

# Verbands-Management (VM)

Fachzeitschrift für Verbands- und Nonprofit-Management



**VM 2/18**

Thomas Zurkinden und Nadja Blanchard

## **Fallstricke und Erfolgsfaktoren im Softwareeinsatz**

Verbands-Management, 44. Jahrgang, Ausgabe 2 (2018), S. 54-60.

Herausgeber: Verbandsmanagement Institut (VMI)  
Universität Freiburg/CH ([www.vmi.ch](http://www.vmi.ch))  
Redaktion: Luisa Wagenhöfer  
Layout: Luisa Wagenhöfer / Paulusdruckerei, Freiburg/CH  
Fotomaterial: Thema «Digitalisierung»: [istockphoto.com](http://istockphoto.com)  
ISBN: 3-909437-52-4  
ISSN: 1424-9189  
Kontakt: [info@vmi.ch](mailto:info@vmi.ch)

Die Zeitschrift VM erscheint dreimal jährlich in den Monaten April, August und November.

Abdruck und Vervielfältigung von Artikeln und Bildern, auch auszugsweise oder in Abschnitten, nur mit Genehmigung des Herausgebers.



Praxisbeitrag

# Fallstricke und Erfolgsfaktoren im Softwareeinsatz

Thomas Zurkinden und Nadja Blanchard

**In unserer Praxis begegnen wir (Beraterinnen und Berater der B'VM) beim Kunden auch häufig Fragen zu IT-Projekten und zur Digitalisierung. Obwohl, oder vielleicht gerade weil wir kein «IT-Dienstleister» sind, werden die damit verbundenen Problemstellungen oft und gerne mit uns diskutiert. Dies kann von Tipps zu bestehenden Standardlösungen über «Dos and Don'ts» aus Kundenprojekten bis hin zu Analyse und Entwicklung von Lastenheften für Software gehen. Die Ausführungen in diesem Beitrag sind eine Auswahl von Empfehlungen für die Einführung von Software, die wir aufgrund unserer Erfahrungen gemacht haben. Dabeisprechen wir beispielsweise die Versuchung, organisatorische Defizite durch den Einsatz von Applikationen zu beheben an. Auch die ungenügende Analyse des eigenen Bedarfs kann zum Problem werden, wie auch unklare Anforderungen an die Kernfunktionen oder ungenügende Evaluation von Anbietern. Zudem ist es wichtig, die künftigen Betriebskosten zu berechnen und Changemanagement-Aspekte nicht zu vernachlässigen.**

Der Einsatz von geeigneter Software hat sich auch bei NPO von der reinen Unterstützung der Supportprozesse zur eigentlichen Leistungserbringung entwickelt. Noch in den frühen 2000er Jahren wurden die Systemumgebungen meist dazu eingesetzt, Elemente wie etwa die Mitgliederverwaltung und die Finanzabläufe sicherzustellen und technisch zu unterstützen.

Seither nimmt die Bedeutung des Internets und vor allem der Web 2.0-Technologien überproportional zu. Und heute gibt es verschiedene NPO, die ihre Kernleistungen auch, oder ausschliesslich, über Software bereitstellen. Beispiele hierzu sind: Beratungsleistungen (z. B. Rechts- und Sozialberatung), Informationsleistungen (z. B. Asthmainformation, Pollen-News) oder Dokumentationsleistungen (z. B. Wander- und Tourenplaner). Gleichzeitig wird dem Empfänger dieser Leistungen dadurch auch bewusst, dass er im Netz

möglicherweise alternative Leistungserbringer findet, die genauso gut oder besser auf seine Bedürfnisse reagieren können (z. B. Rechtsberatungsplattformen von Dritten).

Weiter haben vor allem grössere Organisationen integrierte Systeme eingeführt oder solche mit Schnittstellen untereinander. Aufgaben wie Einsatzplanung, Leistungserfassung, das Führen von Klientendossiers, die Abrechnung, die Verwaltung von Projekten und Personal, das Reporting usw. sind in der Software abgebildet. So wird etwa Benchmarking innerhalb der Organisation und ihrer verschiedenen Einheiten möglich.

Die Möglichkeiten der neuen Technologien sind also begeisternd und gleichzeitig beängstigend. Wir stellen fest, dass die Einführung von Software unsere Kunden sehr stark beschäftigt und oft auch einen finanziellen Risikofaktor darstellt. Dies ist nachvollziehbar; schliesslich handelt es sich bei komplexeren Systemen um das operative Rückgrat der Organisationen, was zum Teil auch die Emotionalität in diesen Projekten erklärt.

