

Warum geht bei Software-Einführungen so viel schief? Nachfolgend werden 20 typische Stolpersteine vorgestellt, die ein Scheitern einer Software-Einführung verursachen können.

Die Standish Group hat 2003 in ihrem Chaos-Raport festgehalten, dass bei etwas mehr als der Hälfte der IT-bezogenen Projekte der Erfolg gefährdet ist; etwa ein Sechstel der Projekte scheitert sogar ganz. Weshalb Projekte scheitern, in denen eine neue Software eingeführt werden soll, zeigen die folgenden Stolpersteine. Umgekehrt können aus diesen Stolpersteinen auch Erfolgsfaktoren abgeleitet werden, die mithelfen, Software-Einführungen erfolgreicher zu gestalten.

1. Anforderungen und Erwartungen Die Anforderungen an die neue Software spezifizieren, was diese genau machen soll (funktionale Anforderungen) und welche weiteren Eigenschaften diese haben soll (nichtfunktionale Anforderungen: Datenumfang, Sicherheitsaspekte, Ergonomie etc.). Im Dialog zwischen dem Besteller, dem Lieferanten und den zukünftigen Benutzern werden die Anforderungen häufig nicht genügend klar definiert. Man sollte von Anfang an die Anforderungen bewusst und systematisch explizit notieren, sämtliche Beteiligte an ihrer Erarbeitung teilhaben lassen (z.B. in Form von Reviews) und sie regelmässig im Verlauf der Einführung aktualisieren. Zu hohe Erwartungen und ungenügend spezifizierte Leistungen können so gleichzeitig vermieden werden.

2. anspruchgruppen Neue Software-Systeme können mehr betroffene Anspruchsgruppen (Stakeholder) haben, als man häufig meint: Benutzer (regelmässige/seltene, erfahrene/unerfahrene, interne/externe/mobile etc.), Administratoren, Besteller, Kostenstellenleiter, Dateneigner, Prozesseigner, Lieferanten usw. Im Prozess der Software-Einführung sollten alle Anspruchsgruppen unbedingt angemessen berücksichtigt werden – vor allem bei der Spezifikation der neuen Software und der eigentlichen Einführung. Neue Software-Systeme können immer auch Ängste und Hoffnungen wecken.

3. Auswahl der Software Die Auswahl der Software selber ist natürlich auch ein wichtiger Stolperstein. Die Auswahl sollte auf umfassend beschriebenen Anforderungen und einer fundierten Kosten-Nutzen-Wirtschaftlichkeits-Analyse (KNW) beruhen und darf keinesfalls ad hoc vor sich gehen oder auf persönlichen Vorlieben basieren.

4. Berater Externe Berater können helfen, fehlendes Know-how einzubringen und die Software- Einführung mit einer höheren Erfolgswahrscheinlichkeit durchzuführen. Für das Software Lifecycle Management und das Software Process Improvement gibt es mittlerweile sogar verabschiedete internationale Standards (ISO 12207 respektive ISO 15504), die als Orientierung dienen können. Externe Berater können jedoch auch selber Stolpersteine darstellen, wenn sie über zu wenig fundierte eigene Erfahrung beziehungsweise Branchen-Know-how verfügen, wenn die Abstimmungen mit ihnen schwierig ist oder wenn sie eigene Interessen verfolgen, die nicht im Einklang mit dem Projekterfolg stehen.

5. Betriebliche Aspekte Software muss nicht nur eingeführt, sondern anschliessend auch betrieben werden. «Betreiben» heisst zum Beispiel, die Verfügbarkeit sicherzustellen, Neukonfigurationen vorzunehmen (bei einem [ERP](#) z.B. ein neues Konto einrichten), Daten zu sichern (Backup), Patches aufzuspielen. Häufig vergisst man bei der Einführung neuer Software betriebliche Aspekte. In der Folge klappt der Übergang vom Projekt in den Betrieb nur harzig oder scheitert aufgrund der fehlenden Vorbereitung vollständig. Diesen Stolperstein kann man umgehen, indem man die Verantwortlichen für den späteren Betrieb bereits frühzeitig in das Projekt einbindet. Spezielle Anforderungen von Seite des Betriebs können so noch ohne grössere Folgekosten berücksichtigt werden, und der Betrieb hat genug Zeit für die Vorbereitung auf seiner Seite.

