

Lünendonk®-Whitepaper

# Software-Modernisierung

Im Spannungsfeld zwischen  
Zwangsläufigkeit und Aufwand

Eine Publikation der Lünendonk GmbH  
überreicht durch

**TNG**  TECHNOLOGY  
CONSULTING



# Inhaltsverzeichnis

<b>VORWORT .....</b>	<b>4</b>
<b>ÄLTERE INDIVIDUALSOFTWARE UND IHRE PROBLEMATIK.....</b>	<b>5</b>
<b>MODERNISIERUNG: KEINE TRIVIALE ENTSCHEIDUNG.....</b>	<b>12</b>
<b>ALTERNATIVEN EINER MODERNISIERUNG VON ALT-SYSTEMEN UND -SOFTWARE.....</b>	<b>16</b>
<b>EXKURS: SOFTWARE-RE-ENGINEERING .....</b>	<b>21</b>
<b>ASPEKTE DER DURCHFÜHRUNG EINER MODERNISIERUNG .....</b>	<b>23</b>
<b>ZUR NOTWENDIGKEIT EINER IT-MODERNISIERUNG .....</b>	<b>27</b>
<b>CHECKLISTE IT-SANIERUNG: WO STEHT IHR UNTERNEHMEN?.....</b>	<b>30</b>
<b>INTERVIEW MIT CHRISTOPH STOCK: IT-SANIERUNG KANN KLAPPEN .....</b>	<b>31</b>
<b>AGILE MIGRATION – EIN NEUER ANSATZ! .....</b>	<b>34</b>



## Vorwort



Mario Zillmann  
Leiter Professional Services,  
Lünendonk GmbH

Sehr geehrte Damen und Herren,

Alt-Software oder Legacy-Software – sie ist den Unternehmen lieb und teuer. Lieb, weil sich die Anwender an viele Funktionalitäten gewöhnt haben und sie auch für den reibungslosen Geschäftsbetrieb notwendig ist. Teuer, weil die Kosten der IT im weitesten Sinne durch ineffiziente Applikationen in die Höhe getrieben werden. Viele CIOs klagen über zu hohe Kosten für den IT-Betrieb und über zu wenig finanzielle Möglichkeiten, in IT-Innovationen investieren zu können.

Oft ist eine Modernisierung der bestehenden Softwarelandschaft aber eine bittere Notwendigkeit. Viele Umfragen unter IT-Fachleuten und Verantwortlichen zeigen auch, dass hierfür ein hohes Bewusstsein vorliegt. Denn die Rolle der IT hat sich dramatisch verändert und tut es weiterhin. Der Wertbeitrag, den die IT für Unternehmen leistet, ist so hoch wie nie. Geschäftsprozesse und Geschäftsmodelle basieren auf einem komplexen Ökosystem von miteinander verbundenen und abhängigen Softwareanwendungen. Ihr reibungsloser Ablauf ist ein wichtiger Baustein für den nachhaltigen Markterfolg.

CIOs müssen den Spagat zwischen der „alten IT-Welt“ und den IT-Innovationen schaffen und beide Welten so miteinander kombinieren, dass der Geschäftserfolg sichergestellt wird.

Aber: Die Modernisierung von IT ist nicht ein „Zwischendurch-Projekt“. Die IT-Verantwortlichen in den Unternehmen sind oft Getriebene, die mit begrenzten Budgets und steigenden Anforderungen kämpfen: Sie müssen den Geschäftsbetrieb mit den bestehenden Anwendungen aufrechterhalten, neue Anwendungen für die Produkt- oder Vertriebswegegestaltung entwickeln oder schlicht „nur“ die Anforderungen gesetzlicher und regulatorischer Auflagen erfüllen.

In der Regel ist alles dringender und wichtiger als eine Modernisierung oder Sanierung der bestehenden IT-Landschaft gerade jetzt!

Das geht so lange gut, bis es zu ernsthaften Ausfällen kommt und damit die Prioritäten und IT-Budgets durcheinandergewirbelt werden. Das vorliegende Whitepaper ist daher ein Plädoyer für eine vorausschauende und geplante Modernisierung der Informationstechnologie in Unternehmen. Es ist in Zusammenarbeit mit den Unternehmen Kienbaum Management Consultants, Steria Mummert Consulting und TNG Technology Consulting entstanden.

Ich wünsche Ihnen eine interessante und nützliche Lektüre!

Mario Zillmann,  
Leiter Professional Services, Lünendonk GmbH

## Ältere Individualsoftware und ihre Problematik

### LEGACY-SOFTWARE – ALT-SOFTWARE MIT WERT

Der Begriff Alt-Software oder Legacy bezeichnet etablierte, historisch gewachsene Anwendungen der Unternehmenssoftware, die nicht mehr oder nur unzureichend kompatibel mit den anderen IT-Kernprozessen sind. Innerhalb der Unternehmens-IT sind es oft großrechnerbasierte ältere Individualentwicklungen oder Software auf Basis von Java oder C/C++, die außerdem vielfach durch unzureichende Dokumentation, veraltete Betriebs- und Entwicklungsumgebungen, zahlreiche Schnittstellen und hohe Komplexität gekennzeichnet sind.

Legacy-Software enthält in der Regel wertvolles Wissen und Erfahrungen, weshalb sie nicht einfach abgelöst werden kann. Es ist in der Regel eine komplexe Entscheidung, ob man Legacy-Software ablösen soll, da unterschiedliche Möglichkeiten bestehen, mit ihnen umzugehen. Denn oft wurde sie entwickelt, um ganz bestimmte Probleme der Unternehmen durch Individualsoftware zu lösen; Lösungen, die durch handelsübliche Software nicht erbracht wurden. Individualsoftware unterstützt daher vor allem die Kernprozesse der Unternehmen und gilt in der Regel als geschäftskritisch.

Diese Besonderheiten verzögern die Ablösung oder Sanierung solcher Systeme oft deutlich über eine technisch sinnvolle Lebenszyklusdauer hinaus.

In diesem Whitepaper liegt der Fokus auf der Sanierung von älteren Legacy-Systemen, die in etwa älter als zehn bis 15 Jahre sind sowie auf Softwarelandschaften, die durch Übernahmen und Fusionen oder einfach durch dezentrale Unternehmensstrukturen entstanden sind, aber die gleichen Aufgaben erledigen.

### STATUS QUO UND WERT VON IT-ANWENDUNGSLANDSCHAFTEN

Der Stand der IT in vielen Organisationen ist nicht zufriedenstellend. Neue Technologien wurden zwar eingeführt, aber die alten Systeme blieben. Diese Aussage trifft vor allem für große Organisationen zu. Typische Beispiele sind Banken, Automobilkonzerne, Telekommunikationsunternehmen, Versicherungen oder Handelsunternehmen.

Die IT-Anwendungslandschaften dieser Branchen haben gemeinsam, dass sie für die meisten Geschäftsbereiche separat und über lange Jahre aufgebaut, immer wieder erweitert und dem jeweiligen Bedarf angepasst wurden. Einzellösungen wurden aufwendig miteinander verbunden, was zu längeren Durchlaufzeiten in den IT- und Geschäftsprozessen führt und Automatisierung behindert.

Heute existiert in fast allen Großunternehmen und Konzernen ein Mosaik an Applikationen mit einer Vielzahl von Anwendungen, Datenbanken und komplexen Schnittstellen, die Prozessstörungen verursachen. Ihre IT ist im Laufe der Jahre zu einer technologisch heterogenen Applikationslandschaft herangewachsen. Da die Alt-Software sich über die Jahre akkumuliert, macht sie einen immer größeren Anteil des gesamten Softwarebestands der Anwenderunternehmen aus. Versicherungen zum Beispiel arbeiten mit besonders langfristigen Geschäftsmodellen von Jahrzehnte laufenden Lebensversicherungen. Für diese ist eine Migration der selbst entwickelten Mainframe-Applikationen mit sehr hohem Aufwand und komplexen Anpassungen der Geschäftsprozesse verbunden. Versicherungen sind in den letzten Jahren beispielsweise häufig auf SAP-Standardsoftware migriert.



## SICHT DER CIOs: STANDARDISIERUNG UND MOBILES ARBEITEN SOLL SICH KÜNFTIG (NOCH) STÄRKER DURCHSETZEN

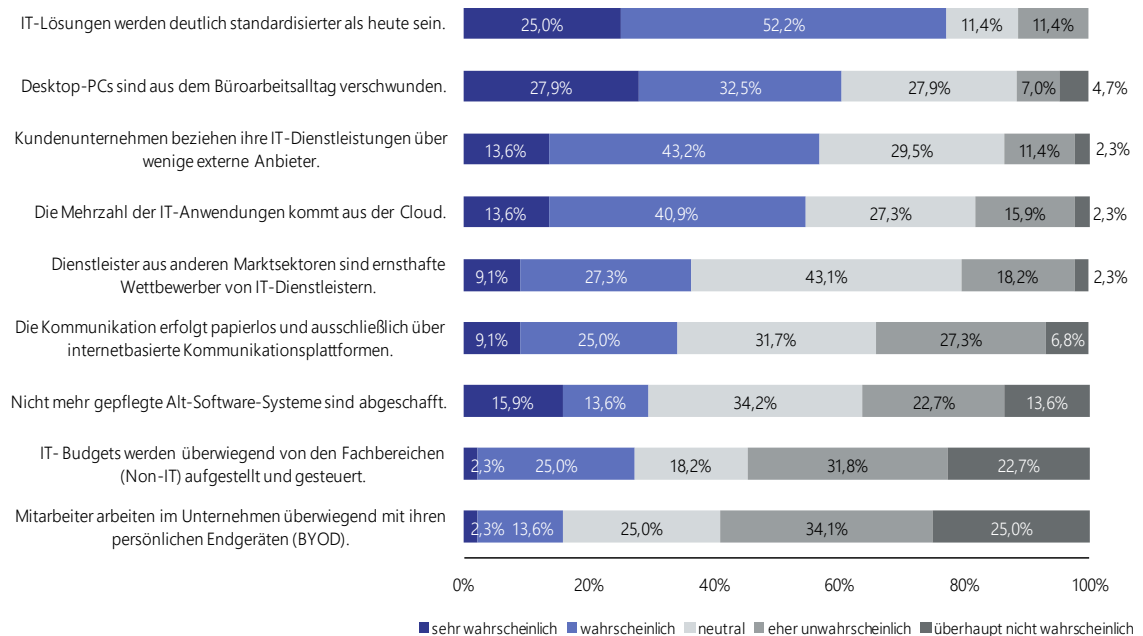


Abbildung 1: Frage: Wenn Sie an die ferne Zukunft denken, zum Beispiel an das Jahr 2020, welche der folgenden Aussagen halten Sie für „überhaupt nicht wahrscheinlich“ (-2) oder „sehr wahrscheinlich“ (+2)? n = 43, Quelle: Lünendonk

Viele ältere, für die individuellen Bedürfnisse der Unternehmen entwickelte Anwendungen lassen sich nur mit hohem Aufwand den sich ständig ändernden Erfordernissen anpassen. Das führt zwangsläufig dazu, dass junge Unternehmen mit einer einheitlichen Softwarelandschaft gegenüber den sogenannten Incumbants im Vorteil sind.

Erstere haben moderne Systeme; sie müssen keine Alt-Anwendungen aufrechterhalten, sie haben weit weniger Schnittstellenprobleme, sie haben per se die zukunftsweisende Technologie für die Anbindung an Web- und Mobile-Anwendungen.

Denn auch Software unterliegt einem Lebenszyklus von der Planung über die Entwicklung, die Wartung und Weiterentwicklung bis zur Außerbetriebnahme. Wie bei vielen Produkten gibt es dabei eine sozusagen optimale technische Lebensdauer, zum Beispiel gemessen am Verhältnis zwischen Produktivität und

Kosten. Bei Legacy-Software wird diese Lebensdauer gedehnt, mit Konsequenzen für die Gesamtkostenstruktur der IT.

### TECHNISCHE SCHULDEN SIND EIN ERNSTZUNEHMENDES PROBLEM GEWORDEN

Durch die verlängerte Lebensdauer laufen bei Alt-Anwendungen vermehrt „technische Schulden“ der IT auf. Diese „Technical Debts“ sind die Menge an Arbeit, die aus unterschiedlichen Gründen nicht erledigt wurde. Ein häufiges Beispiel ist ein duplizierter Code. Weil nicht die Zeit vorhanden war, die Applikation so umzubauen, dass der gleiche Code von zwei verschiedenen Stellen aufgerufen wird, wurde er einfach kopiert.

Das führt zu Problemen, wenn später nur eine dieser Stellen geändert wird, beispielsweise, weil ein Fehler behoben wird, der aber dann in dem zweiten Code-Teil noch enthalten ist. Es entstehen also kumulierte Unzulänglichkeiten in der Applikationsentwicklung.

Entwickler dokumentieren diese technischen Schulden in der Regel nicht, sodass keine Transparenz darüber besteht, wo Nachbesserungsbedarf in den Applikationen besteht und wie der Zustand der Anwendung ist. Weitere Gründe, warum technische Schulden entstehen, sind Zeit- und Ressourcenmangel in der Entwicklung sowie ein nicht optimales Zusammenspiel von Hardware und Software. Das Problem wird auch dann größer, wenn externe Entwickler beteiligt sind, die später nicht mehr zu dem Entwicklungsprojekt befragt werden können.

Technische Schulden ergeben sich auch nahezu zwangsläufig bei einer geschäftsgetriebenen, schnellen Weiterentwicklung der Unternehmens-IT, deren Wertbeitrag vorrangig an der Umsetzung von Projekten und der Time to Market gemessen wird – nicht an der Bereitstellung einer sauberen IT-Architektur.

Technical-Debt-Management fristet ein eher stiefmütterliches Dasein in Unternehmen; aber Code-Fehler können teuer werden. Wer also „technische Schulden“ abbaut, entlastet die Entwicklungsbudgets. Nur wer regelmäßig fehlerhafte Altlasten sucht und bereinigt, kann mit optimaler Verfügbarkeit der IT-Infrastruktur, zuverlässiger Time to Market und Stabilität der Applikationen rechnen. Vorausschauende CIOs plädieren daher für eine rollierende Modernisierung der Anwendungen, statt einzelfallbezogene, unplanbare Ad-hoc-Projekte durchführen zu müssen.