Die Komplexität bei der Einführung neuer Software erhöht sich weiter, wenn man nicht auf Standardsysteme setzen kann, die man für die eigenen Bedürfnisse konfigurieren kann. Werden für die anfallenden Aufgaben Eigenentwicklungen notwendig, steigt der Anspruch an das Projektmanagement nochmals. Dies beleuchtet auch der folgende Abschnitt.

## Der «Chaos Report» der Standish Group

Die Standish Group<sup>1</sup> veröffentlicht jährlich den sogenannten «Chaos Report». Dieser hat das Ziel, Erfolgs- und Misserfolgskriterien von IT-Projekten zu identifizieren. Es muss angemerkt werden, dass die Zahlen der Standish Group von Teilen der IT-Community angezweifelt<sup>2</sup> werden, andere erachten diese als durchaus nachvollziehbar.

Der «Chaos Report 2013»<sup>3</sup> untersuchte IT-Projekte in der Periode 2004 bis 2012 und kam zum Schluss, dass die evaluierten Projekte eine Zeitüberschreitung zwischen 71 und 84 % sowie eine Kostenüberschreitung zwischen 46 und 59 % aufwiesen. Die Rate der schlussendlich implementierten Funktionen betrug zwischen 64 und 74 %. Weiter wird der Schluss gezogen, dass «kleine Projekte» (mit einem Gesamtbudget von unter 1 Million USD) zu 76 % erfolgreich abgeschlossen wurden und «grosse Projekte» (mit einem Gesamtbudget von über 10 Millionen USD) nur zu einer Quote von 10 %.

### Erfolgsfaktoren für IT-Projekte

Da grosse Projekte (gemessen an deren Kosten) grundsätzlich eine höhere Komplexität aufweisen und mit den herkömmlichen Projektmanagement-Methoden nur schwierig zu bewältigen sind, untersuchte der «Chaos Report 2015» unter anderem auch, ob es einen Zusammenhang zwischen Projekterfolg, Projektgrösse und Managementmethode gibt (vgl. Tabelle 1). Hier zeigte sich bei über 10 000 untersuchten Projekten, dass diese umso erfolgreicher sind, je kleiner der Projektumfang ist. Zudem scheint agiles Projektmanagement zu mehr Erfolg zu führen, als das traditionelle.

Was die Erfolgs- und Misserfolgskriterien der Projekte betrifft, kommt die Standish Group über die vergangenen Jahre zur folgenden Erkenntnis:<sup>4</sup>

Die festgestellten Haupterfolgsfaktoren sind:

- Einbindung der Endbenutzer
- Unterstützung durch das obere Management
- Klar definierte Anforderungen

Die Hauptpunkte, die zum Scheitern der Projekte führen, sind:

- fehlende Zuarbeit (ev. Mitarbeit) durch Benutzer
- unvollständige oder unklar definierte Anforderungen
- häufige Anforderungsänderungen

### Agiles Projektmanagement

Bei der Durchsicht dieser Resultate wird deutlich, warum bei Software «agiles Projektmanagement»<sup>6</sup> sinnvoll ist. Es verfolgt die nachstehenden Prinzipien:

- Individuen und Interaktionen sind wichtiger als technische Prozesse und Werkzeuge.
- Funktionierende Software ist wichtiger als umfassende Dokumentation.
- Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlung.
- Reagieren auf Veränderung ist wichtiger als Befolgen eines Plans.

Basis der Arbeit ist also eine Produktvision, die jedoch Raum für Abweichungen zulässt. Dies wird durch eine «modulare Entwicklung» ermöglicht, bei der es sich um ein iteratives, inkrementelles Vorgehen in Etappen handelt:

- Der Kunde erhält am Ende jeder Etappe ein voll funktionsfähiges Zwischenprodukt.
- Dieses wird intensiv getestet; auf dieser Basis erfolgt ein Feedback an die Projektleitung.
- Dann erfolgt die Weiterarbeit am Produkt auf Basis des Feedbacks.

In der Praxis kann dies bedeuten, dass für einen Verband zuerst ein anschlussfähiges und voll funktionsfähiges Modul zur Personenverwaltung entwickelt wird, danach ein Veranstaltungsmanagementmodul, danach ein Rechnungsmodul usw. All diese Module können miteinander kommunizieren und werden nach und nach eingeführt.

