

plenum 

**versicherungen**

die beharrlichkeit  
der kernsysteme



## die beharrlichkeit der kernsysteme

Welche Wege können bei der Ablösung des Kernsystems (Legacyssystem) beschritten werden? Wie reagieren Organisationen typischer, wenn die Entscheidung für die Einführung einer Standardplattform vorliegt? Welche Unterschiede ergeben sich effektiv zwischen der bestehenden Legacysoftware und einer aus der Veränderung einer Standardplattform resultierenden Anwendungssoftware? Wie kann „simultanes Design“ als Lösungsansatz für die erfolgreiche Einführung einer Standardplattform genutzt werden?

### einleitung

Die meisten mittelständischen Versicherer sind heute mit der Frage konfrontiert: „Was machen wir mit unserem Kernsystem?“

Diese Kernsysteme wurden meist noch im letzten Jahrtausend geschaffen und nach 2000 nur noch gewartet und partiell modifiziert. Aus technologischer Sicht handelt es sich dann um ein Legacysystem. Legacyssysteme sind eine spezielle Klasse von Anwendungssystemen die nur schwer zu verändern sind und die ein „gewisses“ Alter haben, daher stellt das Kernsystem der meisten mittelständischen Versicherer ein Legacysystem dar.

Jedes Anwendungssystem ist ein soziotechnisches System, welches applikative Softwareteile und soziale Elemente enthält, die notwendig sind, das Anwendungssystem einzusetzen und aufrecht zu erhalten. Folglich zählt zu jedem Anwendungssystem neben der eigentlichen Anwendungssoftware auch die Software zur Wartung und Weiterentwicklung, darunterliegende Strukturen wie Datenbanken und Archive oder Überwachungsmonitore. Auf der Seite der sozialen Elemente kommen Softwareentwickler, Vorgehensmodelle, Governance, Know-how, Programmierrichtlinien etc. hinzu. Auf der Nutzerseite wird zu diesem System neben dem eigentlichen User auch der Geschäftsprozess, der Informations- und Datenhaushalt gezählt.

Das Legacysystem ist ein Anwendungssystem, welches i.d.R. über Jahrzehnte hinweg in den jetzigen Zustand evolviert, d.h. die unterschiedlichen Bestandteile haben sich strukturell und begrifflich stark aufeinander abgestimmt. Die Abstimmung ist das Resultat diverser Weiterentwicklungszyklen. Hierbei werden immer mehr unternehmensspezifische Begrifflichkeiten, Prozess-, Kommunikations- und Entscheidungsstrukturen in das Legacysystem eingebracht und letztlich zu einem großen Teil in Code „gegossen“. Interessanterweise finden sich in der Struktur der Legacysoftware neben fachlichen Blöcken auch deutliche Spuren der sozialen Struktur des Entwicklungsteams. Die Isomorphie zwischen der Softwarearchitektur und den sozialen Strukturen des Unternehmens wird durch das Gesetz von Conway beschrieben.

### Conwaysches Gesetz

Organisationen sind im Entwurf von Systemen auf Modelle festgelegt, welche die Kommunikationsstruktur der Organisationen abbilden.

Diese sehr enge Verbundenheit der sozialen Strukturen mit dem über Jahre hinweg entwickelten System führt zur Entstehung eines unternehmensinternen Standards, der so weit führen kann, dass für das Legacysystem eine eigene Begriffswelt entsteht, die für Außenstehende nur schwer verständlich ist. Die langjährige Anwesenheit des Systems führt dazu, dass seine Existenz nicht in Frage gestellt wird und immer mehr Wissen über die internen Strukturen des sozioökonomischen Systems verloren geht. Das System ist am Ende für die Beteiligten nicht mehr objektiv beobachtbar, da sie selbst Teil des Systems geworden sind und jede Form der Selbstreflexion schwer ist. Hinzu kommt die schiere Größe solcher Systeme, welche es i.d.R. unmöglich macht, dass ein Einzelner sie vollständig verstehen kann. Im Gegenteil, das Wissen über das System wird immer fragmentierter und es entstehen Kopfmonopole; zum Teil entstehen diese Monopole auch in Fachbereichen, die schon sehr lange die Softwareveränderung begleiten. Genau diese Kopfmonopole spielen eine Schlüsselrolle bei einer Veränderung.