Folglich werden die IT-Prozesse vieler Unternehmen auch in Zukunft Alt-Systeme beherbergen. Eine aktuelle Befragung von Lünendonk unter CIOs aus Großunternehmen und Konzernen ergab, dass nur etwa ein Drittel der Meinung ist, im Jahr 2020 sind alle Alt-Systeme modernisiert. Ein weiteres Drittel traut sich nicht zu, hierzu eine konkrete Meinung abzugeben und das letzte Drittel findet, es ist unwahrscheinlich, dass Alt-Software bis zum Jahr 2020 komplett modernisiert ist.

### **Die Altersstruktur der IT-Landschaft beeinflusst die IT-Kosten**

Die Altersstruktur der Applikationslandschaft beeinflusst auch die Kostenstruktur der gesamten IT. Eine typische Kostenstruktur zum Beispiel in Großbanken ist, dass nur ein knappes Drittel des IT-Budgets für Veränderungs- und Zukunftsprojekte verbleibt; fast 70 Prozent der Kosten dienen der Aufrechterhaltung des Betriebs (Abbildung 2). Auch laut einer Gartner-Studie von 2011 sind mehr als zwei Drittel der IT-Budgets von Unternehmen für die Wartung der bestehenden Infrastruktur reserviert.

Alt-Anwendungen sind teuer: Es kostet immer mehr Ressourcen, sie im Laufe der Zeit zu warten. Die Wartung von Software kann gemessen über die gesamte Lebensdauer eines Systems 60 Prozent der Kosten verschlingen (im Vergleich zu 40 Prozent Entwicklungskosten). Wartungsarbeiten haben dabei verschiedene Aufgaben. Die korrektive Wartung dient der Verbesserung von Fehlern. Präventive Wartung soll Fehler im Produktivbetrieb verhindern. Adaptive Wartung passt die Anwendung an eine veränderte Umgebung (Hardware, Software) an und die verbessernde Wartung überarbeitet die Applikation, zum Beispiel hinsichtlich Performance, Speicherbedarf, Anpassung der Graphical User Interfaces (GUI).

### **CIOs HABEN ZU WENIG GELD FÜR INNOVATIONEN**

Der Anteil von Alt-Software beeinflusst die Gesamtkostenstruktur der IT. Für den Betrieb der IT-Landschaft inklusive Wartung und Pflege sowie Upgrades und Aktualisierungen werden nach Schätzungen von Lünendonk fast 60 Prozent des Gesamt-IT-Budgets (einschließlich Hardware) ausgegeben; für Innovationen und neue Projekte bleibt dann nur noch ein gutes Drittel des Budgets übrig.

Je mehr Zeit und Geldmittel nun die Unternehmen für die Wartung älterer Software aufwenden müssen, desto weniger bleibt für die Realisierung zukunftsweisender Projekte übrig. Damit geraten sie in einen Teufelskreis: Neue Projekte werden unter höchster Priorität



vorangetrieben, wobei wiederum für eine saubere Dokumentation und Architektur wenig Zeit kalkuliert ist. Die technischen Schulden der IT steigen.

Was ist billiger? Die alten Systeme – sofern sie zumindest stabil sind – mit steigenden Kosten weiterlaufen zu lassen? Oder sich in das Abenteuer einer Software-Modernisierung zu begeben? Die Pro- und Contra-Argumente gehen dabei allerdings über eine einfache monetäre Betrachtung hinaus.

Fakt ist, dass viele Unternehmen ihre Hausaufgaben hinsichtlich der IT-Effizienz bereits erledigt haben, wodurch sich die Betriebskosten verringern und mehr Mittel für Innovationsprojekte zur Verfügung stehen.

Laut der aktuellen Lünendonk®-Studie „Der Markt für IT-Beratung und IT-Service“ investieren die befragten CIOs das freigewordene Budget in Change-the-Business-Projekte wie IT-Beratung, Systemintegration sowie Softwareentwicklung und -anpassung. Mehr als die Hälfte (52 Prozent) plant, das IT-Budget für Innovations- und Anpassungsprojekte zu erhöhen. An den Ergebnissen zeigt sich, dass CIOs in den letzten Jahren

bereits deutliche Effizienzerfolge erzielt haben, indem sie ihre IT-Landschaft durch Virtualisierung, Automatisierung und Standardisierung verbessert haben und weniger Kapital in den IT-Prozessen binden. Die Software-Modernisierung mit all ihren Facetten (Re-Engineering, Ablösung durch Standardsoftware etc.) ist ein wesentlicher Teil dieser Effizienzprojekte. Diese Maßnahmen sind auch dringend nötig, denn CIOs sind gefordert, ihren Wertbeitrag zu erhöhen und mehr Ressourcen für Innovations- und Transformationsprojekte auf dem Weg zum digitalen Unternehmen zu schaffen. Allerdings sehen sich viele IT-Leiter noch in einer eher kostenerzeugenden Rolle als in der wertschöpfenden Funktion.

Etwa zwei Drittel ihres IT-Budgets reservieren CIOs allein für den Betrieb der IT. Darin sind auch die Kosten für die Pflege und Weiterentwicklung der Applikationslandschaft (neue Releases, Konsolidierung, Standardisierung etc.) berücksichtigt. Je komplexer und fehleranfälliger die Anwendungslandschaft ist, umso höher sind deren Betriebskosten und umso weniger kann für Innovationsprojekte bereitgestellt werden.

### CIOs VERLAGERN IHRE BUDGETS ZU CHANGE THE BUSINESS

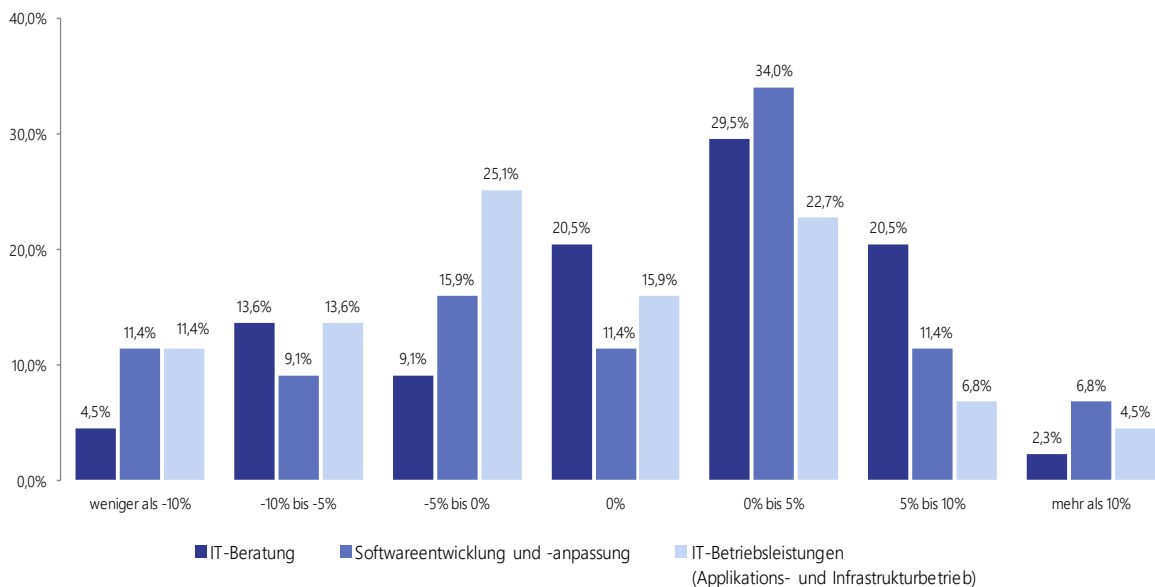


Abbildung 2: Frage: Wie werden sich Ihre IT-Budgets 2015 entwickeln? n = 44, Quelle: Lünendonk



### Das Problem von Legacy-Software:

#### Strukturelle Zukunftsunfähigkeit

Trotz ihres aktuellen Nutzens für die Unternehmen ist Alt-Software strukturell zukunftsunfähig. Das liegt an ihren Charakteristika und den daraus erwachsenden Risiken für das eigentliche Geschäft.

Charakteristische Eigenschaften und Schwächen von Alt-Software und -Systemen

#### Technik/Architektur

- Die Programme sind, da sie über längere Zeiträume entstanden sind, meist komplex.
- Alt-Programme folgen vielfach nicht den Best Practices der IT Infrastructure Library (ITIL).
- Es existieren zahlreiche nicht standardisierte Schnittstellen zu anderen Systemen. Hierdurch entstehen bei deren Kommunikation oftmals Prozessbrüche und die Systeme erfordern einen hohen manuellen Betriebsaufwand.
- Sie sind oft unvereinbar mit heute geltenden Best Practices für Architektur und Programmiermodelle (relationale Datenbanken, NoSQL, funktionale Programmierung, Objektorientierung) und Ergonomie (grafische Benutzeroberfläche, kontextsensitive Hilfen u.a.).
- Es mangelt an Tests und an Möglichkeiten zum automatisierten Testen.
- Automatisierung der Geschäftsprozesse und digitale Geschäftsmodelle sind nur eingeschränkt möglich.

#### Betrieb und Wartung

- Oft sind die Alt-Anwendungen Eigenentwicklungen. Die verwendeten Werkzeuge und Methoden sind individuell, lassen sich aber durch den Einsatz moderner Entwicklungswerkzeuge sanieren.
- Ihre Komplexität erschwert und verzögert selbst einfache Änderungen.
- Alt-Anwendungen haben im Vergleich zu vielen Standardsoftware-Systemen einen höheren Wartungsaufwand. Dies hängt unter anderem mit einem geringeren Automatisierungsgrad zusammen.
- Es ist schwierig, aktuelle Versionen von Individualsoftware in Produktion zu bringen und zu integrieren.

- Änderungen an der Software haben häufig Fehler im Produktivbetrieb zur Folge und ihre Behebung dauert tendenziell länger.
- Mangelnde Funktionalitäten oder instabile Schnittstellen können den Anwendern Ergänzungsarbeiten mit Hilfssystemen wie Excel und Word abverlangen; die Fehleranfälligkeit steigt.
- Bei lizenzierten Alt-Anwendungen laufen Garantien, Support und Lizenzen aus. IT-Abteilungen haben vielfach auch das demografische Problem, Mitarbeiter für die Pflege der Alt-Anwendungen zu finden.

#### Know-how

- Die Dokumentationen sind unübersichtlich, unvollständig, veraltet oder gar nicht mehr vorhanden.
- Vielfach ist keine oder eine falsche Code-Kommentierung vorhanden.
- Der ursprüngliche IT-Entwickler und der kenntnisreiche Anwender sind bereits ausgeschieden.
- Es mangelt an Entwicklern, die diese Systeme noch warten bzw. an aktuelle gesetzliche Anforderungen anpassen können.
- Es fehlt an aktuellem Wissen über die Struktur und Funktionsweise der Alt-Anwendungen.
- Oftmals basieren sie auf veralteten Programmiersprachen.
- Das Unternehmen verfügt nur noch über begrenztes Verständnis des Gesamtsystemzusammenhangs.

#### Flexibilität und Anpassungsfähigkeit

- Die IT-Anwendungslandschaften werden heute entwickelt, um zum Beispiel Mobile Business, Cloud Services und webbasierte Applikationen zu unterstützen; vielen Alt-Anwendungen ist dies nicht mehr möglich.

Risiken aufgrund dieser Schwächen der Legacy Software

#### Sicherheitslücken

Stammen die älteren Systeme von externen Lieferanten, deren Support für diese Version ausgelaufen ist, kann es an Patches und Sicherheitsupdates fehlen, was das gesamte System empfindlich gegenüber Angriffen von außen macht.



Verborgene Kosten und Abhängigkeiten

Alt-Anwendungen laufen gerade bei Banken auch auf ihren angestammten technischen Umgebungen (Hardware, Datenbanken, Libraries, Operating Systems). Die Aufrechterhaltung dieser besonderen Infrastruktur ist aufwendig. Anpassungen an Veränderungen verursachen ein Vielfaches der Kosten im Vergleich zu modernerer Software. Für andere Branchen gelten diese Probleme nicht so stark, gleichwohl die Betriebskosten im Vergleich zu Standardsoftware tendenziell geringer sind.

Teile von Alt-Systemen sind maßgeschneidert; stammt die Software von externen Lieferanten, gerät das Anwenderunternehmen in eine verstärkte Abhängigkeit oder der Lieferant bietet gar keine Unterstützung mehr an. Die tatsächlichen Kosten von Alt-Systemen können verborgen bleiben, wenn man ihnen nicht die notwendigen manuellen Ergänzungsarbeiten oder den Aufwand an zusätzlicher Schnittstellenprogrammierung zurechnet. Ihre Performance wird dann falsch dargestellt.

Mangelnde Compliance

Wirtschaftsprüfer oder IT-Berater, die häufig die Sicherheit und die Regelkonformität von IT-

Anwendungen beurteilen sollen, sind mit der Testierung von Alt-Systemen überfordert. Der Aufwand der Prüfung erhöht sich. Gerade bei kernprozessnahen Anwendungen hat die IT-Prüfung in den letzten Jahren stark zugenommen.

Gefährdung der Kontinuität des laufenden Business

Legacy-Software kann die Kontinuität des Geschäftsbetriebs gefährden, wenn Fehler auftreten und diese durch das Fehlen von Komponenten oder durch zu hohe Komplexität nicht schnell bereinigt werden können.

Fehlende Unterstützung des zukünftigen Business

Alt-Lösungen sind nicht skalierbar oder nicht offen konzipiert. Die IT-Abbildung neuer Produkte für das Business ist schwierig. Sie sind langsam bei der Neueinführung von Produkten und Prozessen (Time to Market) und bei der Reaktion auf Änderungen der gesetzlichen Rahmenbedingungen. Sie sind nur beschränkt anpassungsfähig; Änderungen der Geschäftsprozesse, wie die Einführung neuer Vertriebswege oder neuer Produktkonfigurationen, werden nur eingeschränkt durch Alt-Anwendungen unterstützt.