Tabelle 1: Projekterfolg in Abhängigkeit von Projektgrösse und gewählter Projektmethode<sup>5</sup>

Size	Method	Successful	Challenged	Failed
All Size Projects	Agile	39 %	52 %	9 %
	Waterfall	11 %	60 %	29 %
Large Size Projects	Agile	18 %	59 %	23 %
	Waterfall	3 %	55 %	42 %
Medium Size Projects	Agile	27 %	62 %	11 %
	Waterfall	7 %	68 %	25 %
Small Size Projects	Agile	58 %	38 %	4 %
	Waterfall	44 %	45 %	11 %

### Beobachtungen aus der Beratungspraxis

Die nachfolgenden Ausführungen sind eine Auswahl von Empfehlungen für die Einführung von Software, aufgrund unserer Erfahrungen. Ein Anspruch auf Vollständigkeit wird nicht erhoben.

#### Die Versuchung, organisatorische Defizite durch den Einsatz von Software zu beheben

Es ist unbestritten, dass geeignete Software Prozesse in Organisationen beschleunigen, die Produktivität erhöhen und die Kooperation verbessern kann. Hingegen macht «gute» Software eine «schlechte» Aufbau- und Ablauforganisation nicht besser (höchstens schneller). Dennoch besteht die Versuchung, solche Probleme durch Software zu lösen. Hierzu ein paar Beispiele aus der Praxis:

**Projektmanagement:** Die Geschäftsstelle einer relativ rasch wachsenden Organisation hat den Bedarf, eine bessere Übersicht zu den laufenden Projekten im Verband zu erhalten und diskutiert daher die Einführung einer Projektmanagement-Software. Im Austausch stellen wir fest, dass Mitarbeitende einer Abteilung teils einzig in Rücksprache mit einem ehrenamtlichen Kommissionsmitglied Projekte von strategischer Bedeutung starten, hierzu mit Dritten Kooperationsverträge unterzeichnen und der (eigentlich hierfür zuständige) Vorstand erst Monate später durch Zufall davon erfährt.

Dies ist kein IT-Problem, sondern das Vorherrschen von «Abteilungsdenken», fehlendes Bewusstsein für Governance-Richtlinien und allenfalls Unbedachtheit. Information, Schulung und Bewusstseinsbildung stünden hier als Massnahmen im Vordergrund und nicht die Einführung einer Projektmanagement-Software.

**Kostenrechnung:** Eine Organisation mit verschiedenen Leistungsvertragsgebern arbeitet nach dem Prinzip des Globalbudgets und grenzt die verschiedenen geforderten Leistungen und Einnahmen nicht genügend gut gegeneinander ab. Die Organisation führt keine vollständige Zeit- und Leistungserfassung und kann daher die Vollkosten nicht benennen. Auf dieser Basis kann den Geldgebern auch kein ausreichendes (oder aussagekräftiges) Reporting geliefert werden.

Dieser Mangel kann mit einer Software wohl in Teilen behoben werden. Sinnvolle Zahlen können jedoch auch hier nur geliefert werden, wenn die Mitar-

beitenden ihrer Verantwortung gegenüber den Vertragspartnern gerecht werden wollen und die Regeln zur Erfassung geklärt sind. Primär sind hier Fragen zu Führung und Transparenz, zur Good Governance, zum Bewusstsein um die Rechenschaftspflicht gegenüber Geldgebern und zur Arbeitsphilosophie zu klären.

**Wissensmanagement:** Eine Organisation stellte fest, dass sie über Informationen zu ihren wichtigsten Leistungsempfängern und Projekten verfügt, die den verschiedenen Abteilungen der NPO in einer zentralen Datenbank zugänglich sein sollten. Damit hätten die Mitarbeitenden, welche Beratungen durchführen, jene, die politische Prozesse begleiten und Lobbying betreiben und auch jene, welche die Kommunikationsmittel (Mitgliederzeitschrift, Website, Newsletter, Social Media) verantworten, Zugang zu bereichsübergreifenden Hintergrundinformationen für ihre Alltagsarbeiten.

