

Change-Vorhaben statt reines Technologieprojekt

Legacy-Systeme erfolgreich ablösen – 5

Handlungsempfehlungen

Veraltete Legacy-Systeme bremsen die Effizienz von Unternehmen und erschweren den digitalen Wandel. 5 Gründe, weshalb Unternehmen beim Ersatz von alten Systemen scheitern und was Sie tun müssen, damit Sie Ihre Legacy-Systeme erfolgreich ablösen.

18.11.2025



Legacy-Systeme erfolgreich ablösen – 5 Handlungsempfehlungen

#Digitale Transformation #Organisationsentwicklung
#Requirements Engineer

Requirements Engineers und Product Owner übernehmen eine Schlüsselrolle, wenn es darum geht, Legacy-Systeme erfolgreich abzulösen. Sie stehen im **Spannungsfeld zwischen strategischen Zielen, operativen Anforderungen und technischen Rahmenbedingungen**. Sie müssen nicht nur Anforderungen erfassen und priorisieren, sondern auch mit Unsicherheiten, Altlasten und widersprüchlichen Erwartungen umgehen. Dabei zeigt sich immer wieder: Viele Stolpersteine sind nicht neu, sie treten in ähnlicher Form in zahlreichen Projekten auf.

Die folgenden **fünf Ursachen** gehören zu den häufigsten Gründen, **warum IST-Ablösungen ins Stocken geraten**. Wer sie kennt und gezielt adressiert, erhöht die Erfolgchancen zum **Ablösen von Legacy-Systemen** deutlich:

1. Alt-Systeme ersetzen – Verdeckte Komplexität und unterschätzter Funktionsumfang

Software-Systeme sind oft über Jahre gewachsen. Viele Funktionen sind nicht dokumentiert, leben nur in den Köpfen der Power-User oder verstecken sich in Sonderfällen und Workarounds. In der neuen Software-Landschaft fehlen diese Funktionen und zentrale Prozesse verhalten sich ungewollt anders.

Requirements Engineers führen deshalb frühzeitig eine strukturierte Funktionsanalyse durch, z. B. mit Shadowing, Log-Analysen und Interviews. Sie sollten dabei besonders auf implizite Anforderungen achten und diese explizit dokumentieren.

Product Owners planen ausreichend Zeit für die Validierung von Anforderungen ein. Zudem nutzen sie User Story Mapping oder Impact Mapping, um auch «versteckte» Funktionen sichtbar zu machen.

2. Legacy-IT und Umsystem-Abhängigkeiten – Herausforderungen bei der Ablösung

Software-Systeme sind selten eigenständig operativ. Sie hängen von Umsystemen ab, die andere Release-Zyklen, Verantwortlichkeiten oder technische Standards haben. Das führt bei IST-Ablösungen oft zu Verzögerungen und erhöhtem Koordinationsaufwand.

Deshalb identifizieren **Requirements Engineers Schnittstellen und Abhängigkeiten** frühzeitig. Sie dokumentieren diese im Anforderungsmodell und stimmen sie mit den betroffenen System-Verantwortlichen ab.

Product Owner dagegen planen Puffer für externe Abhängigkeiten ein und sorgen für regelmässige Abstimmungen mit zuliefernden Teams. Es empfiehlt sich, ein Risiko-Backlog zu nutzen, um kritische Abhängigkeiten aktiv zu managen.

3. Scope-Creep vermeiden und fokussiert bleiben

In der Anfangsphase entstehen oft überambitionierte Ziele. Doch schon bald wird der Scope auf ein MVP eingeschränkt und im weiteren Verlauf immer weiter reduziert, was zu Frust bei allen Beteiligten führt. Gleichzeitig schleichen sich neue Anforderungen ein, die ursprünglich nicht geplant waren.

Requirements Engineers definieren klare Scope-Grenzen, verschlanken den Anforderungskatalog des Alt-Systems auf das Wesentliche und dokumentieren, was nicht Teil des Projekts ist. Zudem führen sie ein **Anforderungs-Änderungsmanagement** ein, um neue und geänderte Anforderungen strukturiert zu bewerten.

Die **Product Owners** halten den Product Backlog fokussiert. Sie kommunizieren transparent, was im MVP bzw. den Ausbaustufen enthalten ist und was nicht. Sie pflegen eine Roadmap für spätere Erweiterungen und nutzen Priorisierungsmethoden wie WSJF, um Entscheidungen objektiv herbeizuführen.

4. Übergangslösungen bei Legacy-Systemen – Kosten und Komplexität minimieren

Hybridlösungen mit parallelem Betrieb von Alt- und Neu-Systemen führen zu vervielfachtem Aufwand: **temporäre Schnittstellen, redundante Datenhaltungen mit ein- oder gegenseitiger Synchronisation, umfangreiche Datenmigrationen**. Diese Übergangsphasen dauern oft deutlich länger als geplant.

Die Übergangsszenarien planen **Requirements Engineers** bereits in der Anforderungsanalyse mit ein. Sie bewerten den Aufwand realistisch und dokumentieren technische Schulden, die später aufgelöst werden müssen.

Product Owners vermeiden unnötige Komplexität im Übergang. Sie setzen auf klare Strategien und definieren, wann und wie der Alt-Betrieb abgeschaltet wird.

5. Legacy-Systeme modernisieren – Fachbereiche aktiv einbinden

Fachbereiche bzw. Anwender werden aus Zeitgründen oft nur punktuell eingebunden. Das führt zu unvollständigen Anforderungen, Missverständnissen und geringer Akzeptanz der neuen Lösung.

Die **Requirements Engineers** binden Fachbereiche kontinuierlich ein, z. B. durch regelmässige Reviews, Workshops und Prototyping. Sie nutzen ein Stellvertretermodell,

um Feedback aus den Fachbereichen und Teams effizient zu bündeln.

Ein **Product Owner** stellt hierbei sicher, dass fachliche Ansprechpartner aktiv am Entwicklungsprozess teilnehmen. Er fördert Transparenz und Ownership, um Akzeptanz und Qualität zu steigern.

Fazit für Requirements Engineers oder Product Owners

Die **Ablösung von Alt-Systemen** ist kein reines Technologieprojekt. Es handelt sich um ein vollwertiges Change-Vorhaben, das methodische Klarheit, kommunikative Stärke und strukturiertes Vorgehen erfordert. Wer als Requirements Engineer oder Product Owner die typischen Stolpersteine kennt und gezielt adressiert, schafft die Grundlage für eine erfolgreiche Ablösung, auch wenn der Weg dorthin selten geradlinig verläuft.

Werden Sie durch ein Alt-System ausgebremst? Kontaktieren Sie uns für eine erfolgreiche Strategie zur Ablösung Ihrer Legacy-Systeme.



Der Experte

Michael Albertin

Michael Albertin, Requirements Engineer bei bbv, ist ein IREB-zertifizierter Experte mit über 15 Jahren Berufserfahrung. Dank seiner umfangreichen Erfahrungen in der Healthcare- und Medtech-Branche sowie im öffentlichen Sektor kennt

Michael sowohl klassische Projekte als auch die agile Produktentwicklung in grösseren Vorhaben und weiss, wie Anforderungen analysiert und gemanagt werden können.