Auf Dauer stellt sich ein stabiles Gleichgewicht in dem Legacysystem ein. In diesem Gleichgewicht wird Veränderung, durch die Kopfmonopole kontrolliert und alle Veränderungen, die dieses Gleichgewicht bedrohen, zeigen zunächst exorbitante Kosten oder Risiken auf. Das soziotechnische System „wehrt“ sich, auf allen Ebenen dagegen seine Struktur stark zu verändern. Parallel hierzu verbraucht es aber zum Erhalt des aktuellen Status Quo Ressourcen des Unternehmens, d.h. das System kostet, selbst wenn es nicht stark verändert wird. Ein weiteres Symptom dieses Gleichgewichts sind die immensen Aufwände, die notwendig sind, um geringe strukturelle Veränderungen vorzunehmen; der Grund liegt darin, dass das System sein Gleichgewicht verlassen muss um ein neues zu erreichen und dieser Weg ist meist sehr kostspielig.



Wenn das Legacysystem sein stabiles Gleichgewicht erreicht hat, dann kann es sich nicht mehr – aus sich selbst heraus – strukturell massiv verändern. Der Impuls zu einer Veränderung muss von außerhalb des Systems kommen. Typische äußere Kräfte, die einen solchen Impuls formen können, sind:

- Regulatorische Anforderungen,
- Fusionen oder Übernahmen,
- Marktveränderungen,
- Geschäftsprozessveränderungen,
- Größere Reorganisationen,
- Veränderte IT-Strategien,
- Wegfall von Hardware- und Softwarekomponenten.

Alle diese Kräfte sind dadurch charakterisiert, dass das soziotechnische System nicht mehr in der Lage ist, sich auf die vorgegebene Veränderung anzupassen ohne seine Identität, d.h. seine grundsätzlichen Merkmale, beibehalten zu können. Für das System ist ein vorgegebener Zielzustand durch inkrementelle Veränderungen nicht erreichbar. Folglich scheiden langsame Veränderungen aus, mit der Folge, dass eine revolutionäre Veränderung notwendig wird.

### neue wege

Die Einführung eines neuen soziotechnischen Systems ist eine solche revolutionäre Veränderung. Eine Migration zu einer Standardplattform ist ein möglicher Schritt zur Schaffung eines neuen soziotechnischen Systems. Andere Schritte wären z.B. Business Process Outsourcing oder völliger Verzicht auf bestimmte Geschäftszweige. Jede Form von massiver Veränderung kann i.d.R. nur von Personen außerhalb des bestehenden soziotechnischen Systems initiiert werden und wird – i.d.R. – von Teilen des Systems boykottiert.

Diese Zweiteilung von Personenkreisen – inn- und außerhalb des Legacysystems – erklärt die Schwierigkeit bei der Beantwortung der Frage: „Was machen wir mit unserem Kernsystem?“. Die Outgroup ist in der Lage sich ein „Leben“ ohne das bestehende System vorzustellen, die Ingroup hat Schwierigkeiten bei einem solchen Konzept. Vom bestehenden System wegzugehen fällt umso leichter, je weniger der Entscheider Teil des Systems ist. Nicht Teil des Systems zu sein, ist in den meisten Fällen identisch damit, nicht Teil der Historie des Systems zu sein, folglich kann ein Impuls zum Wechsel am Einfachsten durch Entscheider produziert werden, die von außerhalb des Unternehmens – d.h. nie Teil des soziotechnischen Systems waren – kommen.

### Entscheidung

Ist die Entscheidung gefallen, vom bestehenden Legacysystem wegzugehen, dann stellen sich die ersten Hindernisse ein. Es stellen sich sofort zwei Fragen: Wohin und Wie?

Meist existiert auf dem Markt für kommerzielle Software kein Produkt, welches die Software im bestehenden Legacysystem direkt ersetzen kann. Zu viele Eigentümlichkeiten des Unternehmens sind im Laufe der Historie in die Software gewandert. Ein vollständiger Ersatz bestehender Prozesse wird hingegen als so bedrohlich empfunden, dass die Prozesse nicht fundamental überdacht werden können. Erfahrungsgemäß ist Verhaltensänderung in Organisationen sehr schwierig. Konfrontiert mit dem Dilemma, Änderungen durchführen zu müssen, aber gleichzeitig die bestehenden Mechanismen nicht zu verändern, erscheinen Standardplattformen ein Ausweg.

