Oliver Lisik, Rico Rutz

Tags: Fachbeiträge, Energie und Infrastruktur, Gesundheitswesen, Handel, Transport und Logistik, Industrie, Technologie und Dienstleistung, Öffentliche Verwaltung, Business Analyse und Requirements Engineering

Gutes Requirements Engineering (RE) ist mehr als das systematische Erfassen von Anforderungen. Es schafft die Grundlage für Lösungen, die Mehrwert stiften. Um Orientierung im komplexen Anforderungsprozess zu geben, hat das IREB (International Requirements Engineering Board) neun Prinzipien formuliert, die als Leitplanken für professionelles, verantwortungsvolles RE dienen. Sie helfen, Anforderungen nicht nur korrekt, sondern auch sinnvoll, nutzbringend und anpassbar in jeglichen Situationen zu gestalten. Damit auch Sie die neun Prinzipien künftig anwenden können, stellen wir diese nachfolgend vor und zeigen, wie sie in der Praxis umgesetzt werden.

1. Wertorientierung: Jede Anforderung ist eine Investition

Beschreibung

Das erste und zentrale Prinzip des Requirements Engineerings ist die Wertorientierung. Es zielt darauf ab, jede Anforderung als Investition zu betrachten. Denn Anforderungen sind Ausdruck eines wirtschaftlichen oder nutzbringenden Ziels; ihr Nutzen muss gegenüber den Kosten hinsichtlich Ermittlung, Dokumentation, Validierung, Abstimmung, Verwaltung und Realisierung abgewogen werden. Deshalb muss immer wieder überprüft werden, ob der erwartete Nutzen einer Anforderung die entstehenden Kosten rechtfertigt. Diese Nutzenbewertung ist kein einmaliger Akt, sondern ein kontinuierlicher Entscheidungsprozess, insbesondere in agilen Projekten. Wertorientierung bedeutet daher nicht, dass jede Entscheidung genauestens kalkuliert werden muss, sondern dass Entscheidungen bewusst und nachvollziehbar getroffen werden, stets mit dem



Fokus auf maximalen Beitrag zum Geschäftsziel.

Anwendung

In der praktischen Umsetzung der Wertorientierung ist die Durchführung eines Kosten-Nutzen-Verhältnisses ein sehr gutes Hilfsmittel, um wiederkehrend die Anforderungen neu abzuschätzen. Im Laufe des Projektes können sich Anforderungen ändern, deshalb ist die sorgfältige Vorausplanung des Bearbeitungsaufwandes und des Abstraktionsgrades wichtig. Besonders bei agilen Entwicklungsprojekten hat sich die Kosten-Nutzen-Betrachtung bewährt, indem typischerweise der Return on Investment (ROI) pro Anforderung bewertet und zur Priorisierung genutzt wird. In wasserfallähnlichen Prozessen hingegen wird eine solche wertorientierte Betrachtung aufgrund der Schwierigkeiten einer frühzeitigen, präzisen Abschätzung kaum praktiziert. Empfehlenswert ist, Kosten-Nutzen-Abschätzungen zunächst grob und auf höheren Abstraktionsebenen durchzuführen, sie in späteren Iterationen regelmässig zu aktualisieren und den Aufwand dabei gering zu halten. Prüfen Sie ausserdem, ob Anforderungen tatsächlich notwendig sind und ob der Aufwand zur detaillierteren Ausarbeitung gerechtfertigt ist.

2. Stakeholder: Zusammenarbeit als Erfolgsfaktor

Beschreibung

Jede Anforderung ist Ausdruck von Erwartungen, Interessen oder Wünschen bestimmter Stakeholder – seien es Kundschaft, Systembetreiber oder Gesetzgeber. Das Stakeholder-Prinzip macht diese Tatsache explizit: Es verpflichtet Requirements Engineers dazu, frühzeitig und systematisch zu erfassen, wer an den Anforderungen beteiligt ist oder von ihnen betroffen sein wird. Denn nur wenn die richtigen Personen eingebunden werden, können die Anforderungen als Spiegelbild der relevanten Bedürfnisse gelten. Stakeholder-Fokus bedeutet aber auch aktiv zuzuhören, Widersprüche zu erkennen und Spannungsfelder proaktiv und kollektiv zu lösen.