**TOP-INVESTITIONSTHEMEN DER CIOs 2015**

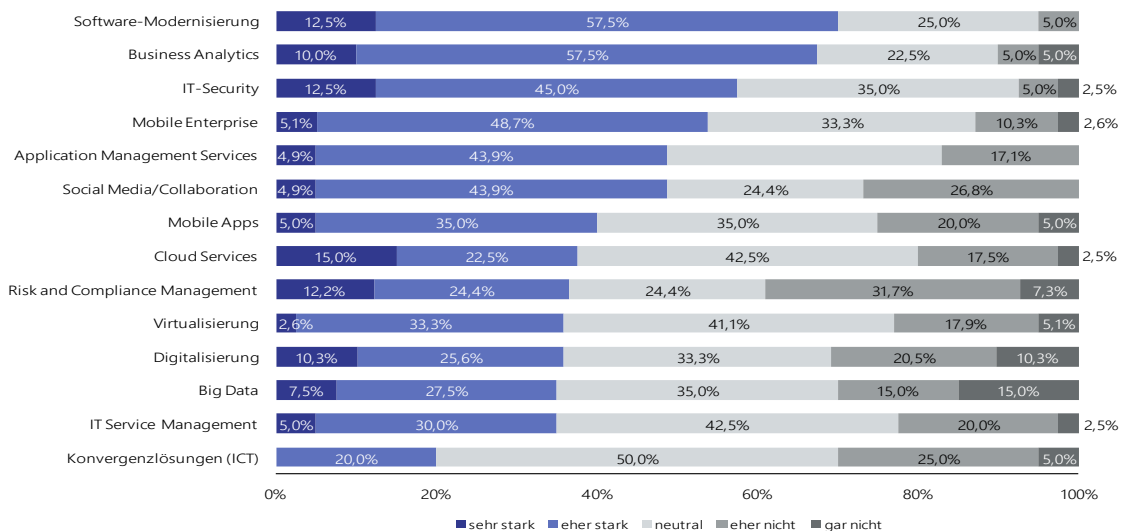


Abbildung 3: Frage: Bei welchen Themen planen Sie Investitionen – 2015? n = 39, Quelle: Lünendonk



### **IMPERATIV: ALTE SOFTWARE MODERNISIEREN!**

Als Konsequenz aus diesen Schwächen und Risiken muss grundsätzlich mit Blick auf eine langfristig tragfähige Lösung die Modernisierung der IT-Systemlandschaften empfohlen werden.

Insofern ist es nicht verwunderlich, dass Software-Modernisierung in der „ewigen Hitliste“ der IT-Chefs stets eine prominente Position einnimmt. Dabei ist zu berücksichtigen, dass die Prioritäten in der Regel von Modethemen und den jeweiligen Anforderungen der Fachbereiche dominiert werden.

Das Thema Software-Modernisierung bleibt aber hochaktuell. Einen aktuellen Ausblick auf die Zukunft der Software-Modernisierung erlaubt die Lünendonk®-Studie 2014 „Der Markt für IT-Beratung und IT-Service in Deutschland“: Die Hälfte der befragten IT-Beratungen erwirtschaftet bis zu 10 Prozent des Umsatzes mit Software-Modernisierung. Und für die Zukunft erwarten 83 Prozent von ihnen eher starke bis sehr starke Investitionen für IT-Services im Rahmen der Software-Modernisierung bei ihren Kunden. Dabei wurden unter „Software Modernisierung“ die unterschiedlichen Varianten wie Re-Engineering, Ablösung durch Standardsoftware oder komplettes Abschalten subsumiert.

Die Top-Investitionsthemen der untersuchten Kunden-Unternehmen werden 2015 folgende Inhalte haben:

- Software-Modernisierung
- Business Analytics
- IT-Security
- Mobile Enterprise
- Application Management

Bereits 2014 stand die Software-Modernisierung an der zweiten Stelle der CIO-Liste der hochpriorisierten IT-Investitionen. 2015 schiebt sie sich an die erste Position; 70 % der befragten Firmen sehen sehr starken bzw. eher starken Bedarf an Modernisierung ihrer Software; das sind immerhin 6 Prozentpunkte mehr als im Vorjahr. Bemerkenswert ist, dass viele der heute propagierten Modethemen damit auf hintere Ränge verdrängt wurden. Dies gilt insbesondere für Big Data und Konvergenzlösungen von Informations- und Kommunikationstechnologie und in gewisser Weise auch noch für Cloud Services, Mobile Apps und Social Media. Eine gewisse Dringlichkeit ist also dem Thema Modernisierung von Alt-Software nicht abzustreiten.

Beim Thema Software-Modernisierung sind sich auch Anbieter und Kunden einig. Die drei Spitzen-Investments aus Sicht der Anbieter („Software-Modernisierung“, „Business Analytics“ und „IT-Security“) werden von der Mehrheit der befragten CIOs bestätigt, wengleich die Anbieter grundsätzlich Kundeninvestments in ihrer Planung höher einschätzen, als es die Kunden tatsächlich planen.



## Modernisierung: Keine triviale Entscheidung

### LEGACY – RISIKOTRÄCHTIG VERWEBT MIT ALTEN ARCHITEKTUREN UND DATENBANKEN

Allerdings gestaltet sich die Modernisierung und Ablösung von Alt-Systemen nicht einfach. Betrachtet man die IT-Anwendungen eines großen Unternehmens – zum Beispiel einer Bank – im Überblick, wird schon intuitiv deutlich, dass in einer über Jahre gewachsenen Systemlandschaft die Modernisierung von Software hochkomplex ist (Abbildung 4). In vielen anderen

Branchen wie Handel, Automobilindustrie oder Versicherungen gestaltet sich die Komplexität ähnlich. Zunächst sind da die Befürchtungen, mit Modernisierungen das laufende Geschäft durch den Eingriff in laufende, bewährte und oft geschäftskritische Prozesse zu stören. Viele der alten Anwendungen dienen der Kundenverwaltung, der Geschäftspartnerbetreuung und unterstützen Millionen finanzielle Transaktionen.

### AUS ALT MACH NEU!

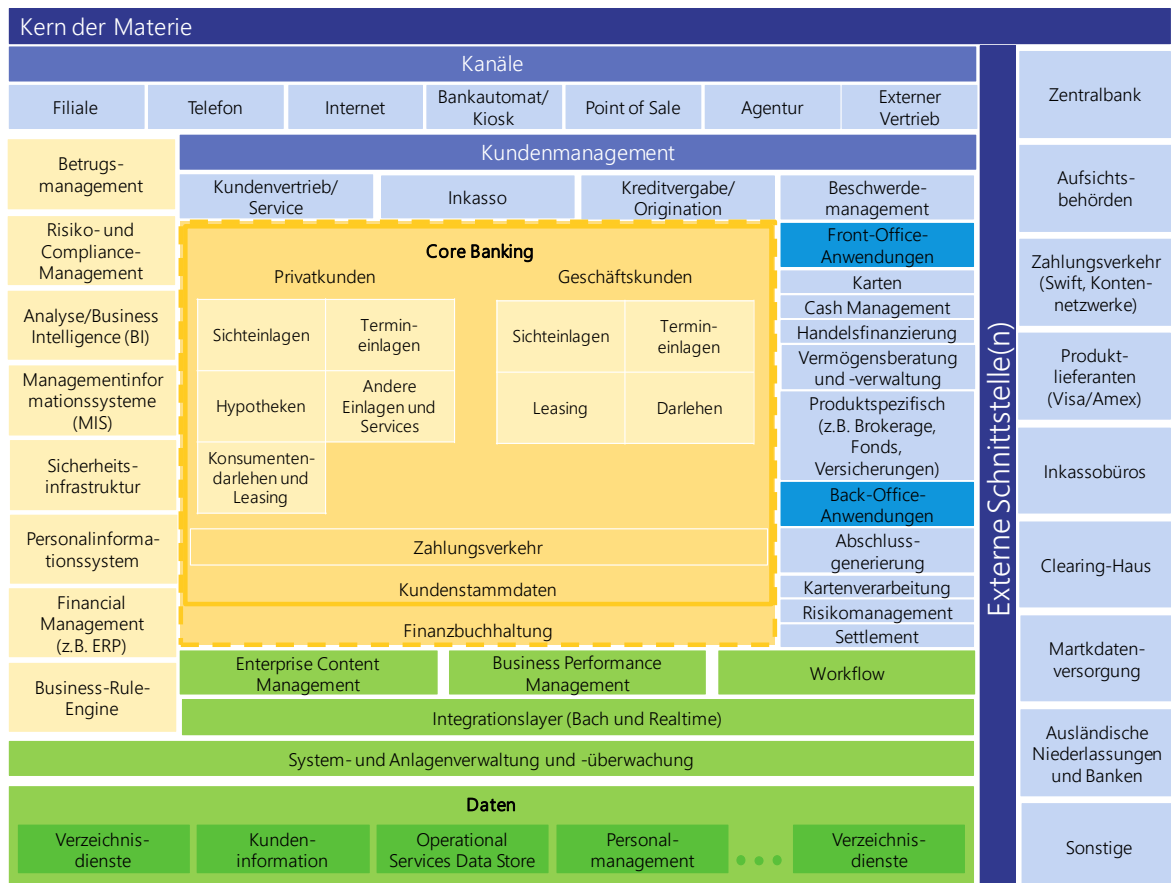


Abbildung 4: Banken-Applikationen in ihrer Einbettung in das Gesamtsystem, Quelle: Deloitte, Aus alt mach neu!



Ältere IT-Anwendungen sind vielfach mit neuen Anwendungen verbunden. Änderungen an einer Stelle können unvorhergesehene Fehler an anderer Stelle verursachen. Probleme, die sich durch Umstellungen auf neuere Systeme ergeben, könnten schnell beachtliche Ausmaße annehmen. Die Auswirkungen von Softwarefehlern bleiben schließlich nie klein – sondern werden hochskaliert durch Millionen Transaktionen der Datenverarbeitung.

Zudem ist in den Alt-Systemen sehr viel Know-how des Unternehmens und der Unternehmensabläufe kodifiziert, das mit neuen Systemen erst mühsam wieder abgebildet werden müsste. Legacy-Software und Geschäftsprozesse in den Unternehmen sind enger miteinander verknüpft, als es möglicherweise vorhandene Dokumentationen und Wahrnehmungen offenbaren. Die Alt-Software enthält das Wissen über Abläufe, Geschäftsregeln und Ausnahmen.

#### SOFTWARE-MODERNISIERUNG ERFOLGT MEIST IN ETAPPEN

Daher wählen viele Großunternehmen auch einen schleichenden Prozess der Modernisierung, indem sie im Rahmen von IT-Projekten die angrenzenden Alt-Systeme, sofern möglich, abstellen, modernisieren oder integrieren. So hoffen sie auf eine sukzessive

Lösung des Modernisierungsproblems. Ferner ist ein Business Case einer kompletten IT-Sanierung durch Modernisierung der Alt-Systeme aus Sicht vieler IT-Entscheider teurer als die Wartung der Alt-Systeme. Tatsächlich rechnet sich eine IT-Modernisierung durchaus, wenn dadurch hohe Wartungs- und Weiterentwicklungsbudgets reduziert werden können, auch, um sie in IT-Innovationen zu reinvestieren.

#### „WAR FOR TALENT“ EINMAL ANDERS: FREELANCER-ALT-KNOW-HOW WIRD KNAPP

In dieser Hinsicht bringen sich CIOs allerdings auf mittlere Sicht in Zugzwang. Die für die Pflege der Anwendungen notwendigen Ressourcen, sprich die Systementwickler und Programmierer mit dem entsprechenden Know-how, sind im Unternehmen bereits rar geworden und am Markt nur noch begrenzt verfügbar. Der Bedarf wird in einigen Branchen über IT-Freelancer gedeckt. In Zusammenhang hiermit gesehen werden muss dann die aktuelle Altersstruktur von IT-Freelancern: Nahezu ein Viertel dieser IT-Freelancer in Deutschland war 2014 über 50 Jahre alt; ein Zeichen für die Konjunktur älterer Programmiersprachen und Anwendungen (Lünendonk GmbH, Der Markt für Rekrutierung, Vermittlung und Steuerung freiberuflicher IT-Experten in Deutschland, 2014) (Abbildung 5).

#### ALTERSGRUPPEN DER IT-FREELANCER

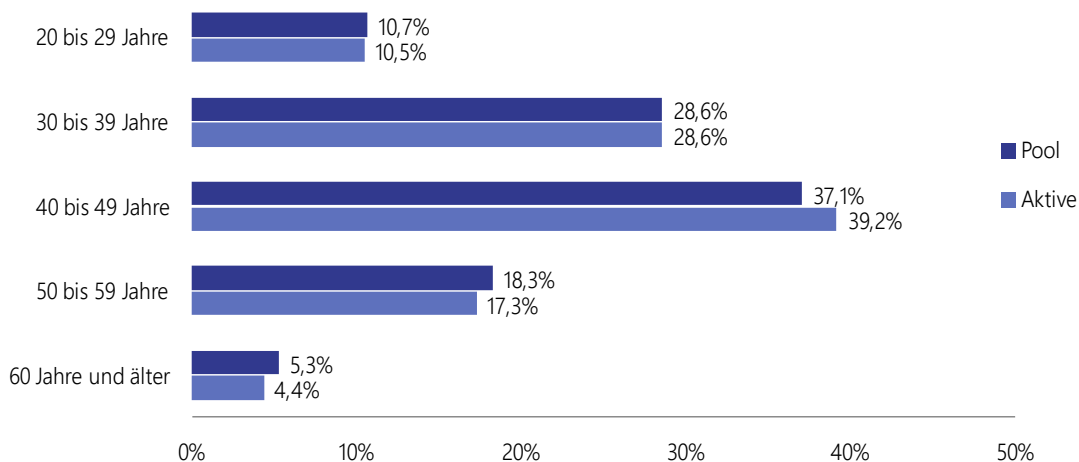


Abbildung 5: Frage: Welchen Altersgruppen gehören Ihre freien IT-Experten an? Mittelwerte, n = 18, Quelle: Lünendonk



#### **Einige illustrierende Beispiele für Konsequenzen von Softwarefehlern**

Eine Software schluckte ein Drittel der Neuversicherungsanträge: Rund ein Drittel der Versicherungsanträge konnte zunächst nicht verarbeitet werden. Aufgrund von Softwarefehlern blieben die im Portal ausgefüllten Antragsformulare liegen und erreichten die zuständigen Versicherer nicht.

Banken überweisen Geldbeträge doppelt: Bei Tausenden Kunden überwies ein Bankenverbund in Deutschland Geldbeträge doppelt. Ursache war ein Softwarefehler in einem Programm, das die Konten auf den neuen Standard des einheitlichen europäischen Zahlungsverkehrs (SEPA) umstellte.

Softwarefehler sorgten für einen Totalausfall: Ausgangspunkt der Probleme war der „Securities Information Processor“, der die Aktienkurse mit anderen Börsen austauscht

#### **ZÖGERN UND VERZÖGERN BEI MODERNISIERUNGEN ERHÖHEN DIE TECHNISCHEN SCHULDEN**

Viele der Systeme und Alt-Anwendungen mit historisch immer weiter gewachsenen Strukturen sind zwar wenig produktiv – verglichen mit neuen Anwendungen „aus einem Guss“. Aber: Sie erfüllen (noch) ihren Zweck. Substanzielle Erneuerungen sind aufgrund der oft vorherrschenden Komplexität unterblieben oder zu spät in Angriff genommen worden.

