

Editorial

Liebe Leserinnen, liebe Leser



Ich muss Ihnen wohl nicht das Unwort des Jahres verraten und werde es auch nicht in dieser Einleitung verwenden. Doch juckt es mich schon etwas, darüber ein paar Worte zu verlieren, denn soviel Unvernunft und

Liederlichkeit im Umgang mit dem Eigentum anderer geht mir einfach nicht in den Kopf. Die Folgen davon sind noch nicht absehbar, aber grundsätzlich zahlen wir alle dafür, einerseits mit unseren Steuern, andererseits ist es wohl nicht zu vermeiden, dass unsere Wirtschaft auch etwas *abbekommt* und wir erneut einige Stunden mehr arbeiten müssen, bis wieder einigermaßen Normalbetrieb einkehrt. Immerhin können wir uns glücklich schätzen, wenn wir dafür arbeiten können, dann ist es nur halb so übel.

Mein Rezept dafür ist, vorwärts schauen und produzieren, was das Zeugs hält. Das vernichtete Geld (das ja nur fiktiv vernichtet ist) ist das kleinere Problem, den Dämpfer aber, den wir als Konsumenten erhalten, ist eher der langfristige Schaden. Dabei kommt mir ein Gedanke: Zufällig ist das ja nicht entstanden, aber wem nützt es und dann, was passiert bei diesem lauten Geschrei im Hintergrund im Stillen? Auf letzteres sollten wir am meisten Acht geben, denn das wird uns langfristig noch viel mehr bewegen, wenn wir es einfach zulassen.

Um hier keinen Pessimismus aufkommen zu lassen, freut es mich ihnen nach einer Pause unsere nächste *Info Bridge* zu präsentieren und ich hoffe, sie finden etwas darin, das auch ihr Interesse erlangt.

Viel Vergnügen!

Mit freundlichen Grüßen

Ihr  Andreas Koch

Das sagen unsere Kunden: Peter Roth, Meteo Schweiz



Die Ausgangslage

MeteoSchweiz versorgt auf den Flughäfen Zürich und Genf die Piloten und Stationsbetreuer der Fluggesellschaften mit den aktuellen Flugwetterdaten der anzufliegenden Destinationen. Dazu wird die Applikation TAMSİ eingesetzt.

Diese bewährte Applikation ist in die Jahre gekommen und soll durch eine neue Anwendung ersetzt werden. Die neue Applikation soll weiterhin die Vorzüge der zu ersetzenden Anwendung TAMSİ beinhalten, aber auch die Vorteile der modernen Technologien bieten.

Die Lösung

Als Basis für die Neuentwicklung dieser TAMSİ Applikation wurde beschlossen, eine umfassende Anforderungsanalyse erstellen zu lassen. Object Engineering GmbH wurde als externer Partner beigezogen, um Analyse und Pflichtenheft für die Neuentwicklung zu erstellen.

Die Ausführung

Folgende Schritte galt es auszuführen:

- Studium und Einführung ins bestehende System.
- Beobachtung der typischen Benutzer am bestehenden System auf dem Flughafen.
- Interviews mit Systemverantwortlichen für TAMSİ am Flughafen und ausgewählten Benutzern.
- Aufstellen und Beschreiben der funktionalen (Use Cases) und nichtfunktionalen Anforderungen.
- Ansatz einer modernisierten Benutzeroberfläche definieren und dokumentieren (GUI und WEB).
- System-Architektur des neuen Systems skizzieren und dokumentieren.

Der Nutzen

Object Engineering erstellte ein präzise formuliertes Pflichtenheft mit den genauen Anforderungspunkten.

Für die Ausschreibung konnten wir den Anbietern ein klar formuliertes Pflichtenheft abgeben.

Dies ermöglichte den Offertstellern fundierte Angebote einzureichen, die sich in einer engen zeitlichen Aufwandsspanne bewegten. Dies wiederum vereinfachte uns die Beurteilung der Angebote und stellt sicher, dass die neue Applikation den gewünschten Anforderungen entspricht. Die professionelle Art der Pflichtenhefterstellung durch Object Engineering dient der MeteoSchweiz auch als Vorgabe zur Erstellung weiterer Pflichtenhefte in späteren Projekten.