Anwendung

In der Praxis ist wichtig, die Stakeholder sorgfältig zu evaluieren. Hierfür können unter anderem Stakeholder-Maps oder Personas verwendet werden. Personas dienen in Situationen, in denen die Anwendungsgruppen noch nicht bekannt sind oder es eine grosse Anzahl an Einzelpersonen gibt. Stakeholder übernehmen in der Praxis oft mehrere Rollen, die sich im Verlauf des Projektes verändern können. Deshalb ist die stetige Nachverfolgung der Veränderungen der Stakeholder zentral, wobei der Kommunikationsbedarf bei Bedarf angepasst werden muss und der Requirements Engineer als Vermittler und Brückenbauer zwischen den Stakeholdern agieren sollte. Denn, die frühe und regelmässige Einbindung der relevanten Personen ermöglicht ungenaue Anforderungen frühzeitig zu erkennen und damit falsche Entwicklungen und



zusätzliche Kosten zu vermeiden. Des Weiteren sind auch andere Anforderungsquellen zu beachten. Ein System muss nicht nur die Wünsche und Bedürfnisse der Stakeholder erfüllen, sondern auch Anforderungen berücksichtigen, die sich aus bestehenden oder konkurrierenden Systemen ergeben.

3. Gemeinsames Verständnis: Implizites Wissen explizit machen

Beschreibung

Eines der häufigsten Probleme in IT- oder Systemprojekten ist nicht technische Komplexität, sondern ein fehlendes gemeinsames Verständnis. Anforderungen sind oft in natürlicher Sprache formuliert – und diese ist interpretationsoffen. Das dritte Prinzip adressiert deshalb das Risiko von Mehrdeutigkeiten und Wissenssilos: Implizites Wissen soll explizit gemacht werden. Es geht darum, eine gemeinsame Sprache und Sichtweise zu entwickeln, in der alle Beteiligten, vom Fachbereich bis zum Entwicklungsteam, dasselbe unter einem Begriff, einem Prozess oder einer Funktion verstehen. So werden Missverständnisse minimiert und Nacharbeiten reduziert.

Anwendung

Zur Schaffung eines solchen gemeinsamen Verständnisses ist es entscheidend, eine ausgewogene Mischung aus explizitem und implizitem Wissen zu finden. Nicht alles lässt sich dokumentieren, auch festgehaltene Inhalte können fehlinterpretiert werden. Die Kunst im Requirements Engineering besteht daher darin, nur so viel zu explizieren wie nötig, um Verständlichkeit und Nachvollziehbarkeit zu sichern. Unterstützend wirken dabei Methoden wie Story Mapping, Storytelling oder das gemeinsame Erarbeiten von Anforderungen und Modellen. Visualisierungen, beispielsweise durch UML- oder BPMN-Diagramme, ergänzen textliche Beschreibungen und schaffen Klarheit. Gleichzeitig fördern räumliche Nähe, stabile Teams und kurze Kommunikationswege das implizite, informelle Wissen. Da Anforderungen immer im Kontext eines konkreten Systems und seiner Umgebung entstehen, ist es wichtig, regelmässig gemeinsam zu prüfen, ob das vorhandene Verständnis noch übereinstimmt, unter anderem durch Reviews, Prototypen oder Priorisierungsworkshops.