Die Realität zeigte jedoch, dass die Einführung der geeigneten Softwarefunktionen nicht das gewünschte Resultat brachte. Zeitdruck im Alltag, fehlende gegenseitige Bedürfnisklärung sowie das klassische «Silodenken» verhinderten den gewünschten Nutzen. Dies ist im Übrigen eine weitere Erkenntnis aus den Wissensmanagementprojekten, die in den beginnenden 2000er Jahren gestartet wurden: der Austausch von Wissen hängt in erster Linie vom Verhalten der Menschen ab und nicht vom Vorhandensein einer Software. Technik und Organisation spielen bei der Einführung solcher Systeme tatsächlich eine nachgelagerte Rolle. Oder anders ausgedrückt: Eine Organisation, die technisch über ein internes «Wiki» verfügt, betreibt noch kein Wissensmanagement.

#### Ungenügende Analysen des eigenen Bedarfs

Um den spezifischen Bedarf an Software zu definieren, muss die Organisation ihre Bedürfnisse kennen. Es ist zielführend, den Bedarf des Hauptamts, des Ehrenamts und der Leistungsempfänger zu erfragen. Dies kann in Workshops mit diesen Anspruchsgruppen erfolgen oder auch durch Befragungen vor Ort. Dabei ist es von grossem Vorteil, wenn die Organisation ihre relevanten Prozesse kennt und skizziert hat. Auf dieser Basis können Fragen nach Durchlaufhäufigkeit, Kosten, Medienbrüchen, Fehlerquellen, Kontakten mit dem Leistungsempfänger etc. gestellt und geklärt werden.

Wird dieser Schritt vernachlässigt und werden wesentliche Teile der aktuellen Softwareumgebung ersetzt, werden möglicherweise teure «Nachbesserungen» notwendig, weil wesentliche Funktionen fehlen und diese nicht budgetiert und «out of scope» waren.

Was heisst das nun konkret? Für eine Vergabestiftung, die pro Jahr über 150 Projekt-Finanzierungsanträge mit einem Gesamtbudget von 30 Mio. CHF beurteilt, ist es wert- und sinnvoll, diesen Prozess von der Projekteingabe, über die Evaluation bis hin zum Entscheid digital abzubilden und zu unterstützen. So werden Antragsteller über ein Formular angehalten, die obligatorischen Angaben zu machen und Begleitunterlagen über das Webinterface der organisationseigenen Datenbank digital einzureichen. Im Backoffice können diese Anträge den verschiedenen Beurteilungsgremien zugeteilt werden, diese haben über ein Login direkten Zugriff auf die Unterlagen und können auch gleich die Begründung für die Beschlüsse erfassen (der Versand von Unterlagen entfällt). Auf dieser Basis kann am Schluss auch der Antragsteller informiert werden. Der gesamte Ablauf ist dokumentiert. Ein Reporting zu den Fördermitteln und den Empfängern für den Stiftungsrat kann in einem digitalen Cockpit erfolgen.

Die genau gleichen Funktionen könnten auch als Anforderung für eine Organisation definiert werden, welche pro Jahr 3 Fördergesuche im Wert von insgesamt CHF 70 000 erhält. Hier stellt sich jedoch die Frage, ob sich die Investition in die Digitalisierung des Prozesses lohnt. In dem Fall ist es wahrscheinlich einfacher und bedeutend kostengünstiger, ein einfaches Webformular für die Gesuchseingabe zu erstellen und die Angaben im Backoffice in der Adressverwaltung, einer Tabellenkalkulation und der normalen Dateiab-

lage zu verwalten. Diese Medienbrüche stellen zwar potentielle Fehlerquellen dar, deren Kontrolle ist aber aufgrund der geringen Anzahl an Projekten kaum ein Problem.

Es kommt auch vor, dass in dieser Phase der Kreis der Anspruchsgruppen zu eng gefasst wird, etwa wenn eine Abteilung davon ausgeht, dass die Einführung einer neuen Funktion der Software nur die eigenen Mitarbeitenden betrifft. Dies geschah bei einem Kunden, bei welchem die Kommunikationsabteilung ihr CRM um «Medienschaffende» erweiterte. Bei der kurzen Erwähnung an einer Sitzung wurde klar, dass weitere Abteilungen einen ähnlichen Bedarf gehabt hätten.

### Unklare Anforderungen an die Kernfunktionen

Führt man die Bedarfsanalyse gewissenhaft durch, steht man an deren Ende in aller Regel vor einer Fülle von Anforderungen, die man unbedingt kategorisieren muss. Die Gliederung in die Kriterien «Muss», «Soll» und «Kann»<sup>7</sup> bietet sich an (vgl. Tabelle 2).