6. Dokumentation Explizite und vollständige Dokumentation aller relevanten Informationen wie Anforderungen, Schnittstellen und Änderungsanträge ist langfristig eine unerlässliche Voraussetzung für eine erfolgreiche Software-Einführung. Fehlende Dokumentation stellt eine Quelle für Fehler, Kommunikationsprobleme und langfristige Intransparenz dar. In der Regel werden Einsparungen bei der Dokumentation unweigerlich mit grösseren Aufwänden für Entwicklung, Tests und Integration bestraft. Gleichzeitig steigen die Abhängigkeit zu bestimmten Personen und die Know-how-Risiken.

7. Ergonomie Aus Zeitgründen werden Belange der Benutzerschnittstelle (Graphical User Interface, GUI) oft sehr niedrig priorisiert. Häufig fehlt auch der Wille, Benutzer rechtzeitig einzubinden. Dies führt dann zu Software, die wenig benutzerorientiert ist und entsprechend mit Akzeptanzproblemen zu kämpfen hat. Angemessene Benutzerbeteiligung (Usability Prototyping und Usability Testing) kann helfen, Aspekte der Ergonomie hinreichend zu berücksichtigen. Für Aspekte der benutzerorientierten Softwaregestaltung wurde sogar ein entsprechender internationaler Standard verabschiedet (ISO 13407).

8. Individual- und Standardsoftware Die eigene Implementierung einer Software, so genannter «Individualsoftware», hat den Vorteil, dass individuelle Anforderungen flexibel berücksichtigt werden können. Gleichzeitig kann Funktionalität bereitgestellt werden, die noch nicht auf dem Markt angeboten wird und so ein Window-of-Opportunity im Markt genutzt werden. Eigene Software-Entwicklung birgt jedoch signifikante spezifische Risiken:

Verschiebung im Terminplan durch Überschätzung der eigenen Fähigkeiten, hohe Fehlerraten, Verwendung einer zu komplexen oder noch unreifen Technologie. Aber auch wenn die Fragestellung «Make-or-Buy?» mit dem Kauf einer Standardsoftware beantwortet wird, ergeben sich eine Reihe von Stolpersteinen: signifikanter Aufwand für Parametrisierung, knappes und teures Know-how, aufgezwungene zukünftige Releasewechsel usw.

9. Integration und Testing Mit dem Kauf und der eventuellen Parametrisierung einer Software ist es noch nicht getan. Die Software muss noch mit den bestehenden Systemen in Einklang gebracht werden. Eventuell treten erst in der endgültigen Produktionsumgebung mögliche Inkompatibilitäten hervor.

10. Kommunikation Fehlende Kommunikation ist ein häufiger Stolperstein in Projekten – nicht nur bei der Einführung neuer Software. Kommunikation ist dabei viel mehr als reine Übermittlung von relevanten Informationen. Einerseits ist nicht nur das «Was» sondern auch das «Wie» wichtig. Andererseits muss die Kommunikation alle relevanten Anspruchsgruppen umfassen und auf Gegenseitigkeit beruhen.

11. Kosten Die totalen Kosten sind bei Software-Einführungen am Ende häufig grösser als geplant: Dies kann von Einnahmeausfällen durch verzögerte Inbetriebnahme über versteckte Lizenzkosten (z.B. über intransparente Verrechnungsmodelle) bis hin zu «vergessenen» Betriebskosten reichen. Damit Kosten im Projekt keine Stolpersteine werden, muss man vor der Freigabe eine ganzheitliche Kostenbetrachtung durchführen und diese auch den Entscheidungsträgern transparent kommunizieren. Nur so kann das spätere Commitment sichergestellt werden.

12. Lieferanten Bei der Auswahl der Lieferanten und Implementierungspartner ist nicht nur auf fachliche Exzellenz, sondern auch auf entsprechendes Branchen-Know-how zu achten. Die Kenntnisse der spezifischen Prozesse und Kulturen einer Branche sind häufig zentral, wenn es um den Erfolg einer neuen Softwarelösung geht.

13. Organisatorische Auswirkungen Neue Softwarelösungen sind immer auch in einen organisatorischen Kontext eingebunden: Einerseits auf Seite Informatik, andererseits auf fachlicher Seite. Neue Software bedeutet neue Prozesse. Da sich Menschen jedoch häufig nur langsam oder gar nicht auf neue Abläufe umstellen wollen, kann diese organisatorische Seite das Scheitern einer technisch perfekten Lösung bewirken. Bereits in der Planungsphase oder bei der Konzeption neuer Software muss man sich Gedanken über die organisatorischen Auswirkungen machen. Beeinflusst die neue Software Kernprozesse der Unternehmung, so muss die Ausgestaltung der späteren Abläufe mit der Anforderungsspezifikation der neuen Software Hand in Hand gehen.