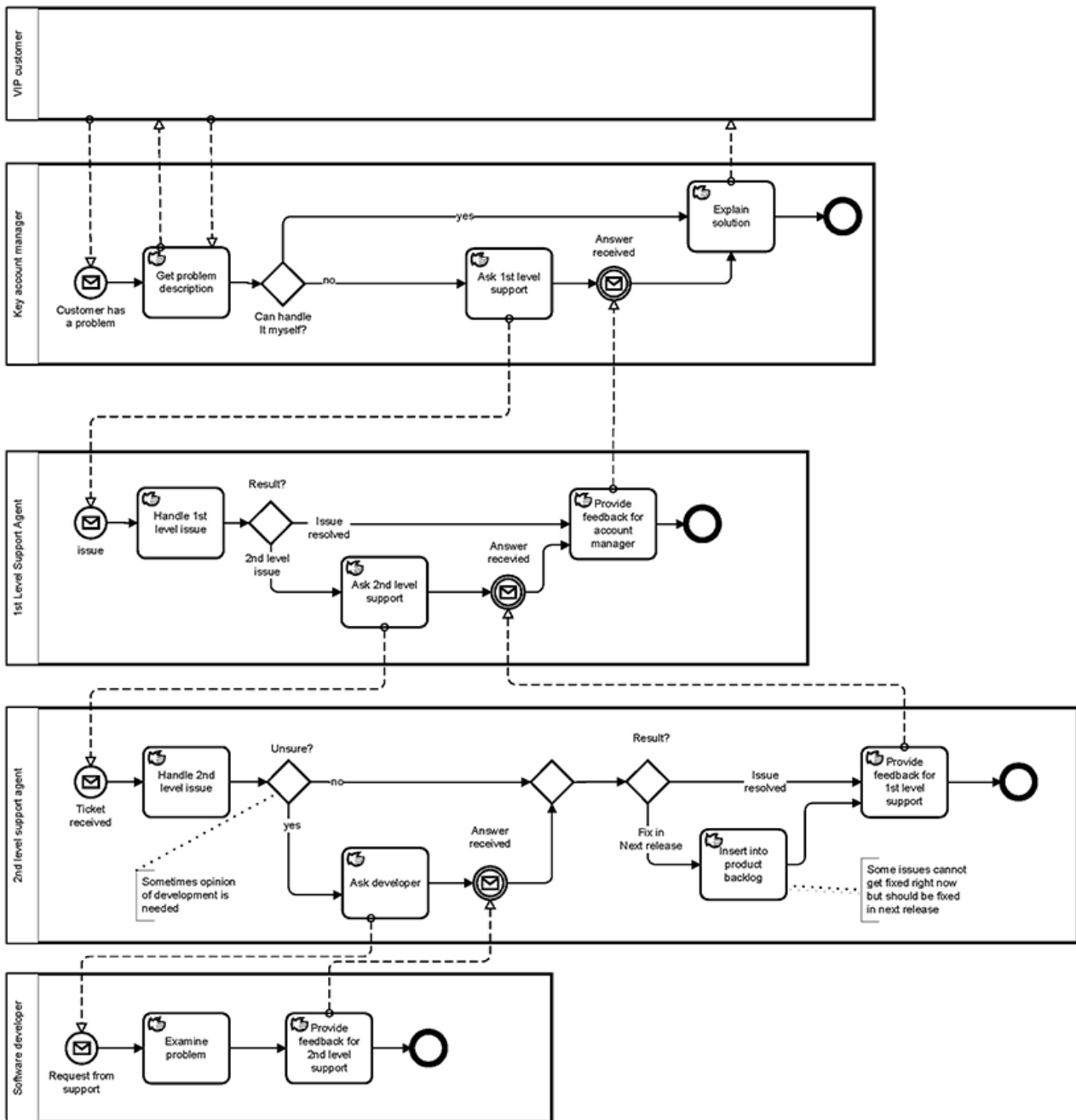


Abbildung 1: Beispielhafte BPMN-Darstellung eines Incident-Management-Prozesses mit menschlichen und systembasierten «Pools» (OMG, 2010, S. 11).

4. Kontextbezug: Wichtigkeit der Setzung von Grenzen



Beschreibung

Ein System interagiert mit seiner Umwelt. Der Kontext, in dem Anforderungen entstehen und umgesetzt werden, ist entscheidend für deren Gültigkeit und Umsetzungsmöglichkeiten. Das vierte Prinzip fordert daher, diesen Kontext explizit zu machen. Dabei sind zentrale Fragen zu klären: Welche Nachbarsysteme, Prozesse, Vorschriften oder Rahmenbedingungen wirken auf das geplante System ein? Welche Aspekte liegen ausserhalb des Projektumfangs, sind aber für das Verständnis und die Bewertung der Anforderungen relevant?