Der Begriff „Standard“ suggeriert zunächst einmal Allgemeingültigkeit und Stabilität. Der Zusatz „Plattform“ erweckt Assoziationen in Richtung Ausbaufähigkeit und Veränderbarkeit. Forciert wird dies durch die Hersteller, die suggerieren, dass durch Konfiguration die Plattform beliebige Funktionalitäten abdecken kann. Meist sind die Hersteller auch noch so „hilfreich“ und haben ein Netzwerk von Partnern, die solche Anpassungen vornehmen können; für beide, Hersteller und ein gutes Geschäft, da zum einen der vorgebliche Standard nie in Frage gestellt wird – oft zitieren hier die Hersteller die Anzahl von Kunden als einen „Beweis“ für die Existenz und die Einhaltung eines Standards – und für die Implementatoren, welche die vollständige Malleabilität der Software in Aussicht stellen. Die Verkaufsargumente werden dann durch sogenannte Success-Stories „untermauert“, Projekte, die die Standardplattform erfolgreich eingesetzt haben. Was hierbei erfolgreich bedeutet, ist meist fragwürdig, da die Beteiligten so intensiv mit dem Erfolg des Projekts verbunden sind, dass sie nicht in der Lage sind, sich einen Misserfolg einzugestehen.



Innerhalb des Unternehmens hingegen beginnt ein intensives Marketing. Auf der Managementebene erscheint der Einsatz einer Standardplattform zukunftsweisend und liefert die Chance, bisherige Probleme zu beseitigen. Diese Nachricht wird innerhalb des Managements intensiv reiteriert und bekommt mit zunehmender Zahl von Wiederholungen immer mehr subjektiven Wahrheitsgehalt.

### **Standardplattform:**

Unter einem Standard versteht man eine allgemein anerkannte Art und Weise etwas durchzuführen oder herzustellen. Eine Plattform hingegen liefert ein abstraktes Funktions- oder Servicemodell, welches allgemeinen Einsatz erlaubt. In den meisten Fällen, wird im Softwaremarketing der Begriff Plattform als Synonym für eine Ansammlung von Anwendungen einer Marke, der Begriff Standard als Euphemismus für (angestrebten) Verbreitungsgrad genutzt.

Die Mitglieder der Ingroup (sozialer Bestandteil des Legacysystems) hingegen liefern schnell Risikoszenarien, um den Wechsel zu verhindern. Typische Argumente sind:

- Die gesamte fachliche Funktionalität des bestehenden Systems ist unbekannt, daher ist unklar, was passiert, wenn es abgeschaltet wird.
- Die Datenmodelle der Standardplattform und unserer Anwendungen lassen sich nicht aufeinander abbilden: Wir werden Daten verlieren.
- Die Historie lässt sich nicht abbilden, aber die Wirtschaftsprüfer fordern dies für die letzten 12 Jahre.
- Die Schnittstellen zu den anderen Systemen sind unbekannt, wenn es zu Problemen kommt, steht alles still.
- Fachbereiche hingegen bemängeln meist fehlende fachliche Funktionen, welche ihnen angeblich sehr viel Zeit einsparen.

Beide Lager kämpfen mit ihren jeweiligen Argumenten um die Meinung innerhalb des Unternehmens. Um die Risikoszenarien abzuschwächen und den Fachbereichen die scheinbar benötigte Funktionalität zu liefern, wird angekündigt auf der Standardplattform die entsprechenden Veränderungen vorzunehmen – sehr zur Freude der Implementatoren. Parallel hierzu wird klar, dass man das vorhandene Legacysystem so lange weiter betreuen muss bis die letzte Schnittstelle ersetzt wurde; ein Vorgang, der mehrere Jahre dauert – sehr zur Freude der Entwickler. Am Ende verschwindet das Legacysystem oft überhaupt nicht vollständig, sondern es bleiben Reste übrig, deren Ablösung teuer ist, ohne einen Mehrwert zu bieten, so dass jeder die entsprechende Investition scheut.

### **Realität**

Bei dem Vergleich zwischen der bestehenden Legacysoftware (Altsoftware) und einer aus der Veränderung einer Standardplattform resultierenden Anwendungssoftware (Neusoftware) ist man schnell geneigt die Neusoftware in Hinsicht auf Pflegbarkeit, Architektur, Lebensdauer und Stabilität, für überlegen zu halten. Ist das wirklich der Fall?

Die Neusoftware hat zwei Quellen, welche ihre Struktur stark beeinflussen. Zum einen die eigentliche Standardplattform und zum anderen die Adaption der Plattform durch den Implementator.