14. Personalfluktuationen Das zentrale Element in Projekten ist und bleibt der Mensch. Im Laufe der Einführung einer komplexen Software erarbeiten die

Beteiligten eine beträchtliche Menge an Knowhow und an Erfahrungen. Kommt es im Verlaufe des Projektes zu personellen Wechseln (z.B. des Projektleiters oder anderer Schlüsselpersonen), so kann dies den Erfolg der gesamten Einführung bedrohen oder signifikante Folgekosten verursachen (z.B., wenn der alte Projektleiter seine Dienste neu als externer anbietet).

15. Planung Wer würde ein Haus bauen, ohne vorher einen detaillierten Bauplan zu erstellen? Vermutlich niemand. Bei der Software-Einführung wird jedoch häufig der erste Schritt gemacht, ohne dass die weiteren Schritte bis ins Detail geplant sind.

16. Politik Die Einführung einer neuen Software verschiebt häufig auch die Einflussbereiche von Beteiligten. Dies kann zu beträchtlichen «politischen» Problemen führen, die von verdeckter Sabotage bis hin zu offener Ablehnung reichen. Politischen Problemen kann man mit der Berücksichtigung der Interessen aller Anspruchsgruppen begegnen. Natürlich ist auch das Commitment «von oben» wichtig, wenn mittels neuer Software-Systeme grössere Veränderungen bewirkt werden sollen.

17. Projektorganisation/Projektleiter Die Projektorganisation ist ein wichtiger Erfolgsfaktor: Alle Beteiligten müssen angemessen vertreten werden; wenige Mitarbeitende mit grösseren Zeitpensen sind einer grösseren Anzahl Mitarbeitenden mit kleinen Pensen vorzuziehen. Die Besetzung der Rollen im Projekt muss sowohl fachlich wie auch menschlich passen. Der Projektleiter ist eine Schlüsselperson bei der Einführung neuer Software. Je grösser und komplexer das Vorhaben ist, desto besser ausgebildet und erfahrener sollte dieser sein. Sollte diese wichtige Rolle extern vergeben werden, so ist unbedingt auf angemessenen Know-how-Transfer zu achten.

18. Rechtliches Die gewissenhafte und vollständige Formulierung rechtlicher Aspekte ist ein weiterer Erfolgsfaktor beziehungsweise Stolperstein. Dies umfasst vor allem den Kaufvertrag für die Software, die eine ganze Reihe kritischer Punkte klären sollte: Vollständigkeit der Lösung, Einhaltung von zeitlichen und inhaltlichen Vorgaben, Lizenzfragen usw. Daneben ist auch die Kenntnis und adäquate Berücksichtigung von gegebenen Rahmenbedingungen und Vorgaben zentral: Dies kann Aspekte des Datenschutzes umfassen, die Pflicht zur Archivierung über einen bestimmten Zeitraum oder die allgemeine Nachvollziehbarkeit von Geschäften. Wird die neue Softwarelösung in einem internationalen Kontext eingesetzt, sind hier die Stolpersteine besonders zahlreich.

19. Sicherheitsaspekte Aspekte der Informationssicherheit umfassen Verfügbarkeit, Integrität (Unversehrtheit) und Vertraulichkeit. Einerseits müssen diese Aspekte für die einzuführende Software berücksichtigt werden, andererseits darf die neue Software keine negativen Einflüsse auf die Informationssicherheit haben. Werden die Anforderungen an die Informationssicherheit in der Spezifikationsphase zu wenig berücksichtigt, so

kann es passieren, dass bei der Einführung die neue Software gar nicht oder erst mit Verzögerungen freigegeben werden kann.

20. Vorgehen Bei der Planung des Vorgehens ist darauf zu achten, sich nicht «zu viel aufs Mal» aufzubürden. Die vollständige Einführung einer komplexen Software mit allen Funktionalitäten klappt selten in einem Schritt. Stattdessen ist ein iteratives Vorgehen zu empfehlen, das erst einmal mit der beschränkten Funktionalität einer Pilotapplikation beginnt, Gelegenheit zum Sammeln von Erfahrungen bietet und gleichzeitig Investition und Risiken in Grenzen hält