

Requirements Engineering in der Softwareentwicklung

Überblick

Requirements Engineering (RE) ist kein optionales Extra, sondern der strategische Kern von Softwareprojekten. Es geht darum, Anforderungen an ein System, Produkt oder Projekt strukturiert zu ermitteln, zu dokumentieren, zu validieren und zu verwalten. Ein strukturiertes RE stellt sicher, dass die Fähigkeiten und Eigenschaften einer Software den Erwartungen der Stakeholder entsprechen und unterstützt die Planung sowie Steuerung der Projektziele. Fehlen oder unklare Anforderungen gehören laut Studien zu den häufigsten Ursachen für Projektmisserfolge, daher bildet ein professionelles Requirements Engineering die Basis für erfolgreiche Softwareentwicklung.

Kernaktivitäten

Die International Requirements Engineering Board (IREB) und andere Fachverbände definieren vier zentrale Aktivitäten, die in jedem RE-Prozess zu beachten sind:

| Kernaktivität | Zweck | Methoden/Techniken |
|--------------------------------------|---|--|
| Ermittlung (Elicitation) | Anforderungen werden gesammelt, verfeinert und präzisiert, damit sie nicht auf Annahmen basieren | Interviews, Workshops, Beobachtung, Dokumentenanalysen und Stakeholderbefragungen |
| Dokumentation (Specification) | Anforderungen werden präzise, überprüfbar und konsistent beschrieben | Use Cases, User Stories, formale Spezifikationen, Lasten-/Pflichtenhefte |
| Validierung (Validation) | Prüfung auf Vollständigkeit, Realisierbarkeit, Widerspruchsfreiheit und Eindeutigkeit | Reviews, Prototypen, Stakeholder-Abstimmungen, Testfälle |
| Management | Verwaltung der Anforderungen über den gesamten Lebenszyklus; Änderungen werden verfolgt und priorisiert | Versionierung, Traceability-Matrizen, Änderungsmanagement, Priorisierungstechniken |

Arten von Anforderungen

Im Requirements Engineering unterscheidet man üblicherweise drei Gruppen:

- **Funktionale Anforderungen** beschreiben, welche Funktionen das System bereitstellen muss. Sie geben vor, was das System tun soll, um den Anwender zu unterstützen.
- **Qualitätsanforderungen (früher „nicht-funktionale Anforderungen“)** legen fest, wie gut eine Funktion umgesetzt sein soll. Dazu gehören Merkmale wie Flexibilität, Robustheit, Sicherheit oder Usability.

- **Rahmenbedingungen** (Constraints) sind organisatorische oder technologische Einschränkungen, die den Freiheitsgrad bei der Lösungsentwicklung begrenzen.

Nutzen für Projekte

Die systematische Durchführung von Requirements Engineering bringt zahlreiche Vorteile für Softwareprojekte:

- **Gesteigerte Qualität und klare Prioritäten:** Gut definierte Anforderungen schaffen Klarheit, Vollständigkeit und ermöglichen Priorisierung sowie Aufwandsschätzung. Dies verbessert die Planung und die Qualität des Ergebnisses.
- **Reduktion von Kosten und Risiken:** Fehler in Anforderungen sind laut Studien die teuersten Softwarefehler; je früher sie erkannt werden, desto kostengünstiger ist ihre Behebung. RE hilft, Fehlentwicklungen frühzeitig zu erkennen und reduziert damit Projektkosten und Supportaufwand.
- **Effizientes Änderungsmanagement:** Anforderungen ändern sich. Ein konsequentes Requirements-Management ermöglicht es Teams, Anpassungen zielgerichtet umzusetzen, ohne Budgets oder Zeitpläne zu gefährden.
- **Verbesserte Projektsteuerung:** Klare und stabile Anforderungen erleichtern Aufwandsschätzung und Budgetplanung. Sie bilden die Grundlage für belastbare Zeit- und Kostenpläne und unterstützen eine realistische Ressourcenplanung.
- **Bessere Kommunikation und Stakeholder-Einbindung:** Ein strukturiertes RE fördert den Dialog mit Stakeholdern und stellt sicher, dass deren Erwartungen verstanden werden. Stakeholdermanagement ist ein integraler Bestandteil von RE.