MeteoSchweiz

Der nationale Wetterdienst MeteoSchweiz erfüllt wichtige Aufgaben für Bevölkerung, Staat und Wirtschaft. Neben dem Bereitstellen von allgemeinen Wetterprognosen warnt MeteoSchweiz die Einsatzbehörden der Kantone, wenn Sturmwinde, starke Niederschläge und Gewitter drohen. Dies geschieht über geschützte Informationskanäle, die rund um die Uhr einsatzbereit sind. Die zivile, militärische und private Luftfahrt ist nicht denkbar ohne eine hochspezialisierte Wetterberatung für Piloten und Flughäfen. Diese Leistungen sind nur dank einer hochentwickelten Informatik möglich, die laufend ausgebaut wird.

Die Wahl des Partners

Object Engineering GmbH verfügt über langjährige Erfahrung mit der Renovation und Ablösung von Altssystemen. Dies beinhaltet die gesamte Kette des Engineerings von der Analyse, der Pflichtenheft-Erstellung über Design, Entwicklung, aber auch Erneuerung und Wartung von Software-Applikationen. Object Engineering verfügt in diesem Gebiet über ausgewiesene Referenzen, was uns die Wahl des Partners sehr vereinfachte.

MeteoSchweiz, Peter Roth
Teamleiter IT-Entwicklung
www.meteoschweiz.ch

Ablösung von Altapplikationen

Nach dem letzten Artikel über den Bau von SOA-Services, betrachten wir hier Altsoftware und Altssysteme etwas genauer. In jeder SOA kommen wir nicht umhin, bestehende Komponenten, die oft als "Altsoftware" oder Legacy bezeichnet werden müssen, in die Architektur-Landschaft einzubinden. Möge nachfolgender Artikel ihnen eine Idee über Wert und Problematik solcher Systeme geben.

Einleitung

Als Altssysteme und Altsoftware bezeichnet man Systeme, welche einen grösseren oder kleineren, aber relevanten Teil der Geschäftslogik und -Prozesse in einem Unternehmen unterstützen, die technologisch in die Jahre gekommen sind. Natürlich gibt es auch technische Systeme wie Produktionssteuerungen, Leitsysteme, Analyse-Systeme, welche als Altssysteme gelten können. Sowohl Hardware- wie Software-Komponenten können als Altkomponenten klassiert werden. Nachfolgend schliessen wir mit Altssystem auch die dazugehörige Altsoftware mit ein oder meinen sogar nur die Altsoftware.

Was ist alt?

Bei fortschreitendem Ausbau der IT-Landschaft und der Sicherstellung vom Betrieb können Altssysteme zum Hemmschuh werden. Durch Ablösung solcher Altssysteme kann die Flexibilität wiederhergestellt werden und neuere, effektivere Technologien können flächendeckend eingesetzt werden.

Wann werden Systeme und Software zu Altssystemen oder Altsoftware? Grundsätzlich wird ein jedes System nach zwei bis drei Jahren nach dessen Entstehung zum Altssystem. Wenn an einem System keine Veränderung mehr vorgenommen wurde, entschwindet es aus der Erinnerung der Verantwortlichen und sicherlich der Entwickler, welche sich wieder um neuere Systeme kümmern müssen. Je länger die Zeit voranschreitet, je weniger Personen gibt es, die das System noch gut kennen. Zwei Jahre ist in etwa die Halbwertszeit des Know-hows über eine Software oder ein System. Sobald die nächste Anpassung gemacht werden muss, stellt sich heraus, wie gut und praxisnah dokumentiert wurde. Obiges gilt in der Regel unabhängig von der verwendeten Technologie. Auch in Java und in Java EE werden Applikationen mit der Zeit zu Altsoftware. Es

gibt Systeme, die technologisch und auch funktionsmässig veraltet sind, andere Systeme wiederum, sind nur technologisch im Hintergrund.



Wie entstehen Altssysteme?

Wenn in unregelmässigen Abständen Anpassungen an einem System gemacht werden, dann werden in der Regel keine grundlegenden Überarbeitungen oder Re-Designs mehr gemacht. Der Grund dafür ist naheliegend. Für die Funktions-Anpassung oder -Erweiterung ist gerade genügend Budget vorhanden und der enge Terminplan lässt keine grössere Überarbeitung zu.

Dies führt dann innerhalb gerauerer Zeit (4-6 Jahre) dazu, dass ein „Patchwork“ entsteht, welches Änderungen immer schwieriger gestaltet und zu mehr und mehr Nebeneffekten führt. Insbesondere geschieht dies, wenn mehrere unterschiedliche Entwickler die Änderungen vornehmen und dem Software-Design nicht mehr genügend Sorge tragen und dieses nicht gut dokumentieren (zum Beispiel in Form von UML) und verständlich halten. So kann es schnell passieren, dass Anpassungen am „falschen Ort“ der Applikation vorgenommen werden und dadurch spätere Erweiterungen erschwert oder mit den vorhandenen Ressourcen (Personal, Geld und Zeit) praktisch unmöglich werden.