Etliche Unternehmen haben auch nicht die notwendigen freien Ressourcen für großskalige Software-Modernisierungen. Oft sind sie durch den regulären Betrieb bereits weitgehend ausgelastet – und durch dringende neue Projekte im Grunde auf Jahre hinaus ausgebucht.

Solange die Alt-Anwendungen also noch ihren Zweck erfüllen, wird oftmals keine unmittelbare Notwendigkeit für ihre Modernisierung oder Ablösung gesehen.

Stattdessen werden aufkommende Probleme mit Patches und Übergangslösungen schnell und möglichst ohne großen Aufwand scheinbar gelöst. Die „technische Schuld“ der unternehmenseigenen IT nimmt auch aus diesem Grunde stetig zu.

Drei – falsche – Top-Begründungen für eine Zögerlichkeit bei Software-Modernisierungen und ihre Gegenthesen:

1. Zeit- und Ressourcenmangel: Die zwei am häufigsten genannten Gründe, sich der Modernisierung von Alt-Anwendungen zu verweigern, sind „keine Zeit“ und „keine Ressourcen“. In der Regel gibt es dringendere Prioritäten in der Unternehmens-IT, die entweder aufgrund tagesaktueller Unterstützung des Business oder wegen geschäftskritischer IT-Projekte Vorrang haben.

Gegentese: Die Situation mit den Alt-Systemen wird aber durch Nichthandeln nicht besser. Eher schlimmer: Vorhandenes Know-how von Alt-Anwendungen wandert aus dem Unternehmen ab bzw. fällt, auch unternehmensextern, aufgrund der Demografie weg. Die Anforderungen an die IT-Umgebung werden höher.

Mit der Zeit werden die Probleme, die durch Alt-Anwendungen verursacht werden, drängender. Dann sind sie unverhofft auf der Agenda, da sie den Geschäftsbetrieb empfindlich stören und Wettbewerbsnachteile verursachen können. Besser ist, Modernisierung vorausschauend zu planen. Der CIO muss hier regelmäßig Überzeugungsarbeit in der Unternehmensführung leisten.

2. Never touch a running system: Eine oft gehörte Aussage von CIOs: Lläuft doch!

Gegentese: Lläuft noch! In der Regel wird es immer schwieriger für Legacy-Anwendungen, mit dem Rest einer modernen IT-Landschaft Schritt zu halten. Kurzfristig mag Kapselung bzw. Wrapping von Alt-

Anwendungen helfen; eine Dauerlösung ist es selten. Der alte Leitsatz der IT, „Never touch a running system“, ist in modernen IT-Abteilungen durch „Always run a changing system“ ausgetauscht. Durch den Einsatz agiler Entwicklungsmethoden lassen sich Anwendungen sanieren und in die anderen IT-Prozesse sauber integrieren.

3. Verlust von wertvollem Investment: Wertvolles altes Know-how ist in den Alt-Anwendungen kodifiziert. Dort sind kritische Geschäftsprozesse abgebildet, unternehmenswichtiges Know-how ist festgeschrieben, begründete Ausnahmeregelungen sind dargestellt und

insgesamt wichtiges Betriebs-Know-how des Unternehmens ist verankert.

Gegenthese: Ein nur zu wahres Argument. Allerdings: Der Zugang zu den Alt-Anwendungen wird immer schwieriger, immer weniger Programmierer wissen, wo welche Informationen abgelegt sind, welche Regeln angewendet wurden und wo diese im Code verborgen sind. Besser ist es, geplant die Ablösung von Alt-Software zu betreiben als später in die Falle einer notwendigen Softwarearchäologie zu tappen und mühsam ehemals kodifiziertes Wissen zu verstehen und auf neue Anwendungen übertragen zu müssen.



## Alternativen einer Modernisierung von Alt-Systemen und -Software

Modernisierungen der Alt-Systeme (Plattformen und Anwendungssoftware) können unterschiedlich radikal erfolgen. Ein Extrem ist die bewusste Beibehaltung der Alt-Systeme trotz eventueller Nachteile. Das andere ist die Komplettvergabe an einen externen Anbieter. Dazwischen gibt es eine Reihe von Möglichkeiten von der Kapselung wertvoller Legacy-Elemente zur Nutzung in neuen Umgebungen über die Migration von Altanwendungen in vielfältigen Formen bis zum Einsatz von Standardsoftware.

### OPTIONEN FÜR SOFTWARE-MODERNISIERUNGEN

#### **Kompromisslose Beibehaltung des Alt-Systems – aber mit Problemen**

Alt-Systeme können trotz aller typischen Schwächen wertvoll sein. In ihnen liegen wertvolle, kodifizierte Informationen über die Kerngeschäfte des Unternehmens. Die Neuprogrammierung mit moderneren Programmiersprachen wäre in Einzelfällen unverhältnismäßig aufwendig. Die Investition rechnet sich schlichtweg nicht: Der Aufwand für Programmierung und Implementierung einer neuen Software übersteigt in manchen Fällen den Wartungsaufwand der alten. Daher versuchen einige Unternehmen, manche Alt-Anwendungen beizubehalten.

Die Beibehaltung des Alt-Systems in der gewohnten Umgebung spart die Kosten einer Umstellung. Das System bleibt wie es ist und es erfährt den notwendigen Aufwand an Maintenance und Weiterentwicklung. Oftmals – zum Beispiel im Bankensektor – sind Alt-Anwendungen auch oder gerade wegen der geringen Anzahl ihrer Funktionalitäten kein besonderes Problem

im laufenden Betrieb, da der Wartungsaufwand gering ist.

Schwieriger wird es allerdings bei der Pflege und Anpassung der Systeme, für die Ressourcen am Markt nur begrenzt verfügbar sind. Aufgrund der vielfältigen Korrekturen und Anpassungsmaßnahmen über den gesamten Softwarelebenszyklus werden die Anwendungen und Systeme allerdings in der Regel immer komplexer (Software-Entropie) und damit auch teurer im Unterhalt.

Zudem muss sich das Unternehmen auch mit den bereits angeführten Nachteilen der Alt-Systeme arrangieren. Die begrenzten und schwerfälligen Möglichkeiten einer Anpassung der Software an neue technische und geschäftliche Anforderungen macht den Ausbau von Vorteilen gegenüber der Konkurrenz schwierig.

Eine Linderung dieser Nachteile kann durch die Anwendung von „Softwarearchäologie“ erfolgen. Softwarearchäologie dient der Wartung von Alt-Software, die nicht sauber dokumentiert ist. Mit „Reverse Engineering“ von einzelnen Softwaremodulen sowie Werkzeugen und Techniken zum Verständnis von Programmstrukturen deckt diese Methode die ursprünglichen Design-Informationen auf. Softwarearchäologie kann zudem Fehler in der ursprünglichen Software aufdecken, auch solche, die bisher nicht zu Problemen geführt hatten.

Auch teilweises Beibehalten der Alt-Anwendungen kann unter bestimmten weiteren Umständen eine bedenkenswerte Option sein.





### Kapselung oder Wrapping von Alt-Software

Beim Wrapping wird die Legacy-Software umschlossen und über Schnittstellen der Umgebung zur Verfügung gestellt. Teile der Alt-Software werden beibehalten, aber in neue Umgebungen von Programmen und/oder Plattformen eingebettet. Die Legacy-Software mit ihren Nachteilen bleibt somit im Kern bestehen und muss weiterhin gewartet werden. Der Vorteil ist, dass kodifiziertes Wissen der Alt-Programme erhalten bleibt, aber gleichzeitig die kürzeren Laufzeiten, geringeren Betriebskosten und die höhere Sicherheit der neuen Systeme genutzt werden können.

COBOL zum Beispiel gilt als veraltet, nur für Mainframes und nicht in moderne IT integrierbar. Die Praxis hat gezeigt, dass es vielfach einfacher und kostengünstiger ist, COBOL-Programme zu modernisieren und zu kapseln als Anwendungssoftware neu zu schreiben. Eine Umfrage der Computerworld unter mehr als 350 IT-Managern (2012) fand heraus, dass in 54 Prozent der Fälle mehr als die Hälfte der intern entwickelten transaktionsreichen Geschäftsanwendungen COBOL-Applikationen sind. Insbesondere in Banken, Versicherungen, in der Touristik oder in den Finanzverwaltungen basieren viele IT-Systeme auf dieser Programmiersprache. Für die Unternehmen bleiben damit Investitionen der Vergangenheit geschützt.

### Neuentwicklung: Greenfield und agile Softwareentwicklung

Vielfach erfordern jedoch die Umstellung auf cloudbasierte Strukturen und die Anforderungen der Fachabteilungen nach State-of-the-Art-Anwendungen für das Business einen radikaleren Ansatz der Software-Modernisierung. Die Neuentwicklung „auf der grünen Wiese“ ersetzt die alte Software komplett. Aber: Eine solche Neuentwicklung ist zeitaufwendig, kostenintensiv, schwer abzuschätzen und nicht ohne Risiko.

### Cold Turkey

Eine vollständige Neuentwicklung ist sinnvoll, wenn Zeit sowie finanzielle und personelle Ressourcen ausreichend verfügbar sind und andere Optionen – aus welchen Gründen auch immer – schlechter erscheinen. Denn die Kosten solcher Entwicklungsprojekte sind oft höher als die von Re-Engineering-Projekten. Zudem erreichen nicht alle diese Projekte ihre Ziele.

### Re-Engineering

Re-Engineering kann als Gegenstück zur kompletten Neuentwicklung betrachtet werden. Hierbei werden zunächst in einer Art Software-Analyse der Alt-Systeme (Reverse Engineering) der Programmcode und die Funktionalitäten der zu modernisierenden Software sozusagen wieder aufgedeckt. In einer zweiten Phase, dem Forward Engineering, werden die Funktionalitäten dann auf modernen Umgebungen mit zeitgemäßen Programmen realisiert.

### Agile Softwareentwicklung

Eine Möglichkeit, neue Anwendungen zu entwickeln, ist die agile Softwareentwicklung. Agile Softwareentwicklung als Methode setzt möglichst viele Rückkopplungsprozesse und zyklisches (iteratives) Vorgehen auf allen Ebenen der Programmierung, der Zusammenarbeit im Team und beim Management des Entwicklungsprojekts ein.

Dabei wird das neue System nicht in allen Einzelheiten genau geplant und dann in einem einzigen Durchgang entwickelt, sondern kurze Planungs- und Entwicklungsphasen wechseln ab. Der Grund ist, dass Anforderungen oft zu Projektbeginn noch gar nicht vollständig bekannt sind und sich zudem während der Projektlaufzeit noch ändern können. Nachdem die Ziele, die mit der Software erreicht werden sollen, festgelegt und gewichtet sind, wird der Plan für eine erste Version ausgearbeitet, die Entwicklung beginnt, Anpassungen erfolgen später.



### Software-Modernisierung durch Migration

Migration ist das Transferieren einer Anwendung in eine neue Umgebung mit allen notwendigen Änderungen. Neben den Programmen (Systemprogramme und Anwenderprogramme) können auch die Hardware und die Architektur sowie die Entwicklungsumgebung von der Migration betroffen sein. Auf jeden Fall betroffen sind die Anwenderdaten und die Benutzerschnittstellen. Bei der Legacy-Migration wird eine Alt-Anwendung auf eine neue Anwendungssoftware umgestellt. War ein solches Migrationsprojekt (Porting) früher zwingend notwendig mit einer Neuprogrammierung der Anwendungscode verbunden, stehen mittlerweile automatisierte Werkzeuge zur Verfügung.

Die Durchführung von Migrationen kann nach verschiedenen Strategien ablaufen, die jeweils unterschiedliche Risiken oder auch Chancen hinsichtlich des Gesamtprojektrisikos oder des Schnittstellenaufwands bergen:

- Big-Bang-Strategien: Big-Bang mit vollständiger Ablösung des Alt-Systems oder lokaler Big-Bang. Bei

Letzterem entwickeln Unternehmen mit dezentraler Organisation zunächst ein zentrales Mastersystem, anschließend erfolgt ein Roll-out sukzessiv als lokaler Big-Bang.

- Sukzessiv-Strategien, die ein auf Sicherheit bedachtes schrittweises Einführen vorsehen: Dies kann zum einen die sukzessive Einführung neuer Funktionen oder neuer Organisationseinheiten sein oder die allmähliche Einführung für komplette Prozessketten.

Migrationen erfordern ein gutes Verständnis des Alt-Systems, da sie durch versteckte Abhängigkeiten zu anderen Programmen nicht vorhersehbare Effekte auslösen können. Argumente, die für eine Migration sprechen, sind die Erhaltung des in die Legacy-Systeme integrierten Business-Know-hows sowie die im Laufe der Jahre in die bestehenden Systeme investierten Mittel. Zudem kann – im Gegensatz zu einer Neuentwicklung – von einem System mit einer bekannten Menge an Codes ausgegangen werden, so dass für Migrationen zuverlässigere Aufwandsschätzungen möglich sind.

Basissysteme	Architektur und System	Produktive Anwendungen
Hardware: Umstellung von veralteter Hardware	IT-Architektur: Anpassung der zugrundeliegenden Architektur	Anwendungsprogramme: Austausch (eines Teils) der Software
Entwicklungsumgebung: Ersetzen der Entwicklungsumgebung	Systemsoftware: Austausch (eines Teils) der Software	Anwendungsdaten: Transfer von Dateien in ein anderes System
		Benutzerschnittstellen

Abbildung 6: Interdependente Aktionsfelder in Migrationsprojekten

Vorteile	Nachteile
Geringes Investitionsrisiko	Abhängigkeit vom Servicegeber
Transparente IT-Kosten	Langsamere Datenübertragungsgeschwindigkeit
Beschleunigte Implementierung	Geringere Anpassungsmöglichkeiten
Verringerung der IT-Prozesskomplexität	Geringere Daten- und Transaktionssicherheit
Mobilität	Rechtliche Probleme bei grenzüberschreitenden Datenhaltungen
Konzentration auf das Kerngeschäft	

Abbildung 7: Ausgewählte Vor- und Nachteile einer Auslagerung als SaaS

### ABLÖSUNG VON ALT-SOFTWARE DURCH STANDARDSOFTWARE

Die Modernisierung durch Eigenentwicklungen und Weiterentwicklungen ermöglicht Unternehmen eine bessere Kontrolle und auch eine stärkere Individualisierung der benötigten Software. Auf der anderen Seite bieten (vor allem Standard-)Lösungen externer Anbieter eine Verringerung der Komplexität und des Risikos.