Anwendung

Ein zentrales Hilfsmittel der Kontextanalyse ist das Systemkontext-Diagramm. Es zeigt Schnittstellen zu anderen Systemen, zu Nutzer:innen sowie zur Umwelt und hilft dabei, den Systemrahmen klar zu definieren. Wichtig ist auch, bewusste Auslassungen zu dokumentieren und festzuhalten, was nicht Teil des Systems ist. IREB schlägt zusätzlich zur Systemgrenze eine separate Scope-Grenze vor. Auf diese Abgrenzung kann jedoch verzichtet werden, da sie die Komplexität erhöht. Eine klare Systemgrenze reicht aus, um festzulegen, was verändert werden darf, vorausgesetzt, alle relevanten Aspekte sind korrekt dem System oder dem Kontext zugeordnet.

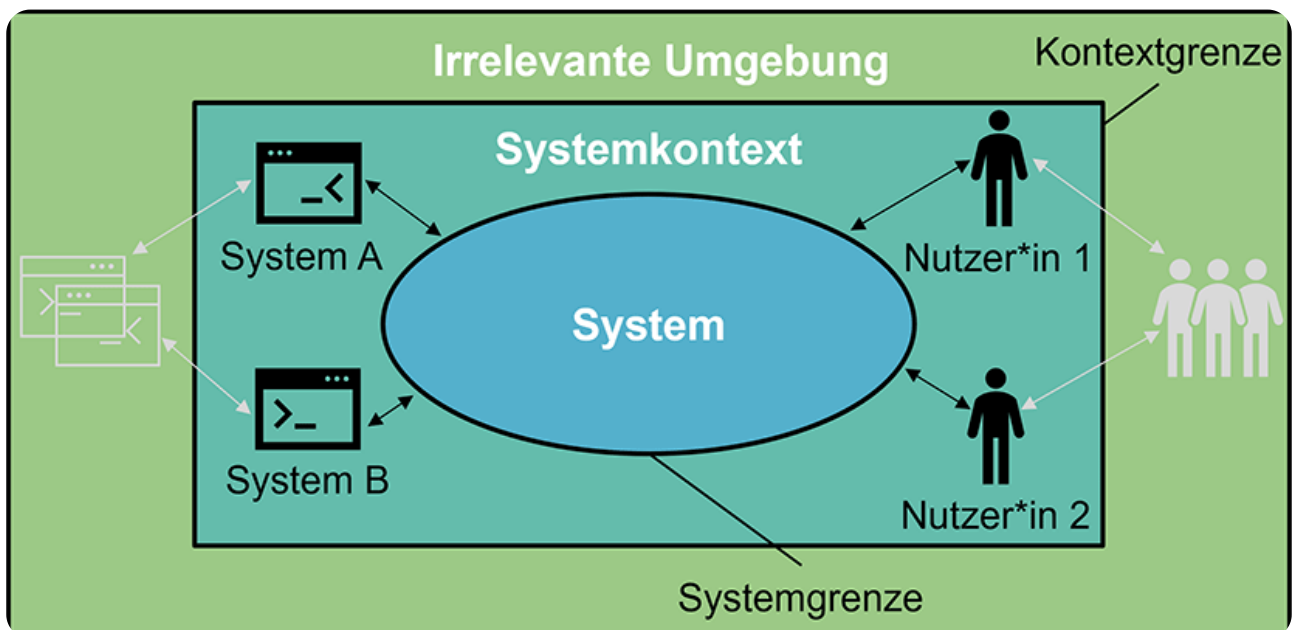


Abbildung 2: Beispielhaftes Systemkontext-Diagramm (eigene Darstellung in Anlehnung an CPRE, 2024, S. 63).



5. Verbindung von Problem, Anforderung und Lösung: Das dynamische Spiel meistern

Beschreibung

Das fünfte Prinzip beschreibt das dynamische Zusammenspiel zwischen einem zu lösenden Problem, den daraus abgeleiteten Anforderungen und den entwickelten Lösungen. In der Theorie folgt auf die Problemanalyse eine Phase der Anforderungserhebung und anschliessend die Lösungsentwicklung. Doch in der Praxis verläuft dieser Prozess selten linear. Anforderungen beeinflussen die Lösung, umgekehrt offenbaren Lösungsansätze häufig neue Einsichten in bestehende Probleme. Dieses Prinzip macht darauf aufmerksam, dass Anforderungen nie isoliert betrachtet werden dürfen. Stattdessen müssen sie kontinuierlich auf ihre Ursprünge und potenziellen Auswirkungen reflektiert werden. Besonders wichtig ist die Unterscheidung zwischen dem Was (Anforderung) und dem Wie (Lösung), um unnötige Vorfestlegungen zu vermeiden und Innovationen Raum zu geben.