Technische Spezifikation und Prozessdetails

- **Vorgehensmodelle:** Requirements Engineering ist ein iterativer Prozess, der die Softwareentwicklung kontinuierlich begleitet. In agilen Modellen (z. B. Scrum) werden Anforderungen in kurzen Zyklen erfasst, priorisiert und verfeinert, während klassische Modelle (z. B. Wasserfall) stärker auf formale Spezifikationen setzen.
- **Normen und Standards:** IEEE 24765:2017 definiert die Aktivitäten „Elicitation“, „Analysis“, „Specification“ und „Validation“. Das IREB bietet international anerkannte Zertifizierungen und Leitlinien für die Professionalisierung des Requirements Engineerings.
- **Toolunterstützung:** Es gibt zahlreiche spezialisierte Werkzeuge für Requirements Engineering, z. B. IBM DOORS Next, Polarion Requirements, Enterprise Architect, Visure Requirements oder Helix ALM. Auch verbreitete Werkzeuge wie Jira, MS Word oder Excel werden in vielen Organisationen genutzt.
- **Agile Erweiterungen:** Moderne Trends integrieren Tests direkt als Anforderungen, um Akzeptanzkriterien automatisch zu überprüfen. Dabei zeigen Studien, dass Testfälle in agilen Umgebungen als Anforderungen fungieren können; dies führt zu sofort überprüfbaren Spezifikationen, stellt aber besondere Anforderungen an die Behandlung nicht-funktionaler Anforderungen.

Einsatzgebiete und Anwendungsszenarien

Requirements Engineering ist branchenunabhängig. Beispiele:

1. **Individualsoftwareentwicklung:** RE unterstützt bei der Definition von Nutzeranforderungen, Funktionsumfang und Qualitätsmerkmalen und ermöglicht eine schnelle Anpassung an sich ändernde Geschäftsprozesse.
2. **Regulierte Branchen (z. B. Medizin- oder Automobilindustrie):** Dokumentation, Validierung und Traceability sind entscheidend, um gesetzliche und normative Anforderungen zu erfüllen. Stakeholder müssen verstehen, wie kritische Systeme funktionieren; daher werden erklärbare Anforderungen und Datenqualitätskriterien immer wichtiger.
3. **KI-basierte Systeme:** Anforderungen definieren die nötige Datenqualität, legen Regeln für Erklärbarkeit fest und verlangen regelmäßige Revalidierung der Modelle.
4. **Legacy-Modernisierung und Cloud-Migration:** RE hilft dabei, bestehende Systemfunktionen und Schnittstellen zu analysieren und klare Migrationsanforderungen festzulegen, um Datenkonsistenz und Betriebssicherheit zu gewährleisten.

Best Practices für die Umsetzung

- **Stakeholder frühzeitig und kontinuierlich einbinden.** Nur wer weiß, was Nutzer erwarten, kann passende Lösungen entwickeln.
- **Anforderungen SMART formulieren:** Spezifisch, messbar, attraktiv, realistisch und terminiert.
- **Traceability sicherstellen:** Lückenlose Rückverfolgbarkeit von der Anforderung über Design und Implementierung bis zum Test erleichtert Änderungen und Nachweise.
- **Iteratives Vorgehen:** Requirements Engineering begleitet den gesamten Entwicklungsprozess und ist keine einmalige Phase.
- **Professionelle Tools nutzen:** Zur Unterstützung von Versionsverwaltung, Priorisierung und Zusammenarbeit.

Schlussfolgerung

Requirements Engineering schafft Klarheit über das „Was“ und „Warum“ eines Softwareprojekts. Es reduziert Risiken, verbessert die Produktqualität und ist die Grundlage für verlässliche Planung sowie Budgetkontrolle. Während neue Trends wie KI, Automatisierung und Agilität zusätzliche Herausforderungen mit sich bringen, bleibt das Ziel unverändert: klare, abgestimmte und dokumentierte Anforderungen sind Voraussetzung für den Projekterfolg. Ein professionelles Requirements Engineering ist daher für moderne Softwareentwicklung unverzichtbar.