Der Einsatz von Standardlösungen, zum Beispiel für Kernbankenanwendungen, ist sinnvoll, wenn sie möglichst viele Geschäftsprozesse des Unternehmens mit einigen Anpassungen abdecken (Customizing). Standardsoftware von externen Herstellern kostet Lizenzgebühren und verursacht in der Regel hohen Customizing-Aufwand, da eben viele Benutzer auf noch mehr liebgegewonnene Funktionen ihrer alten Anwendungen nicht verzichten möchten. Im Gegenzug für seine Lizenzgebühren gewährleistet der Hersteller durch Aktualisierung und neue Releases die Aktualität der Standard-Geschäftsprozesse (zum Beispiel Buchhaltung, Personalverwaltung etc.).

Die Ablösung von Alt-Anwendungen durch moderne Standardsoftware mit ausreichendem Funktionsumfang ermöglicht einen relativ raschen Übergang auf neue, funktionsfähige Systeme. Bei dem Übergang zu Standardlösungen kann es zu einer Koexistenz von Alt-Anwendungen und neuen Anwendungen kommen. In den meisten Fällen jedoch wird eine vollständige Ablö-

sung der Alt-Software angestrebt – aus Effizienzgründen.

Allerdings ist bei größeren Projekten der Aufwand nicht zu unterschätzen. Unternehmen mit großen Fallzahlen und komplexen Anbindungen an die umgebende Software haben auch bei Standardanwendungen einen erheblichen Anpassungsaufwand. Ein Beispiel ist das Transformationsprogramm „Magellan“ der Deutschen Bank, das 2011 startete und bis Ende 2015 alle Alt-Anwendungen für Sparen, Baufinanzierung, Inlands- und EU-Zahlungsverkehr auf Standardsoftware umgestellt haben soll. Die Bank gibt rund eine Milliarde Euro für die SAP-Einführung aus.

Magellan umfasst die gesamte IT und alle Abwicklungsprozesse des Geschäftsbereichs Privat- und Geschäftskunden in Deutschland. Seit Juli 2012 werden mehr als fünf Millionen Sparkonten der Deutschen Bank auf der neuen Hochleistungsplattform geführt. Die Plattform bietet das künftige gemeinsame Fundament für die Filialen von Deutscher Bank und der 2012 übernommenen Postbank.

### OUTSOURCING: STATE-OF-THE-ART-LÖSUNGEN SAAS/CLOUD/WEBBASIERT

Software und IT-Infrastruktur können auch von einem externen IT-Dienstleister betrieben und vom Kunden als Service genutzt werden. „Software as a Service“ (SaaS) ist ein Teilbereich des Cloud Computings. Der Servicegeber übernimmt die komplette IT-



Administration und Dienstleistungen wie Wartung und Updates. Die IT-Infrastruktur und alle IT-administrativen Aufgaben werden an den Dienstleister ausgelagert. Für die Nutzung und den Betrieb zahlt der Servicenehmer eine nutzungsabhängige Gebühr und spart die Anschaffungs- und einen Teil der Betriebskosten.

### SUMMARISCHE BEURTEILUNG VON METHODEN DER SOFTWARE-MODERNISIERUNG

Die zur Verfügung stehenden Modernisierungstechniken lassen sich für ein Unternehmen grob bewerten. Die Parameter dafür reichen von den Kosten bis hin zur Analyse der Abhängigkeiten von Alt-Systemen. Die summarische Bewertung ersetzt keine Due Diligence im Einzelfall – gibt aber Orientierung für die strategische Ausrichtung einer Software-Modernisierung.

	Re-Hosting	Re-Engineering	Kapselung/ Wrapping	Migration auf neue Anwendung	Übergang auf Standard- Software	Outsourcing
<b>Kosten</b>	Niedrig	Hoch	Moderat	Fallabhängig	Hoch	Fallabhängig
<b>Aufwand und Zeitbedarf</b>	Niedrig bis Moderat	Hoch	Moderat	Moderat - Hoch	Hoch	Moderat
<b>Wiederverwendbarkeit der Investitionen</b>	Hoch	Niedrig	Moderat- Hoch	Wenig	Fast keine	Keine
<b>Risiko</b>	Niedrig	Hoch	Moderat	Mittel	Hoch	Moderat
<b>Nutzer-Erfahrung</b>	Wie bisher	Neu und besser	Neu	Neu	Neu	Neu
<b>Funktionalitäten</b>	Wie bisher	Wie bisher und besser	Wie bisher	Wie bisher und besser	Weniger oder wie bisher	Wie bisher und besser
<b>Agilität</b>	Wie bisher	Hoch	Niedrig	Höher	Moderat	
<b>Auswirkung auf Geschäftsabläufe</b>	Keine	Ja	Keine	Ja	Ja	Ja
<b>Abhängigkeit von Altssystemen</b>	Keine	Keine	Eingeschränkt	Während Übergang	Keine	Während Übergang

Abbildung 8: Vergleich von Ergebnissen verschiedener Software-Modernisierungsmethoden, Quelle: Ergänzt nach Infosys, Legacy Modernization

## Exkurs: Software-Re-Engineering

### WAS IST SOFTWARE-RE-ENGINEERING? EINE KURZBESCHREIBUNG

Software-Re-Engineering, manchmal auch IT-Sanierung genannt, ist eine Methode zur nachhaltigen Modernisierung von Alt-Anwendungen durch eine grundlegende Überarbeitung.

Gerade unternehmenskritische Anwendungen, die aufgrund des kodifizierten Know-hows nur schwer durch Neuentwicklungen oder Standardprodukte zu ersetzen sind, können durch Re-Engineering wieder zukunftsfähig werden.

Re-Engineering als Methode ist der Neuentwicklung von aktuell eingesetzter Software in bestimmten Fällen überlegen:

- Es ist die bessere Alternative, wenn die Alt-Software auf grundsätzlich zukunftsfähigen Technologien basiert oder durch Updates zukunftsfähig werden kann. Denn die Software kann auch während der Sanierung weiter verwendet und sogar um wichtige neue Features ergänzt werden.
- Ein weiterer typischer Anwendungsfall für Re-Engineering liegt vor, wenn die Daten, die Datenschemata und die Schnittstellen zu benachbarten Systemen komplex und nicht ausreichend oder gar nicht dokumentiert sind. Die Neuentwicklung der Software oder auch der Kauf einer Standardsoftware sind in solchen Fällen keine einfachen Lösungen, während die Alt-Anwendung diese Komplexitäten im Grundsatz beherrscht.

Der Zustand der Alt-Anwendung bestimmt dabei die Vorgehensweise des Re-Engineerings.

### VORGEHENSWEISE BEIM SOFTWARE-RE-ENGINEERING

#### **Analyse der Funktionsweise der Alt-Software:**

Zuerst muss die Funktionsweise der Alt-Anwendung verstanden werden. Eine vollständige und aktuelle Dokumentation erleichtert dies; aber bei Alt-Anwendungen ist die Dokumentation oft unvollständig, veraltet oder schlicht nicht vorhanden und die Entwickler sind in der Regel schon längst nicht mehr verfügbar.

#### **„Reverse Engineering“ zum Verstehen der Funktionalitäten der Alt-Anwendung:**

Ein großer Teil des Re-Engineerings besteht daher im sogenannten „Reverse Engineering“, um den Aufbau und die Funktionalitäten der Alt-Anwendung zu verstehen: Aus der bestehenden Implementierung werden das Design und die Anforderungen herausgearbeitet. Dies kann vor oder während der Stabilisierung der Anwendung erfolgen.

#### **Stabilisierung der Alt-Anwendung durch automatisierte Tests:**

Alt-Anwendungen sind typischerweise nicht oder nur rudimentär automatisiert getestet. Diese Tests, die idealerweise nach jeder Codeänderung automatisch ausgeführt werden, sind wichtig, um auszuschließen, dass Änderungen an einer Stelle des Programmcodes Fehler an anderen bereits getesteten Stellen zur Folge haben (Regression).

#### **Paralleler (Wieder-)Aufbau einer konsistenten Dokumentation:**

Alle Informationen über Software und „Technical Debt“, die man aus dem Programmcode der Alt-Software gewinnt, werden dokumentiert. In manchen Fällen lassen sich Teile der Dokumentation automatisch er-



stellen, zum Beispiel Ablaufdiagramme von wichtigen Funktionalitäten.

#### **Balance zwischen Stabilisierung und neuen Features:**

Während der Analyse und Stabilisierung kann die Alt-Anwendung weiterhin produktiv eingesetzt werden. Die Entwicklung neuer Features allerdings muss warten, bis die Anwendung wieder stabil ist. Gerade am Anfang ist es wichtig, dass das Entwicklungsteam sich ganz auf die Stabilisierung der Software konzentriert. Je weiter die Stabilisierung und Modernisierung fortschreiten, desto mehr neue Features können eingeführt werden. Dies ist dann meist auch der beste Zeitpunkt, um Modernisierungen an Bibliotheken und anderer verwendeter Software vorzunehmen.

#### **RE-ENGINEERING DURCH AGILE SOFTWAREENTWICKLUNG**

Empfehlenswert ist, agile Methoden für das Software-Re-Engineering einzusetzen. Re-Engineering-Projekte profitieren besonders von agilen Methoden, da sie sich typischerweise ständig ändern und schwer abschätzbar sind. Bei der agilen Softwareentwicklung wird in kurzen Zyklen entwickelt, in der Regel dauert einer dieser Zyklen ein bis zwei Wochen, in denen alle Phasen der

Softwareentwicklung durchlaufen werden. Große Aufgaben werden in kleine, übersichtliche Pakete zerlegt – und zu erledigende Aufgaben werden bewusst nur über kurze, überschaubare Zeiträume geplant.

Ein weiteres Prinzip von agilen Entwicklungsmethoden ist die eigenverantwortliche Arbeit der Entwickler, die sich ihre Aufgabenpakete aus den gesamten für den Zyklus geplanten Aufgaben selbst aussuchen.

Gängige Praxis der agilen Softwareentwicklung ist zudem das Vier-Augen-Prinzip für alle Code-Änderungen, die in Produktion gehen sollen: Entweder in Form des „Pair Programming“, also der gemeinsame Entwicklung durch zwei Entwickler an einem Rechner, oder als „Code Review“, der Überprüfung des Codes durch einen zweiten Entwickler. Durch diese Methoden wird das Know-how im ganzen Team verbreitet.

Elementar für die agile Softwareentwicklung ist eine gute und konstruktive Kommunikation im Team. Zum einen sollen Fortschritte und Probleme kommuniziert werden, zum anderen ist der Austausch über Prozesse und mögliche Verbesserungen wichtig. So reagiert das Team dynamisch auf Herausforderungen und Eigenheiten des Projekts.

## Aspekte der Durchführung einer Modernisierung

### PLANUNG: KONZEPTIONELLE FRAGEN EINES MODERNISIERUNGSPROJEKTS

Vor einer Modernisierung von Alt-Anwendungen muss – nach Festlegung auf eine bestimmte Art der Vorgehensweise – eine Reihe von Fragen beantwortet werden:

- Grundsätzlich und am Wichtigsten: Sind die bestehenden Anwenderdaten sicher, wie können sie gesichert bleiben und wie sind sie während des Umstellungsprozesses zugänglich?
- Falls eine neue Anwendung zum Einsatz kommen soll, welches ist die beste Alternative (für eine Migration)?
- Wie lange wird es dauern, bis die neuen Anwendungen reibungslos laufen?
- Wie viel Zeit wird es das eigene IT-Personal und die eigene Fachabteilung kosten, die Umstellung durchzuführen bzw. sich an die Veränderungen anzupassen?
- Welche Notfallpläne sind erforderlich, falls sich die Modernisierungen nicht so wie geplant entwickeln?

Klassischerweise gliedert sich die Vorbereitung und Durchführung einer Software-Modernisierung in typische Projektmanagement-Phasen. Am Beispiel der Modernisierung in Form einer Migration werden die

Phasen Vorstudie, Konzept und Design, Migration und Abschluss durchlaufen (Abbildung 9).

Zur Planung eines Modernisierungsprojekts bietet sich die Gliederung in zwei Phasen an: erstens die Beurteilungsphase, zweitens die Durchführungsphase.

In der Beurteilungsphase wird zunächst die Effektivität der Unternehmens-IT beurteilt. Hier geht es um Applikationen, um Daten, um die Infrastruktur und die Abläufe der IT. Dann wird eine Solldefinition der neuen Architektur entworfen: Hieraus abgeleitet wird die Modernisierungsstrategie in einem Business Case unter Berücksichtigung der voraussichtlichen Kosten (Total Cost of Ownership – TCO).

Auf dieser Grundlage wird dann der Durchführungsplan entwickelt. In einem umfassenden Projekt können drei Arbeitsgebiete unterschieden werden:

- Die Modernisierung der Applikationen (durch zum Beispiel Migration, Re-Engineering oder Re-Hosting) und der Datenbanken (Migrationen und Konsolidierung)
- Die Modernisierung der Infrastruktur (Kapazität, Infrastruktur, Auslegung)
- Die Modernisierung des Betriebs (IT-Sicherheit, Monitoring, Ressourcen Zuordnung)



Vorstudie	Konzept und Design	Durchführung	Abschluss
Legacy-Analyse	Zieldesign	Projektmanagement	Übergabe
Anforderungsanalyse	Zielanwendungen	Konfigurations- Management	Mitarbeiter-Qualifizierung
Analyse Migrationsumgebung	Migrationsstrategie	Änderungsmanagement (ITIL Change Management)	
	Projektstrategie	Migration (Systeme, Anwendungen, Daten)	
		Tests	
		Betriebswirtschaftliches Change Management	

Abbildung 9: Phasen einer Software-Modernisierung (Beispiel Migration)

## DIE FLANKIERUNG DER MODERNISIERUNG: CHANGE MANAGEMENT

### Change Management in IT- Modernisierungsprojekten

Change Management im engeren Sinne der IT ist ein Thema der IT Infrastructure Library (ITIL) und wird dort als Prozess definiert, der alle Anpassungen an der IT-Infrastruktur aufnimmt (Requests for Change (RfC)), kontrolliert, effizient und unter Minimierung von Risiken für den Betrieb bestehender Business Services durchführt. Wir verstehen hier Change Management (Veränderungsmanagement) betriebswirtschaftlich: als Steuerung der Verhaltensänderung von Mitarbeitern und Anwendern.