Anwendung

In der Praxis bewährt sich eine Anwendung nach dem «Twin-Peaks-Modell», das die parallele, iterative Entwicklung von Anforderungen und Architektur betont. Anforderungen sollten zunächst Bedürfnisse und Problemursachen klar benennen, ohne Lösungen vorwegzunehmen. Eine bewusste Trennung zwischen Anforderungen und Randbedingungen schafft Klarheit und bewahrt Gestaltungsspielräume. In agilen Projekten empfiehlt sich die Formulierung von User Stories mit Akzeptanzkriterien und die schrittweise Entwicklung von Lösungsideen im Team. Architekturentscheidungen können neue Anforderungen erzeugen und umgekehrt. Die enge Abstimmung zwischen Requirements Engineering und Architektur ist daher entscheidend für Konsistenz und Qualität gemäss IREB.

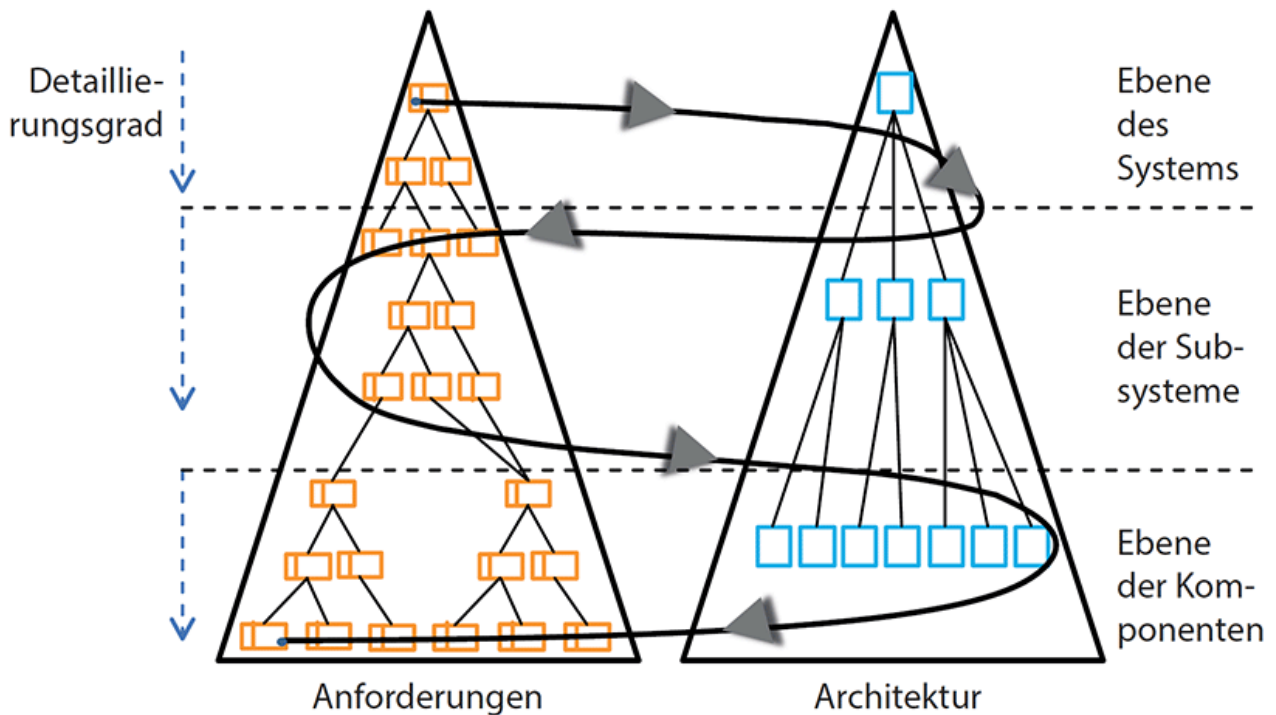


Abbildung 3: Das Twin-Peaks-Modell mit den Wechselwirkungen zwischen Architekturentwurf und RE (Pohl & Rupp, 2021, S. 35 in Anlehnung an Nuseibeh, 2001; Ward & Mellor, 1985).