Es liegt auf der Hand, dass eine Softwareumgebung kostenintensiver wird, je mehr «Muss-Kriterien» sie erfüllt. Gleichzeitig sinkt die Wahrscheinlichkeit, eine bestehende und bewährte Standardlösung am Markt zu finden. Damit kann auch der Vorteil verloren gehen, eine stets «release- und updatefähige» Lösung zu haben.

Uns sind mehrere Kundenbeispiele bekannt, bei welchen dies nicht mehr der Fall ist. Damit wird auch der Unterhalt der Software im Betrieb immer teurer (weil die Updates für einen einzelnen Kunden erstellt werden müssen) und gleichzeitig ist man an Hersteller und Lieferanten gebunden. Gelegentlich entsteht

Tabelle 2: Muss-, Soll- und Kann-Kriterien

Kriterium	Beschreibung	Beispiel
Muss	Die Funktion ist ausnahmslos und zwingend notwendig, ansonsten ist das System nutzlos.	Pro Mitglied müssen mehrere Adressen (bspw. Privatadresse, Rechnungsadresse, Dienstadresse etc.) verwaltet werden können.
Soll	Die Funktion ist äusserst zweckdienlich, ansonsten ist das System weniger effizient/effektiv	Die Mitglieder sollen ihre Adressen über ein gesichertes Webportal selbstständig anpassen können.
Kann	Die Funktion ist wünschenswert, das System wäre mit ihr «attraktiver»	Es kann ein Gültigkeitsdatum für Adressen erfasst werden, damit angekündigte Adressänderungen (bspw. bei Umzug) im Voraus erfasst werden können und sich am Stichtag aktivieren.

dadurch eine Abhängigkeit zu einem einzelnen Mitarbeitenden dieser Dienstleister, da die Systeme zu wenig dokumentiert sind oder deren Handhabung zu komplex ist.

Die Definition der entsprechenden Funktionen und deren Kategorisierung sind daher zentrale und bedeutende Aufgaben. Diese umfassen allenfalls auch interne Aushandlungsprozesse und ungeliebte Entscheide. Gleichzeitig bietet der Entscheid für modulare Systeme auch die Möglichkeit, «Soll»- und «Kann»-Funktionen in Zukunft nachzurüsten. Sind die gewünschten Funktionen geklärt und bewertet, ist es sinnvoll, ein sogenanntes Lastenheft zu erstellen. Dieses könnte beispielsweise folgende Elemente enthalten:

- Beschreibung der vorherrschenden IST-Situation: Vorstellung der eigenen Organisation und der Leistungsfelder (dies setzt die später beschriebenen Anforderungen in einen Kontext und macht sie besser verständlich), Beschreibung der aktuell eingesetzten Softwarelösungen und deren Defizite.
- Beschreibung der gewünschten SOLL-Situation: Was soll mit der neuen Softwareumgebung erreicht werden... und was nicht (Nicht-Ziele)?
- Konkrete Anforderungen: Beschreibung von typischen Anwendungsfällen (z. B. Mitgliederrechnungen aussenden, Veranstaltungen durchführen, ...), konkrete Funktionsbeschreibungen (z. B. Adressverwaltung, Mitgliederportal, Reportingsystem ...), nicht-Funktionale Beschreibungen (z. B. webbasiert, mandantenfähig, mehrsprachig ...)
- Mengengerüst(e): Anzahl Nutzer, Datensätze, Speicherbedarf ...

### Ungenügende Evaluation von Anbietern

Es ist ein komplexes Unterfangen, die möglichen Anbieter von geeigneter Software zu finden. Hilfe bieten Übersichten, wie sie von Fachzeitschriften<sup>8</sup> bereitgestellt werden, der Austausch mit befreundeten Organisationen oder auch eigene Webrecherchen. Zudem ist es von Bedeutung, die Kriterien zur Beurteilung der Offerten vorab festzulegen. Von den Anbietern sollte man daher verlangen, sich an den vorgegebenen Offertaufbau zu halten, um die Vergleichbarkeit zu verbessern. So ist es möglich, die Angebote nach Erfüllungsgrad der «Muss»-, «Soll»- und «Kann»-Kriterien zu gewichten und beispielsweise das Verhältnis von Kosten und Gesamterfüllungsgrad zu berechnen.