Unter Veränderungsmanagement lassen sich dann alle Aufgaben, Maßnahmen und Tätigkeiten zusammenfassen, die eine bereichsübergreifende und weitreichende Veränderung einer Organisation bewirken sollen. Auch IT- oder Technologiemodernisierungen erfordern möglicherweise Veränderungen der Aufbau- und Ablauforganisation, sicher aber die Auseinandersetzung mit neuen Applikationen und die Einarbeitung in sie.

Die Grundeinstellung bei Veränderungen im Unternehmen ist bei allen Betroffenen unterschiedlich: Daher reichen die üblichen Reaktionen von sehr positiv über neutral bis sehr negativ. Das Durchlaufen dieser

einzelnen Stadien während eines Veränderungsprozesses und der allmähliche Abbau der Widerstände und die Zunahme der Akzeptanz werden wesentlich von einer professionellen Change Communication unterstützt. Veränderungsmanagement soll die Durchführung der notwendigen Veränderungen managen und Change Communication soll die Akzeptanz verbessern, indem Ablehnung durch überzeugende Kommunikation in Zustimmung verwandelt wird.

### Mitarbeiter migrieren mit: Vier Schwerpunkte für das begleitende Change Management

Durch Software-Modernisierungen können innerhalb der für den Anwendungsbetrieb verantwortlichen IT-Einheiten und bei den Anwendern unterschiedliche Veränderungen eintreten (Abbildung 10).

Veränderungen sehen viele Menschen eher als Risiko, weniger als Chance. Durch eine pro-aktive Informationspolitik und rechtzeitige Kommunikation, die den Mitarbeitern die Veränderung, ihre Gründe und Notwendigkeit erklärt, können Gerüchte und unbegründete Ängste vermieden werden. Je nach Art und Umfang der Veränderungen sind Maßnahmen erforderlich, die die Mitarbeiter auf die Veränderung vorbereiten. Gutes Veränderungsmanagement durch die Abteilungen IT und HR kann die Akzeptanz der Modernisierung nachhaltig absichern.





Typische Veränderungsbereiche	IT-Support und Anwendungsbetrieb	Anwender
Neue Technologien, für die derzeit kein ausreichendes Wissen vorhanden ist	✓	
Veränderte Betriebsprozesse und Verantwortlichkeiten	✓	✓
Wegfall von bisherigen Aufgaben und Kompetenzgebieten	✓	
Neue Aufgaben und Organisationsstrukturen	✓	✓
Austausch von Ansprechpartnern sowie internen/externen Dienstleistern	✓	✓
Ersatz von beherrschten durch unbekannte Anwendungen	✓	✓

Abbildung 10: Typische Veränderungsbereiche bei Software-Modernisierungen

Bei der Einführung neuer IT-Systeme wie auch der Ablösung bestehender Systeme ist es Aufgabe des Change Managements, die Reibungsverluste zu minimieren. Change Management kann zur erfolgreichen Einführung neuer IT-Systeme vier wesentliche Beiträge leisten:

- Erstens: Projektziele, Anforderungen und den Umfang der anstehenden Veränderungen sauber kommunizieren.
- Zweitens: Die späteren Anwender und ihre Linienvorgesetzten realistisch auf die Veränderung vorbereiten – mit allgemeinen Informationen, mit spezifischeren Antworten und schließlich mit der Anwenderschulung und dem Support.
- Drittens: Einführung von Change Management als Schnittstelle zum Betriebsrat und zum Datenschutzbeauftragten, die bei Veränderungen von IT-Systemen mitsprechen.
- Viertens: Planung der projekt- und unternehmensinternen Kommunikation.

### Kommunikation mit unterschiedlichen Stakeholder-Gruppen

Schnittstelle zu Betriebsrat und Datenschutzbeauftragtem

Der Betriebsrat hat nach dem Betriebsverfassungsgesetz bei der Einführung von neuen und der Ablösung alter IT-Systeme in der Regel ein Mitspracherecht.

Gespräche mit dem Betriebsrat sollten spätestens dann erfolgen, wenn das Lastenheft für das neue System vorliegt. Ähnlich liegt der Fall bei der Kommunikation mit dem betrieblichen Datenschutzbeauftragten.

Kommunikation innerhalb des Modernisierungsprojekts

Bereits mittelgroße IT-Projekte in Großunternehmen können die Mitarbeiteranzahl und das Budgetvolumen eines mittelständischen Unternehmens erreichen: Mitarbeiter arbeiten Monate oder Jahre an einem solchen Projekt. Informations- und Kommunikationsprobleme in solchen Teams aus verschiedenen eigenen Fachabteilungen, eigenen IT-Fachleuten und externen IT-Spezialisten, jeweils mit eigenen Interessenlagen und Sichtweisen, können substantiell sein und den Projektablauf beeinflussen. Change Management kann und muss hier Orientierung und Richtung geben.

Veränderungsmanagement in der Anwenderorganisation

Um abschätzen zu können, wie die Anwender auf die Veränderung vorbereitet werden müssen, ist zu prüfen, wo sich wesentliche Veränderungen ergeben: in den Arbeitsabläufen oder Prozessen, hinsichtlich der Ansprechpartner oder Organisationseinheiten oder bei der Bedienung der Anwendung? Hat die Software-Modernisierung Auswirkungen auf die Anwender, werden üblicherweise Anwendervertreter einbezogen.



Auf Veränderungen der Arbeitsabläufe oder Verantwortlichkeiten, auf wegfallende oder zusätzliche Aufgaben müssen die Anwender vorbereitet werden. Durch Einführungs- und Schulungskonzepte werden die fachlichen und organisatorischen Voraussetzungen geschaffen.

#### **Einführungs- und Schulungskonzept für Anwender**

Kriterien für ein Einführungs- und Schulungskonzept für Anwender ergeben sich aus den folgenden Punkten:

- Kenntnisse und Erfahrung der Anwender im grundsätzlichen Umgang mit IT-Systemen?
- Anzahl und Standorte der betroffenen Anwender?
- Einfluss der Anwendungen für den Geschäftserfolg?
- Einfluss der Änderungen auf die Arbeitsabläufe der Anwender?
- Zeitpunkt der Veränderung?
- (Doppel-)Belastung der Anwender während der Software-Modernisierung?

Diese Kriterien bestimmen auch das Format des Konzepts: Sollen Präsenzschulungen erfolgen oder soll lediglich Schulungsmaterial bereitgestellt werden? Ist eine Schulungsumgebung zur Einübung der neuen Anwendungen außerhalb des Produktionsbetriebs notwendig? Sollen alle Nutzer geschult werden oder reichen Multiplikatoren zur Weitervermittlung des Wissens?

Oft werden die Formate in Mischformen zu passenden Schulungs- und Einführungskonzepten für Anwender zusammengestellt.

#### **Kommunikationsinstrumente des Change Managements in Migrationsprojekten**

Change Communication kann einen erheblichen Beitrag zum Gelingen eines Modernisierungsprojekts leisten. Bewährte Instrumente zur Motivation insbesondere des Projektteams sind beispielsweise:

- Regelmäßige persönliche Informationsveranstaltungen zum Beginn eines Modernisierungsprojekts und jeweils im direkten Anschluss an Lenkungsausschusssitzungen.
- E-Mail-Newsletter für regelmäßige aktuelle und auch Ad-hoc-Informationen zu Projektfortschritten und aktuellen Fragen.
- Gelegentliche Workshops, um Zwischenbilanzen zu ziehen und die Projektausrichtung zu bestätigen.
- „Celebrations“ von wesentlichen Projektfortschritten für das Projekt- und Umsetzungsteam.

Zukünftige Anwender müssen informiert und geschult werden. Einige Instrumente, die hier im Rahmen eines zu erarbeitenden Schulungskonzepts und -plans zum Einsatz kommen, sind:

- Schulungsmaterial, zugeschnitten auf unterschiedliche Benutzerkreise und -vorkenntnisse, zum Beispiel Administratoren, Trainer, Anwender.
- Erarbeitung von Anwenderdokumentationen mit hohem Beispielanteil.
- Schulungen der Anwender (Einzel- oder Gruppenschulungen, Online-Schulungen, Webinare).
- Engagement professioneller externer Trainer.
- Einsatz von Mitarbeitern aus den betroffenen Bereichen als Multiplikatoren (Train-the-Trainer-Konzepte) zur Unterstützung der System Einführung.
- Einrichtung einer Support-Hotline.
- Bereitstellung einer Online-Hilfe, auch als Nachschlagewerk für einzelne Funktionen.
- Probetrieb zur Vertiefung durch praktische Übung.

## Zur Notwendigkeit einer IT-Modernisierung

### LEITFRAGEN UND ANALYSETECHNIKEN

Vor allen Überlegungen zur Art und Weise einer Software-Modernisierung aber muss festgestellt werden, ob sie überhaupt notwendig ist. Als sinnvoll für eine grundlegende Entscheidung hat sich eine Gegenüberstellung des Werts bzw. der Bedeutung der IT-Anwendungen für das operative Geschäft mit der technischen Qualität der zur Verfügung stehenden Anwendungen erwiesen. Sie ermöglichen die schnelle Klassifizierung der Dringlichkeit von Modernisierungen.

Eine Beurteilung der Notwendigkeit einer Ablösung von Alt-Software muss die dabei infrage stehende Anwendung von mehreren Blickrichtungen aus betrachten: von der technischen Perspektive über den Geschäftswert bis zur organisatorischen Einbindung. Die Beantwortung der folgenden (typisierten) Fragen kann Hinweise auf die geeignete Modernisierungsstrategie geben:

- Ist die Anwendung geschäftskritisch? Nein? Dann bedarf es keiner Modernisierungsstrategie für diese Anwendung.
- Was sind die Geschäftsziele des Unternehmens? Je nach Ausrichtung des Geschäfts ergeben sich bestimmte Anforderungen an die Modernisierung von Alt-Anwendungen.
- Unter welchen Vorbedingungen und Voraussetzungen kann eine Weiterentwicklung der Anwendung stattfinden? Das Alt-System mag diese Bedingungen erfüllen – oder auch nicht.
- Was ist die voraussichtliche Lebensdauer der Anwendung? Die Lebensdauer der Alt-Anwendungen hängt auch ab von der zugrundeliegenden Hardware, die das Alt-System unterstützt.
- Wie lang ist die wünschenswerte Lebensdauer der Anwendung? Wird die Anwendung nur für eine überschaubare Periode eingesetzt, lohnt eine Mo-

dernisierung nicht. Ist sie dagegen wesentlicher Bestandteil der Geschäftsgrundlage, sind Investitionen gerechtfertigt.

- Wie ist der technische Status der Anwendung? Veralterte Software ist schwierig zu analysieren und nur mit hohem Aufwand zu modernisieren.
- Ist das Unternehmen, das das System einsetzt, offen für Veränderungen? Die Veränderungsbereitschaft ist ein Erfolgsfaktor für Modernisierungsprojekte.
- Stehen ausreichend Mittel für die Modernisierung der Anwendung im Unternehmen bereit? Faktoren wie der technische Reifegrad, das Vorhandensein von Fertigkeiten und Know-how der Mitarbeiter beeinflussen Modernisierungsprojekte wesentlich.

Verschiedene Analysemethoden helfen dabei, diese Fragen zu beantworten. Eingesetzt werden Methoden vom Benchmarking bis zur Risikoanalyse (Abbildung 11).

Praxisbezogenen wird man detailliertere Fragen hinsichtlich der Performance von Alt-Anwendungen stellen müssen. Unterschiedliche Organisationen haben detaillierte Leitfäden zur Beurteilung von Modernisierungserfordernissen und Migrationserfordernissen entwickelt. Einen praxisbezogenen Fragebogen bot das Fachmagazin „die bank“ für die Beurteilung von Weiterentwicklung oder Ablösung von Legacy im Rahmen von Bankenfusionen (Abbildung 12).



**SOFTWARE-MODERISIERUNG: ZWANG, PFLICHT ODER STRATEGISCHES INSTRUMENT?**

Schnelle technologische Entwicklungen erfordern entsprechende Investitionen. Geschäftliche Anforderungen treiben die Informationstechnologie. In wettbewerbsintensiven Umfeldern müssen neue Funktionalitäten, Produkte und Dienstleistungen schnell auf den Markt gebracht werden. Kunden wollen mit jeder Technologie, auf jedem Vertriebsweg und an jedem Ort Produkte und Dienstleistungen kaufen sowie in Echtzeit Informationen erhalten.

Aber nicht nur betriebswirtschaftliche und technische Faktoren treiben den CIO an. Steigende Anforderungen, um neue gesetzliche Vorgaben zu erfüllen, und

Berichtspflichten erhöhen die Komplexität der Anwendungslandschaften. Gleichzeitig soll die IT „industrialisierter“ arbeiten, sie soll das Geschäft und gewissermaßen sich selbst standardisieren und automatisieren. Außerdem sind die Mittel knapp, Sparmaßnahmen und Kapitaleinschränkungen reduzieren den Handlungsspielraum.

Software-Modernisierung kann Zwang bedeuten, wenn Alt-Anwendungen nicht mehr ausreichen. Sie kann Pflicht sein, wenn CIOs vorausschauen. Sie kann auch Instrument des CIO sein, wenn für die Zukunft mehr Budgetmittel für strategische IT-Investitionen statt für Maintenance frei werden.