6. Validierung: Regelmässige Prüfungen als Mittel zum Ziel

Beschreibung

Das sechste Prinzip erinnert daran, dass eine Anforderung erst dann wirklich «existiert», wenn sie von den Stakeholdern validiert wurde. Eine Anforderung kann gut klingen, logisch aufgebaut sein und trotzdem an den tatsächlichen Bedürfnissen der Stakeholder vorbeigehen. Deshalb fordert das Prinzip der Validierung jede Anforderung systematisch zu prüfen: Nicht nur auf Vollständigkeit oder Eindeutigkeit, sondern auch auf Zweckmässigkeit, Zustimmung und Realisierbarkeit. Validierung ist kein einmaliger Akt, sondern ein integraler Bestandteil des gesamten Requirements-Prozesses.

Anwendung

In der Praxis ist die Validierung ein zentraler Baustein des Requirements Engineerings und sollte frühzeitig und kontinuierlich erfolgen. Methoden reichen von klassischen Reviews (z. B. Inspektionen) über informelle Peer-Reviews bis hin zu empirischen Tests mit Prototypen. Wichtig ist, dass alle relevanten Stakeholder



einbezogen werden, denn die Zustimmung einzelner Personen genügt nicht. Anforderungen müssen, abgestimmt auf ihre Konsistenz, mit Zielen geprüft und regelmässig hinterfragt werden. Voraussetzung dafür ist eine zielgerichtete Kommunikation, um Informationslücken zu erkennen und zu schliessen. Kontinuierliche Validierung sichert die Qualität der Anforderungen und vermeidet teure Fehlentwicklungen.

7. Evolution: Der Weg ist nicht geradlinig

Beschreibung

Wie bereits erläutert, können sich Anforderungen unter anderem durch neue Erkenntnisse, wechselnde Marktbedingungen, technologische Neuerungen oder regulatorische Eingriffe verändern. Das Prinzip der Evolution akzeptiert diese Realität und erklärt Wandel nicht zum Problem, sondern zur Normalität. Gute Anforderungen sind nicht nur stabil, sondern auch anpassbar. Evolution bedeutet nicht Chaos, sondern die Fähigkeit, mit Veränderungen umzugehen, ohne den Überblick oder die Kontrolle zu verlieren.

Anwendung

Anforderungsänderungen sind in Projekten unvermeidbar, lassen sich aber durch geeignete Massnahmen kontrollieren. Ein professionelles Requirements Management muss daher Änderungsfähigkeit bewusst einplanen, etwa durch kurze Entwicklungszyklen, in denen Anforderungen temporär stabil bleiben. Änderungswünsche werden am Zyklusende gesammelt, priorisiert und gegebenenfalls in einem der nächsten Iterationen umgesetzt. Ergänzend braucht es ein strukturiertes Change Management, das Änderungen nachvollziehbar macht und ihre Auswirkungen analysiert. Dazu gehören die Attribuierung sowie Versionskontrolle einzelner Anforderungen und deren definierter Änderungsprozess (beispielsweise über Change Requests). Die Nachverfolgbarkeit hilft, die Auswirkungen auf andere Bedingungen frühzeitig zu erkennen, wobei Tools wie Jira in der Festhaltung unterstützen können.

8. Innovation: Kreieren statt reagieren

Beschreibung

Das achte Prinzip fordert, Requirements Engineering nicht nur als reaktive Disziplin zu verstehen, die auf bestehende Bedürfnisse reagiert, sondern als einen kreativen Prozess, der aktiv neue Möglichkeiten



eröffnet. Stakeholder äussern oftmals nur Aspekte, die sie aus ihren bisherigen Erfahrungen kennen. Wirkliche Innovation entsteht aber dort, wo über das Gewohnte hinausgedacht wird: Etwa durch das gezielte Hinterfragen bestehender Annahmen, durch kreative Methoden oder durch das Einbringen von Zukunftsperspektiven. Innovation ist dabei nicht Selbstzweck, sondern eine gezielte Strategie zur Steigerung des langfristigen Nutzens.