Die Plattform muss a priori eine gewisse funktionale Reife haben, sonst kann sie für Nutzer kaum sinnvoll eingesetzt werden. Diese funktionale Reife kann aber nur über eine längerfristige Evolution erreicht werden, so dass die in der Plattform enthaltene Software stets ein gewisses Alter besitzt. Zwar wird sie meist jünger sein als die Altsoftware, trotzdem kann sie durchaus 10-20 Jahre aufweisen. Daneben finden sich in der Plattform immer Strukturen, die ein Abbild der Herstellerorganisation bzw. der ersten einsetzenden Organisationen darstellen, das vorab erwähnte Gesetz von Conway. Folglich müssen die Strukturen in der Software, sprich die Softwarearchitektur, nicht zu den Strukturen des Unternehmens, sprich Prozesse und Kommunikationswege, passen. Diese Verwerfungen werden dann meist durch eine spezielle Form der Implementierung überbrückt, mit der Konsequenz, dass die implizit vorhandenen Architekturen in der Plattform in der Neusoftware verwässert erscheinen. Der theoretisch beste Weg, die Adaption der eigenen Prozesse, findet selten statt, da im Vorfeld die vermittelte Veränderbarkeit der Plattform einen Adaptionsprozess ausgeschlossen hat und die wenigsten Projekte in der Lage sind, solche Veränderungen ad hoc anzustoßen, geschweige denn umzusetzen.



Der Implementator hingegen nutzt als ein Modell seine eigene Unternehmensstruktur und die Struktur des Unternehmens für das er die Neusoftware schaffen soll, mit dem Resultat, dass nun diverse übereinander liegende Strukturen abgebildet wurden. Die so geschaffene Neusoftware ähnelt oft der Altsoftware in bestimmten Aspekten und ist meist nicht sehr viel besser wartbar. Nach der Freigabe entsteht ein neues soziotechnisches Anwendungssystem nun allerdings mit dem Implementator und der Neusoftware als Teile des Systems. Recht rasch findet dieses System dann einen Gleichgewichtszustand und entwickelt ein Eigenleben anlog zum alten Legacysystem. Neben dem reinen Ersatz eines einzelnen Legacysystems zeigen Standardplattformen noch ein anderes Phänomen auf: Sie bilden rasch ein Schwerkraftzentrum im Unternehmen. Meist wird mehr als eine Anwendung auf der Plattform realisiert, mit der Konsequenz, dass das entstehende soziotechnische System sprunghaft wächst und deutlich größer wird als das alte System. Das so gewachsene neue System wird sich noch stärker im Unternehmen verankern und allein durch seine schiere Größe und die daraus resultierende Komplexität immer weniger beherrschbar werden.

## **Lösung**

Wie kann man diesem Circulus Vitiosus entkommen? Der ursprüngliche Fehler liegt im Festhalten an den bestehenden Prozessen. Jede Form von Ersatz eines soziotechnischen Systems muss mit der Frage nach der Sinnhaftigkeit der bestehenden Prozesse beginnen. Wenn diese Prozesse nicht wirklich differenzierend sind, dann müssen sie in der bestehenden Form auch nicht aufrechterhalten werden. Dann können auch die Prozesse aus einem Standardsoftwarepaket genutzt werden. Dieser Schritt ist zwar schwierig und weist viele Widerstände auf, führt aber letztlich zu einer Konzentration auf wirklich wichtige Prozesse. Selbst wenn der Prozess differenzierend ist, dann stellt sich die Frage, ob seine heutige Prozessform noch sinnvoll ist und welche steuernden Parameter wirklich gebraucht werden. Hier empfiehlt sich simultanes Design, d.h. die parallele Entwicklung des Systems auf der Prozess- als auch auf der Softwareebene. Insofern ist klar, was eine Standardplattform leisten muss: Sie muss genügend Freiheit für das simultane Design liefern. Für alle anderen Einsatzformen gibt es meist Standardsoftware!

Auf der Seite der Governance ist es besonders wichtig für Kontroll- und Investitionsmechanismen zu sorgen, die es erlauben, das Altsystem tatsächlich vollständig abschalten zu können. In den meisten Fällen gibt es für das Abschalten keinen Business Case, der sich auf kurze Dauer rechnet, dies führt langfristig zu einer Proliferation an nichtabschaltbaren Systemen, die zu einer sehr hohen Gesamtkomplexität massiv beitragen.