<b>Analysemethode</b>	<b>Kurzbeschreibung</b>
Benchmark-Methode	Bewertung des gesamten Anwendungsportfolios im Vergleich mit Industriestandards
Finanzanalyse	Analyse der Kostenstruktur der Anwendung hinsichtlich der Investitions- und der Betriebskosten
Redundanzanalyse	Analyse von Redundanzen der Applikation im Einsatz des Unternehmens, zwischen verschiedenen Ländern und Geschäftseinheiten, Werken, Prozessen oder Funktionen
Ablösungsanalyse	Identifizierung von älteren Applikationen mit wenigen Nutzern und eingeschränkten Funktionalitäten
Rationalisierungsanalyse	Beurteilung des Schwierigkeitsgrads einer Modernisierung: Machbarkeitsanalyse für Erweiterungen, Restrukturierungen oder Konsolidierungen
Vergleichende Analyse	Analyse verschiedener Attribute und Dimensionen einer Applikation: zum Beispiel Stabilität, Kritikalität, Wertbeitrag zum Geschäft, Kosten
Architekturanalyse	Analyse von Übereinstimmung oder Nichtübereinstimmung der unterliegenden Applikationstechnologie mit der bevorzugten IT-Technologie im Unternehmen
Risiko-Analyse	Analyse der möglichen Risiken von Anwendungsausfällen, auslaufender Unterstützung durch Hersteller, Know-how-Verfügbarkeit und Anwendungsstabilität

Abbildung 11: Analysewerkzeuge der Bewertung von Legacy-Software, Quelle: Capgemini, Application Modernization and Retirement – Sustaining business innovation in the face of mounting IT complexity

**Weiterentwicklung oder Ablösung von Spezialsoftware und Eigenentwicklungen**

Funktionen	Stellenwert der Software	Einsatz in einem Nicht-Kerngebiet des Unternehmens?
	Umfang Software	Nicht alle benötigten Funktionen sind abgedeckt?
	Qualität der Software	Fachabteilung bemängelt Einsatzfähigkeit?
Flexibilität	Skalierbarkeit	Neue Daten verlangsamen das System?
	Erweiterungen	Erweiterungen schwierig? Ergänzung des Software-Codes notwendig?
	Schnelligkeit von Erweiterungen	Fachliche Änderungen erfordern langen Vorlauf?
	Tests	Automatisierte Testumgebung existiert nicht? Änderungen werden am Beispiel getestet?
Technologie	Architektur	Architekturprinzipien nicht bekannt oder dokumentiert?
	Schnittstellen	Anbindung an andere Systeme nur mit hohem Aufwand?
	Programmiersprache	Programmiersprache nur noch von wenigen Mitarbeitern beherrscht?
	Performance	Lange Antwortzeiten?
	Ausfallzeiten	10 % oder mehr?
IT-Personal	Personal und Fertigkeiten	Softwarebetreuung durch kleines Team älterer Kollegen?
	Wartung u. Weiterentwicklung Support	Fehlerbehebungen ausschließlich von älteren Kollegen?
	Support	Support auf Zuruf von nur wenigen Personen?
Effizienz	IT-Budget	Kostenstruktur für den Betrieb der Software unbekannt? Budget wird regelmäßig überschritten?
	Fachabteilung	Die Fachabteilung muss zusätzlich zur Software ergänzende Standardsoftware nutzen?

Abbildung 12: Beurteilung der Weiterentwicklung oder Ablösung von Spezialsoftware und Eigenentwicklungen, Nach: die bank 2.2010 für die Beurteilung im Rahmen von Bankenfusionen: Stringente IT-Strategie bei Bankenfusionen.



## Checkliste IT-Sanierung: Wo steht Ihr Unternehmen?

Den richtigen Zeitpunkt für eine Softwaremodernisierung zu erkennen, ist nicht immer einfach. Es spielen einfach zu viele, vor allem unternehmensinterne, Faktoren hinein. Anhand der folgenden zwei

Checklisten können IT-Verantwortliche den Status Quo ihrer Prozesse überprüfen und einen ersten Hinweis auf Handlungsbedarf bei der Softwaremodernisierung erhalten.

Frage:	Ist meine Individual-Software ein IT-Sanierungsfall?
<input type="checkbox"/>	Konnten Business-Innovationen aufgrund hoher Entwicklungskosten der Software nicht umgesetzt werden?
<input type="checkbox"/>	Gibt es häufig Ausfälle der Software mit Auswirkungen auf den Geschäftsprozess?
<input type="checkbox"/>	Wird die Software überwiegend manuell getestet?
<input type="checkbox"/>	Sind zum stabilen Betrieb der Software operative Workarounds notwendig (z.B. regelmäßige Neustarts)?
<input type="checkbox"/>	Brauchen neue Entwickler mehr als einen Monat Einarbeitungszeit, um produktiv an der Software entwickeln zu können?
<input type="checkbox"/>	Baut die Software auf End-of-Life-Technologien auf?
<input type="checkbox"/>	Gibt es für die Software keine vollständige Gesamtdokumentation?
<input type="checkbox"/>	Widerspricht die Software den Vorgaben der Enterprise-Architektur?
<input type="checkbox"/>	Werden technische Schulden in der Software nicht dokumentiert?
<input type="checkbox"/>	Sind von der Software gleichzeitig mehrere Software-Stände in Produktion?

Abbildung 13: Wenn Sie mindestens fünf Fragen mit „Ja“ beantworten, liegt ein Sanierungsbedarf vor, Quelle: TNG

Frage:	Gibt es Sanierungsbedarf in meinem System-Stack?
<input type="checkbox"/>	Gibt es in Ihrem System-Stack Anwendungen, die noch im Produktivbetrieb sind, obwohl sie schon abgelöst wurden?
<input type="checkbox"/>	Gibt es in Ihrem Unternehmen eine Schatten-IT, also Software die nicht aus IT-Budgets finanziert ist und nicht von der IT betrieben wird?
<input type="checkbox"/>	Gibt es in Ihrem System-Stack mehrere Anwendungen, die die gleiche Aufgabe erfüllen?
<input type="checkbox"/>	Haben Sie schon einmal überlegt, große Teile Ihres System-Stacks auf einmal auszutauschen?
<input type="checkbox"/>	War die SEPA-Einführung in Ihrem Unternehmen ein komplexes Projekt?

Abbildung 14: Wenn Sie mindestens drei Fragen mit ja beantworten, liegt ein Sanierungsbedarf vor. Quelle: TNG



## Interview mit Christoph Stock: IT-Sanierung kann klappen



Christoph Stock,  
Partner, TNG Technology  
Consulting

**LÜNENDONK:** Herr Stock, als Partner einer IT-Beratung werden Sie häufig mit Unternehmenssoftware konfrontiert, die sanierungsbedürftig ist. Was sind aus Ihrer Erfahrung die häufigsten Ursachen dafür, dass eine Anwendung für Ihre Kunden zum Sanierungsfall wird?

**STOCK:** Technische Schulden der IT sind hier der häufigste Grund. In der Softwareentwicklung erleben wir häufig einen enorm hohen Zeitdruck auf die Entwicklerteams. Oft muss das Entwicklerteam bestimmte dringende Features schneller abschließen als es mit einer sauberen Implementierung möglich wäre. Dann werden technische Schulden gemacht. Wenn die Entwickler nicht die Gelegenheit bekommen, diese Schulden durch Nachbesserung wieder „abzubezahlen“, ist schon der erste Schritt in den Sanierungsfall getan.

Die technischen Schulden werden immer größer, bis irgendwann der Punkt erreicht ist, an dem neue Features zu teuer werden und nicht mehr implementiert werden können.

Eine Software wird aber selten über Nacht zum Sanierungsfall, es ist ein schleichender Prozess, den man oft erst bemerkt, wenn es schon zu spät ist.

**LÜNENDONK:** Was können Sie Unternehmen raten, damit sie diesen Prozess schon früh erkennen?

**STOCK:** Wichtige Indikatoren sind steigende Aufwandsschätzungen für neue Features sowie häufige Probleme im Produktivbetrieb. Einen weiteren Hinweis gibt ein Blick auf den Issue-Tracker. Wenn sich die Anzahl der offenen Issues häuft und gerade komplexe Issues oft lange unbearbeitet bleiben oder als „Won't fix“ geschlossen werden, ist dies kein gutes Zeichen. Hier sollten die Verantwortlichen eingreifen.

Eine gute, vollständige Dokumentation sollte in einem gesunden Projekt genauso vorhanden sein wie eine Auflistung der technischen Schulden. Das Projektteam muss immer auch genügend Zeit bekommen, um bereits aufgehäuften technischen Schulden wieder abzuarbeiten. Merkt man, dass die Auflistung der technischen Schulden nicht mehr gepflegt wird oder dass die Pflege einer Gesamtdokumentation zugunsten einer Delta-Dokumentation pro Release aufgegeben wurde, sind das schon sehr klare Alarmzeichen.

**LÜNENDONK:** Gibt es weitere Anhaltspunkte?

**STOCK:** Des Weiteren sollten Best Practices bei der Softwareentwicklung eingesetzt werden, wie etwa automatisierte Builds, Tests und Continuous Integration. Ebenso sollte es jedem Entwickler leicht möglich sein, eine eigene Testumgebung aufzusetzen. Das Deployment der Software sollte umfassend und einfach, am besten automatisiert erfolgen.

Aufhorchen sollte man auch, wenn Fachbereiche oder Mitarbeiter der IT sagen „... das gefällt uns auch nicht,



das ist aber historisch so gewachsen ...“ Schließlich gibt es noch den Punkt, an dem man wirklich merkt, dass es so nicht mehr weitergehen kann: Wenn Releases nicht abgenommen werden, weil die Qualität nicht mehr stimmt und wichtige Features nicht implementiert werden konnten.

**LÜNENDONK:** Welche Wege gibt es, aus einer solchen Situation herauszukommen? Wann kann man die Software noch retten und wann muss ich einen Austausch planen?

**STOCK:** Das ist die Frage, die wir von unseren Kunden immer zuerst gestellt bekommen. Eine pauschale Antwort darauf gibt es leider nicht. Wer seine Software neu schreiben will, muss sich Fragen stellen wie: Kann ich die Anforderungen an das System umfänglich beschreiben? Kann ich längere Zeit auf Feature-Entwicklung verzichten? Möchte ich ein Projekt mit vielen sich ändernden Schnittstellen zu Nachbar-Applikationen haben? Traue ich mir eine gegebenenfalls komplexe Datenmigration zu? Gibt es eine Standardsoftware, zu der ich migrieren kann?

Wenn man die meisten Fragen nicht bejahen kann, bleibt eigentlich nur die Sanierung. Immerhin hat die Alt-Software den Vorteil, dass sie mit den beteiligten Nachbar-Applikationen, deren Schnittstellen und den Daten in der Datenbank trotz aller Datenkorruption umgehen kann.

**LÜNENDONK:** Wie sieht ein Sanierungsprozess in der Praxis aus? Was sind die wesentlichsten Schritte?

**STOCK:** In der Anfangsphase wird der Fokus auf Stabilisierungsmaßnahmen und das Beseitigen von drängenden Softwareproblemen gelegt. Neue Features müssen hinten anstehen und sollten in der ersten Zeit eine große Ausnahme sein. Die Automatisierung von Build- und Deploy-Prozessen und der Aufbau einer Continuous Integration läuft parallel zu dieser ersten Phase. Sind diese Grundlagen gelegt, werden automa-

tisierte Softwaretests geschrieben, um Schritt für Schritt die Testabdeckung zu erhöhen. Hierbei gehen wir typischerweise nach einem Top-down-Ansatz vor.

Zuerst werden Integrationstests geschrieben, die große Teile der Applikation inklusive der simulierten Schnittstellen abprüfen. Parallel dazu wird pro beseitigtem Softwarefehler auch immer ein entsprechender automatisierter Test geliefert, um Regression zu verhindern. Damit bekommt man schnell die Sicherheit, um auch große Sanierungsaufgaben anzugehen.

**LÜNENDONK:** Woran können IT-Verantwortliche erkennen, ob ein Sanierungsprojekt auf dem richtigen Weg ist?

**STOCK:** Kurioserweise ist ein Merkmal einer erfolgreich laufenden Sanierung, dass die Anzahl der gemeldeten Fehler steigt. Dies mag verwundern, aber man muss bedenken, dass gerade wenn die Sanierung sehr akut war, oft ganze Teile der Anwendung nicht funktionieren haben. Statt eines Fehlers „Workflow X funktioniert nicht“ hat man nun zehn Fehlermeldungen „In Workflow X tritt für Fall Y ein Fehler auf“.

Das sind Fehler, die auch vorher schon vorhanden waren, aber von anderen Fehlern verdeckt wurden. Mit fortschreitender Sanierung geht die Anzahl der Fehler dann aber schnell wieder zurück.

**LÜNENDONK:** Und wann würden Sie sagen, ist eine Softwaresanierung beendet?

**STOCK:** Je weiter die Sanierung fortschreitet, umso mehr können neue Features eingebaut werden. Wenn die Weiterentwicklung wieder problemlos möglich ist und die Stabilität im Betrieb wiederhergestellt wurde, kann die Software wieder im ausgesetzten, regulären Prozess weiterentwickelt werden. Aber auch dann sollte immer genug Zeit für das proaktive Angehen von technischen Schulden eingeplant werden, sonst





wird die Software nur allzu schnell wieder zu einem Sanierungsfall.

**LÜNENDONK:** Wie können IT-Verantwortliche überhaupt vermeiden, dass eine Softwareanwendung zum Sanierungsfall wird?

**STOCK:** Die Entwickler sollten genug Zeit haben, um technische Schulden wieder abzubauen. Zudem ist es sehr vorteilhaft, wenn man das Team in die Abschätzung von Projektplänen involviert. Gerade agile Methoden sind hier von großem Wert, da man den Entwicklern die Möglichkeit gibt, selbst festzulegen, wie viel Arbeit sie in einer Iteration schaffen.

Wir erleben, dass immer mehr große Unternehmen auf agile Entwicklungsmethoden umstellen. Das ist aus unserer Sicht ein großer Trend in der Branche.

Ein sehr interessantes Konzept, gerade für große IT-Landschaften, ist auch die Auftrennung von IT-Projekten in Core IT und Fast IT. Mit der Fast IT werden Ideen auf Businessstauglichkeit hin geprüft, bei der Architektur und den Prozessen im Betrieb werden aber keine großen Vorgaben gemacht. So können hier schnell und einfach Ergebnisse bewertet werden.

Wenn eine Idee nicht funktioniert, dann wird die entsprechende Software auch schnell wieder verworfen. Nur die erfolgreichen Projekte werden in die Core IT überführt, wo dann strengere Prozesse vorgeschrieben sind. So hat man seine technischen Schulden immer im Griff und kann Innovation weiter treiben, ohne die Gesamtstabilität zu gefährden.



## Agile Migration – ein neuer Ansatz!



Christoph Stock,  
Partner, TNG Technology  
Consulting

Auch wenn wir Re-Engineering in der Regel bevorzugen, gibt es doch Fälle, in denen andere Ansätze zur Software-Modernisierung sinnvoll sind. Zum Beispiel dann, wenn eine Standardsoftware existiert, die die bestehenden Bedürfnisse komplett abdeckt, oder wenn es im System-Stack bereits eine Anwendung gibt, die die erforderlichen Aufgaben wahrnimmt und lediglich angepasst werden soll, um eine Alt-Anwendung ersetzen zu können.

Bei Software-Modernisierungs-Projekten stellt sich oft die Frage, wie man die eigentliche Migration von der alten Anwendung auf die neue gestaltet. In sehr einfachen Fällen wird die alte Anwendung einfach durch die neue ersetzt und alles funktioniert. Aber das ist eher die Ausnahme.