Wir stellen gelegentlich fest, dass Kunden in die «Falle des günstigsten Anbieters» tappen. Ein Beispiel: Sie schreiben elf Anbieter an, davon stellen sechs eine Offerte. Dies kann vorab schon aufzeigen, dass die Organisation möglicherweise komplexe Anforderungen stellt, die gar nicht alle Anbieter auf dem Markt erfüllen können oder wollen. Wenn von den sechs offerierenden Anbietern fünf davon in etwa in der gleichen Höhe ein Angebot stellen, der letzte jedoch nur halb so teuer ist, sollte nicht unbedingt sofort zugegriffen werden. Wenn man davon ausgehen darf, dass sich die fünf anderen untereinander nicht abgesprochen haben, könnte das «günstigste» Angebot auch das schlechteste sein. In diesem Fall ist zu prüfen, ob die Anforderungen vom günstigsten Offerte-Steller verstanden und richtig kalkuliert wurden. Ansonsten könnten teure «Nachbesserungen» die Folge sein.

Bei einem unserer Kunden war genau dies der Fall: der Auftrag war die Erstellung einer Datenbank, die als Informations- und Geschäftsverwaltungsmodell dienen sollte. Es wurde der günstigste Anbieter gewählt, der zwar das offerierte Informationssystem, jedoch keine voll funktionsfähige Datenbank erstellte.

Heute hat der Kunde Probleme, neue Datensätze einzupflegen sowie neue Selektionen und Reportings zu erstellen (was nicht weiter verwunderlich ist, da diese eine Datenbank als technische Grundlage bedingen). Hätte der Kunde das Angebot gewissenhaft evaluiert, wäre ihm dies aufgefallen. Weiter ist es sehr ratsam, sich direkt mit den bestehenden Nutzern der offerierten Software auszutauschen (Referenzen). Ihnen gelingt es, aus der eigenen Perspektive Unzulänglichkeiten und Vorteile des Systems und die Reaktionsgeschwindigkeit des Anbieters (und die Zufriedenheit mit diesem) zu benennen.

### Keine Testplattform vor dem Entscheid

Insbesondere wenn man davon ausgeht, dass Standardsoftware die gewünschten Funktionen abdeckt, sollte man darauf bestehen, den Zugang zu einer lauffähigen Testumgebung zu erhalten. Falls irgend möglich, sollte dies vor dem Entscheid für oder gegen einen Anbieter möglich sein. Dadurch können diese besser miteinander verglichen werden. Bei Software, die vollständig entwickelt werden muss, sollte zumindest eine Demonstration lauffähiger Kundensoftware möglich sein.

Wir wissen von Organisationen, die den Entscheid für einen Anbieter auf der Basis einer «Software-Attrappe» gefällt haben. Im Gegensatz zu einer funktionsfähigen Testumgebung (einem Prototypen), ist eine solche «Simulation» nicht lauffähig. Sie dient nur dazu, Funktionen der Software zu zeigen, nicht aber auszuführen. Es können auch keine Testdaten erfasst werden. Uns sind Fälle bekannt, bei welchen Anbieter lediglich «Screenshots» von Anwendungen in Präsentationen zeigten. Spätestens hier wäre Skepsis angebracht.

### **Ungenügende Evaluation der Betriebskosten**

Die reinen Beschaffungs- und Entwicklungskosten eines Systems haben nicht genügend Aussagekraft. Viel besser interpretierbar sind die «Vollkosten» über einen Zeitraum von 3, 5 oder 7 Jahren. Dabei werden alle Kosten erfasst, die nach dem «Roll out» anfallen wie etwa Lizenzen, Serverbetrieb, Release-Kosten, Schätzungen für künftige Entwicklungskosten etc.

Prüft man diese, kann sich beispielsweise zeigen, dass der «teuerste» Anbieter nach der Betriebszeit von 5 Jahren zum «günstigsten» wird. Dies kann der Fall sein, wenn zwar die Investitionskosten höher sind als bei der Konkurrenz, die jährlichen Folgekosten jedoch bedeutend tiefer ausfallen.