Anwendung

Um Innovation zu fördern, haben sich Methoden wie Design Thinking, Reverse Brainstorming oder «What-if»-Analysen bewährt. In Workshops können gezielt Perspektivwechsel erzeugt werden. Dies beispielsweise durch das Einnehmen der Sichtweise eines extremen Nutzers/einer externen Nutzerin. Auch der Einsatz von frühen Prototypen, Mock-ups oder MVPs (Minimum Viable Products) ermöglicht es, neue Ideen risikofrei zu testen und mit Feedback weiterzuentwickeln. Requirements Engineers sollten hier bewusst Freiräume schaffen, in denen unkonventionelle Anforderungen unabhängig vom operativen Projektlauf weiterverfolgt werden können.

9. Systematik und Disziplin: Der eigene Baukasten

Beschreibung

Das neunte und abschliessende Prinzip unterstreicht die Notwendigkeit eines strukturierten und methodisch fundierten Vorgehens im Requirements Engineering. Auch wenn Kreativität, Flexibilität und Innovation zentrale Bestandteile guter Anforderungen sind, wird ein tragfähiges Fundament benötigt. Dieses Fundament besteht aus klaren Prozessen, bewährten Methoden und kontinuierlicher Reflexion. Damit das Vorgehen nachvollziehbar, replizierbar und anpassbar ist, braucht es genau diese Systematik und Disziplin, unabhängig davon, wie ein Projekt aufgesetzt ist.

Anwendung

In der Praxis zeigt sich systematisches Requirements Engineering als ein lebendiger, projektspezifischer Baukasten aus bewährten Praktiken, Templates, Checklisten und Leitfäden. Dieser Baukasten sollte nicht starr, sondern anpassbar und reflektiert sein. Zudem sollte dieser Kasten auf den Einsatzbereich, die Projektbedingungen, die Teamstrukturen und die Prozessreife abgestimmt werden. Requirements Engineers analysieren dafür gezielt den Projektkontext und wählen passende Techniken aus, statt sie unreflektiert zu übernehmen. Nach Projektphasen oder Sprints sollte regelmässig hinterfragt werden, welche Methoden funktioniert haben und welche verbessert werden können – sei es in Retrospektiven, Meilenstein-Reviews



oder gezielten Lessons-Learned-Runden. Auch der Einsatz moderner Werkzeuge wie Confluence oder modellbasierter Ansätze unterstützt dabei, Qualität und Effizienz zu sichern. Wichtig ist: Systematik soll Orientierung und Raum für Kreativität bieten.

Fazit

Die neun Prinzipien des Requirements Engineerings bieten einen praxisnahen Orientierungsrahmen für die erfolgreiche Entwicklung komplexer Systeme. Sie machen deutlich: Gutes Requirements Engineering ist kein starres Regelwerk, sondern lebt von einem ausgewogenen Zusammenspiel aus Struktur und Flexibilität. Gemeinsames Verständnis, Kontextbewusstsein, kontinuierliche Validierung und Änderungsfähigkeit sind ebenso entscheidend wie systematisches Vorgehen und die Bereitschaft zur Reflexion. Wer diese Prinzipien bewusst anwendet und situativ anpasst, schafft die Grundlage für tragfähige Anforderungen und damit für erfolgreiche Projekte!

Möchten Sie mehr über dieses relevante Thema erfahren oder wissen, wie wir Sie bei einem herausfordernden Vorhaben unterstützen können? Zögern Sie nicht uns per Telefon unter 058 320 30 00, per E-Mail unter office@app.ch oder mittels des untenstehenden Formulars zu kontaktieren. Unsere Expertinnen und Experten freuen sich auf Ihre Anfrage.

Quellen

-
- CPRE. (2024). *Certified Professional for Requirements Engineering* (Version 1.2). Foundation Level Handbuch. CPRE.
-
- Nuseibeh, B. (2001). Weaving the Software Development Process Between Requirements and Architectures (S. 115-119). In *Computer* (Band 34 (3)). The Open University.
-
- OMG. (2010). *BPMN 2.0 by Example* (1. Version). Object Management Group.
-
- Pohl, K. & Rupp, C. (2021). *Basiswissen Requirements Engineering* (5. Version). dpunkt.verlag GmbH.
-
- Ward, P. & Mellor, S. (1985). *Structured Development of Real-Time Systems – Introduction and Tools* (Band 1). Prentice Hall, Upper Saddle River, 1985.
-