In der Regel kommunizieren Softwareanwendungen untereinander. Da eine Systemlandschaft historisch wächst, werden zumeist nicht alle Komponenten gleichzeitig erneuert. Aufgrund des komplexen Ineinandergreifens der verschiedenen Anwendung ist eine teilweise Erneuerung mit hohen Kosten verbunden. Zudem ist eine solche Investition nicht nachhaltig, da die neu geschaffenen Schnittstellen überflüssig werden, wenn weitere Alt-Anwendungen abgelöst werden.

Die Schnittstellen zwischen einer alten und neuen Anwendung sind eigentlich nur dann die gleichen, wenn von einer früheren Version einer Standardsoftware auf eine neuere migriert wird. Zumeist ist es nicht sinnvoll, alte Schnittstellen für die neue Software zu übernehmen, da man auf neue Technologien oder Architekturen, die sich in der Zwischenzeit etabliert haben, setzen möchte.

Auch fachliche Anforderungen spielen eine Rolle, beispielsweise wenn neue Produkte unterstützt werden sollen oder zusätzliche Daten benötigt werden. Im Ergebnis sind neue und alte Anwendungen schlicht nicht kompatibel. Zudem ist es aufgrund des sehr großen Aufwands in den meisten Fällen nicht sinnvoll, eine Alt-Anwendung nach der anderen zu migrieren.

### Der große Wurf?

Eine andere Option bietet die Big-Bang-Migration. Man koordiniert die einzelnen Ablösungsprojekte und migriert alle Anwendungen auf einmal. Dadurch entsteht keine Schnittstellenproblematik zwischen alten und neuen Anwendungen. Vor allem aus diesem Grund wird die Big-Bang-Migration so oft bevorzugt.

Allerdings birgt dieser Ansatz viele Risiken. Zum einen erfordert die Koordination viel Arbeit und führt zu Reibungsverlusten. Die einzelnen Migrationsprojekte werden meist von verschiedenen Teams bearbeitet. Da alle Migrationen gleichzeitig erfolgen müssen, muss der Fortschritt der Teams synchronisiert werden.

Verzögert sich die Arbeit bei einem Projekt, werden auch die anderen Teams mit ihrer Migration warten müssen. Treten während oder nach der Migration schwerwiegende Probleme an nur einer Stelle auf, ist der einzige Weg, von den neuen Anwendungen wie-

der auf die alten zu schalten und nach erfolgreicher Fehlerbehebung erneut die Migration zu versuchen. Solche Migrationsprojekte sind deshalb sehr teuer, riskant und schwer durchzuführen.

Big-Bang-Migrationen werden deshalb häufig hinausgezögert oder schlichtweg gar nicht durchgeführt, weil niemand das mit ihnen verbundene Risiko auf sich nehmen möchte.

### Agile Migration als Alternative

Der Ansatz der agilen Migration setzt genau bei diesen Problemen an. Anstelle einer großen Big-Bang-Migration, wird jede Anwendung einzeln migriert. Damit Probleme mit den alten Schnittstellen vermieden werden, lagern wir sie in eine eigene Komponente, den Adapter, aus.

Der Vorteil ist, dass die neuen Anwendungen frei von Altlasten bleiben. Aus Sicht der Entwickler gibt es nur die anderen, neuen Anwendungen. Somit werden alle Probleme der Big-Bang-Migration von dem Adapter abgefangen. Da er gewissermaßen als „Überset-

zer“ zwischen den neuen und alten Anwendungen arbeitet, kann er Verzögerungen und Rollbacks abfangen.

Jede Anwendung kann nun unabhängig von den anderen geplant und umgesetzt werden, die Projekte sind entkoppelt. Nachdem alle Komponenten erfolgreich migriert wurden, kann der Adapter abgeschaltet werden und alle „Altlasten“ sind nun restlos aus der Systemlandschaft entfernt. Darin liegt der große Vorteil der Entscheidung, die alten Schnittstellen in eine eigene Komponente auszulagern. Würde das nicht geschehen, blieben diese „Altlasten“ in der neuen Anwendung bestehen und müssten erst aufwendig wieder zurückgebaut werden.

Vor einer Migration sollte man deshalb abwägen, welchen Ansatz man wählt, und sich für denjenigen entscheiden, der für die eigene Situation am sinnvollsten ist.

Die nachfolgende Tabelle stellt die agile Migration und die Big-Bang-Migration gegenüber:

	Big Bang Migration	Agile Migration
Anzahl der Migrationen	Eine einzige, komplexe Migration	Mehrere kleine, einfachere Migrationen
Umgang mit Verzögerungen	Verzögerungen stellen ein hohes Risiko für das Gesamtprojekt dar	Verzögerungen betreffen immer nur ein Teilprojekt
Umgang mit Fehlern bei der Migration	Rollback aller beteiligten Anwendungen notwendig	Rollback einzelner Anwendungen möglich
Koordination des Gesamtprojekts	Das Gesamtprojekt muss sehr sorgfältig koordiniert und gesteuert werden	Die Teilprojekte sind weitestgehend unabhängig voneinander

Abbildung 15: Vergleich der Varianten von Software-Migrationen



Anschaulich werden die Unterschiede in Abbildung 21 dargestellt.

Hier werden die gleichen Systeme auf beiden Wegen migriert. Im unteren Teil, der die Big-Bang-Migration zeigt, wird nur ein Schritt durchgeführt, der allerdings teuer und riskant ist. Im oberen Teil, der die agile Migration veranschaulicht, werden stattdessen viele kleine Schritte durchgeführt. So kann man mit geringerem Risiko ein System nach dem anderen migrieren und

dabei flexibel auf geänderte Anforderungen und Probleme eingehen.

Am Ende der Migration ist das Ergebnis in beiden Fällen gleich: eine neue Systemlandschaft, ohne Rückstände der Alt-Anwendungen. Die agile Migration, mit der wir und unsere Kunden sehr gute Erfahrungen gemacht haben, ist in den meisten Fällen eine lohnenswerte Alternative zur Big-Bang-Migration.

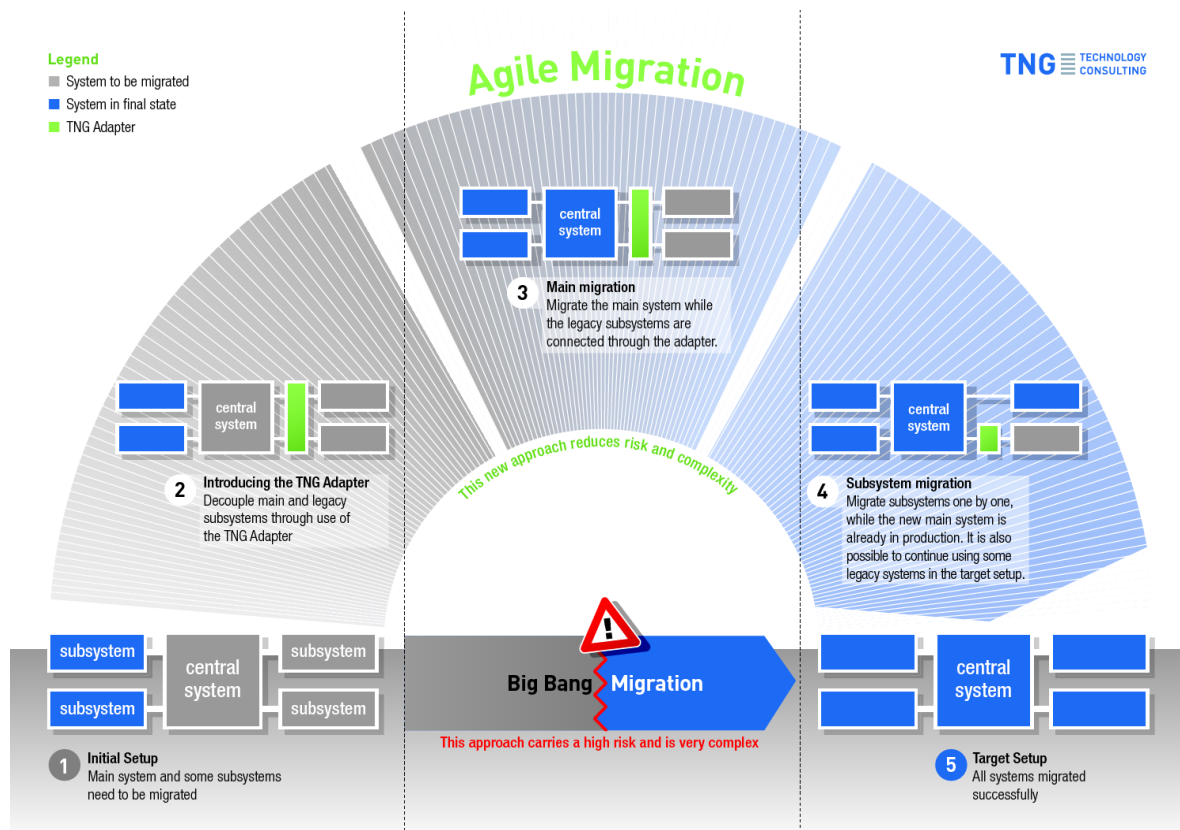


Abbildung 16: Unterschiede zwischen der agilen Migration und der Big-Bang-Migration auf einen Blick.



## TNG Technology Consulting

Die Firma TNG Technology Consulting wurde im Januar 2001 mit dem Ziel gegründet, eine möglichst optimale Firmenstruktur für kompetente Fachleute mit einer ausgesprochenen Vorliebe für Informationstechnik zu schaffen. Dazu wurden Erkenntnisse aus der Unternehmensforschung mit Eigenschaften der Vorbilder im Markt und eigene Vorstellungen der Gründer kombiniert. Das Ergebnis ist eine partnerschaftliche, wertgetriebene Unternehmensberatung mit Fokus auf High-End-Informationstechnik.

TNG hat sich ganz der technischen Innovation und deren Umsetzung für die Interessen der Kunden verschrieben. So konnten wir IT-Abteilungen großer Konzerne als Auftraggeber gewinnen und halten. Unsere Kunden kommen hauptsächlich aus der Telekommunikations- und Versicherungsbranche sowie dem E-Commerce, aber auch aus dem Automobilsektor, der Logistik, dem Finanzwesen und anderen Branchen.

Wir bieten unseren Kunden Know-how und Unterstützung in den Sparten Softwareentwicklung, Administration und Betrieb sowie IT-Management.

Dabei kümmern sich unsere über 170 Mitarbeiter sowohl um die Beratung des Kunden als auch um die Implementierung der Technik. Die Projekte werden möglichst in cross-funktionalen Teams bearbeitet und erstrecken sich über den gesamten Softwarelebenszyklus bis hin zum Design von hochverfügbaren Rechenzentren, Business Intelligence und dem Erstellen von Studien für das Management.

### KONTAKT

TNG Technology Consulting GmbH  
Christoph Stock, Partner  
Anschrift: Betastraße 13a, 85774 Unterföhring  
Telefon: +49 89 2158 996-0  
E-Mail: christoph.Stock@tngtech.com  
Internet: www.tngtech.com



## Lünendonk GmbH

Die Lünendonk GmbH, Gesellschaft für Information und Kommunikation (Kaufbeuren), untersucht und berät europaweit Unternehmen aus der Informationstechnik-, Beratungs- und Dienstleistungs-Branche. Mit dem Konzept Kompetenz<sup>3</sup> bietet Lünendonk unabhängige Marktforschung, Marktanalyse und Marktberatung aus einer Hand. Der Geschäftsbereich Marktanalysen betreut seit 1983 die als Marktbarometer geltenden Lünendonk®-Listen und -Studien sowie das gesamte Marktbeobachtungsprogramm.

Die Lünendonk®-Studien gehören als Teil des Leistungsportfolios der Lünendonk GmbH zum „Strategic Data Research“ (SDR). In Verbindung mit den Leistungen in den Portfolioelementen „Strategic Roadmap Requirements“ (SRR) und „Strategic Transformation Services“ (STS) ist Lünendonk in der Lage, ihre Beratungskunden von der Entwicklung der strategischen Fragen über die Gewinnung und Analyse der erforderlichen Informationen bis hin zur Aktivierung der Ergebnisse im operativen Tagesgeschäft zu unterstützen.

### KONTAKT

Lünendonk GmbH –  
Gesellschaft für Information und Kommunikation  
Mario Zillmann, Leiter Professional Services  
Anschrift: Ringweg 23, 87600 Kaufbeuren  
Telefon: +49 8341 96636-26 Telefax: +49 8341 96636-66  
E-Mail: [zillmann@lunenendonk.de](mailto:zillmann@lunenendonk.de)  
Internet: [www.lunenendonk.de](http://www.lunenendonk.de)



## ÜBER LÜNENDONK

Seit 1983 ist die Lünendonk GmbH spezialisiert auf systematische Marktforschung, Branchen- und Unternehmensanalysen sowie Marktberatung für Informations-technik-, Beratungs- und weitere hochqualifizierte Dienstleistungsunternehmen. Der Geschäftsbereich Marktforschung betreut die seit Jahrzehnten als Marktbarometer geltenden Lünendonk®-Listen und -Studien sowie das gesamte Marktbeobachtungsprogramm. Die Lünendonk®-Studien gehören als Teil des Leistungsportfolios der Lünendonk GmbH zum „Strategic Data Research“ (SDR). In Verbindung mit den Leistungen in den Portfolio-Elementen „Strategic Roadmap Requirements“ (SRR) und „Strategic Transformation Services“ (STS) ist die Lünendonk GmbH in der Lage, ihre Kunden von der Entwicklung strategischer Fragen über die Gewinnung und Analyse der erforderlichen Informationen bis hin zur Aktivierung der Ergebnisse im operativen Tagesgeschäft zu unterstützen.

Managementberatung

Informations- und  
Kommunikations-Technik

Wirtschaftsprüfung /  
Steuerberatung

Technologie-Beratung /  
Engineering Services

Zeitarbeit /  
Personaldienstleistungen

Facility Management /  
Industrieservice

LÜNENDONK GMBH  
Ringweg 23  
87600 Kaufbeuren  
Telefon: +49 8341 96636-0  
Telefax: +49 8341 96636-66  
E-Mail: [info@lunenendonk.de](mailto:info@lunenendonk.de)  
Internet: <http://www.lunenendonk.de>

Erfahren Sie mehr unter

<http://www.lunenendonk.de>

Copyright © 2014 Lünendonk GmbH, Kaufbeuren  
Alle Rechte vorbehalten