### **Vernachlässigung von Changemanagement-Aspekten**

Schliesslich ist nicht ausser acht zu lassen, dass insbesondere grössere Veränderungen in der Softwareumgebung auch Aspekte des Changemanagements berücksichtigen müssen. Veränderungen bei einem wesentlichen Arbeitsinstrument wie einer Datenbank bedeuten für die Anwender Anpassungen, die ihren Alltag stark beeinflussen. Es ist darum von besonderer Wichtigkeit, den Betroffenen aufzuzeigen, warum die Veränderung notwendig ist, wie der Weg der Veränderung aussieht und was nach der Veränderung besser sein wird als heute.



## **mAx** Der perfekte Klapptisch

Das neue, besonders flexible Tischprogramm mAx von Wilkhahn vereint die klassischen Vorteile von Klapptischsystemen mit der erstklassigen Wilkhahn Gestaltungsqualität. Dank seiner praktischen Einhandbedienung, der innovativen Selbststabilisierung und der integrierten Verkettungsmöglichkeit ist mAx ideal für das Einrichten von multifunktionalen Räumen. Der Entwurf von Andreas Störiko überzeugt in jedem Szenario.

**Wilkhahn**

## Fussnoten

- <sup>1</sup> <https://www.standishgroup.com>.  
<sup>2</sup> Liebhart 2009.  
<sup>3</sup> Standish Group 2013.  
<sup>4</sup> Wikipedia 2018.  
<sup>5</sup> Hastie & Wojewoda 2015.  
<sup>6</sup> Projektmanagement: Definitionen, Einführungen und Vorlagen o.J.  
<sup>7</sup> Schaefer 2011.  
<sup>8</sup> Verbändereport 2016.

## Literatur

- Hastie, S. & Wojewoda, S. (2015). *Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch*. URL: <https://www.infoq.com/articles/standish-chaos-2015> (18.06.18).
- Liebhart, D. (2009). Das Märchen von den gescheiterten IT-Projekten. *netzwoche 06/2009*. URL: [https://www.trivadis.com/sites/default/files/downloads/pr/Netzwoche\\_Carte-Blanche\\_DAL\\_090402.pdf](https://www.trivadis.com/sites/default/files/downloads/pr/Netzwoche_Carte-Blanche_DAL_090402.pdf) (18.06.18).
- Projektmanagement: Definitionen, Einführungen und Vorlagen (o.J.). *Agiles Projektmanagement*. <http://projektmanagement-definitionen.de/glossar/agiles-projektmanagement/> (18.06.18).
- Schaefer, I. (2011). *Anforderungsanalyse – Software Engineering I*. Vorlesungsunterlagen Software Systems Engineering. Technische Universität Braunschweig. URL: <https://www.tu-braunschweig.de/Medien-DB/isf/sse/v2-re.pdf> (18.06.18).
- Standish Group (2013). *Chaos Manifesto 2013. Think Big, Act Small*. URL: <https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/GENREF/S130301C.pdf> (18.06.18).
- Verbändereport (2016). *Verbandssoftware als Eier legende Wollmilchsau*. URL: <https://www.verbaende.com/verbaendereport/fachartikel/Verbandssoftware-als-Eier-legende-Wollmilchsau> (18.06.18).
- Wikipedia (2018). *Chaos-Studie*. URL: [https://de.wikipedia.org/wiki/Chaos-Studie#cite\\_note-4](https://de.wikipedia.org/wiki/Chaos-Studie#cite_note-4) (18.06.18).

## Die Autoren



### Thomas Zurkinden / [thomas.zurkinden@bvmberatung.net](mailto:thomas.zurkinden@bvmberatung.net)

Thomas Zurkinden (lic. rer. pol.) studierte Betriebswirtschaftslehre an der Universität Freiburg/CH. Nach seinem Lizentiat arbeitete er von 2002 bis 2006 am Verbandsmanagement Institut (VMI). Seit 2007 ist er als Berater für die Beratergruppe für Verbands-Management (B'VM) in Bern tätig. In dieser Funktion ist er im Mandat Generalsekretär der FAMH, des Verbands der medizinischen Laboratorien der Schweiz.



### Nadja Blanchard / [nadja.blanchard@bvmberatung.net](mailto:nadja.blanchard@bvmberatung.net)

Nadja Blanchard studierte Gesellschaftswissenschaften an der Universität Freiburg/CH mit Schwerpunkt Kommunikation. Danach sammelte sie diverse Berufserfahrungen in der Kommunikation sowie im Marketing in Kanada, Deutschland und der Schweiz. Seit 2011 ist sie bei der Beratergruppe für Verbands-Management (B'VM) als Beraterin tätig.