Offensichtlich ist also: Altssysteme entstehen nicht nur durch Zeit, sondern auch durch die Häufigkeit von kleinen Änderungen ohne periodische Design-Überprüfung und Überarbeitung.

Gründe für Altssystem-Renovation

Nachfolgend sind einige Ursachen aufgeführt, welche für die Entstehung, respektive den Bedarf einer Altssystem-Renovation oder sogar des -Ersatzes den nötigen Handlungsbedarf geben:

- Anpassungs-Patchwork, welches die Anpassungen des Systems an neue Anforderungen zu einem zu hohen Risiko werden lassen, dass andere Systeme oder Prozesse nicht mehr korrekt ablaufen könnten.
- Zu hohe Komplexität, veraltete Technologie, für welche die nötigen Ressourcen (Know-how-Träger) nicht mehr verfügbar sind.
- Know-how-Verlust durch fehlendes oder „Überalterung“ des Personals, verlorene Dokumentation (Aktualität, Gültigkeit, fehlende Archivierung, Format), fehlende oder nicht aktuelle Programm-Quellen oder -Bibliotheken.
- Undokumentierte oder fehlende Schnittstellen.
- Technologie-Einschränkungen (Speicherlimiten).
- Applikationslogik, welche zu viel des (veralteten) Geschäftsprozesses beinhaltet, was eine übergeordnete Steuerung des Geschäftsprozesses übermässig komplex, enorm aufwändig oder unflexibel macht.
- Zu kostspielige Teilkomponenten, wie Software-Lizenzen für verwendete Systemkomponenten oder Hardware-Wartung, welche typischerweise regelmässig im Preis steigen, bevor der Lieferant sie gar nicht mehr wartet.
- Überflüssige und nicht mehr verwendete Funktionen (toter Code), welche aber weiterhin gepflegt werden müssen, da man nicht genau weiss, ob diese noch verwendet werden.
- Nicht fachgerechtes oder falsches Design und Programmierung aus frühen Entwicklungsphasen führen in der Regel zu einem von allen beteiligten Entwicklern peinlichst gemiedenen Code mit dem Ruf „Bitte nicht berühren!“
- Technologie-Konsolidierung, da zu viele verschiedene oder veraltete Technologien für die Implementation des Altsystems verwendet wurden, wofür keine dafür spezialisierten Ressourcen mehr auf dem Markt verfügbar sind.

... ein bewährtes Vorgehen

- Grundlegende Änderung der IT-Architektur-Strategie des Unternehmens, zum Beispiel durch Umstellung auf Serviceorientierte Architektur, welche eine hohe Modularität und eine Prozesssteuerung ausserhalb der Applikationskomponenten voraussetzt.

Es gibt sicher noch mehrere weitere Kriterien, welche aufgeführt werden könnten, doch obiges deckt einen grossen Teil der typischen Gründe ab.



Arten von Renovationen

Es gibt verschiedene Möglichkeiten die Renovation von Altsystemen durchzuführen, hier einige der typischen aufgeführt:

- Schnittstellen-Migration, wie zusätzliche Schnittstellen oder Anpassung von Schnittstellen und die schrittweise Überführung in eine neue, aber auch Optimieren von falsch konzipierten Schnittstellen.
- Parallelaufbau einer neuen Applikation, welche schrittweise die alten Funktionen übernimmt. Dabei verwendet man einen speziellen Adapter, welcher wahlweise die Funktion auf dem neuen oder alten System aufruft. Man sollte jedoch die letzten 10-20% der Funktionalität gut im Auge behalten, da es sich meist um Funktionen handelt, welche nicht dem systematischen Design folgen, Nebeneffekte haben oder im schlimmsten Fall versteckte Funktionen, welche nicht das erwartete Resultat erbringen. Die Isolation dieser Funktionen kann einen Grossteil der Zeit benötigen.
- Big Bang-Ansatz, bei welchem das System vollständig durch ein neukonzipiertes und eventuell erweitertes System ersetzt wird. Dieser Ansatz ist sehr wohl abzuwägen, aber es ist sicherzustellen,

dass im Nichterfolgsfall das noch bestehende System weiterhin eingesetzt werden kann.

- Das Beibehalten der Datenbank (und des Modells) und dem Neukonzipieren der Funktionalität typischerweise auf einer neuen Plattform und Technologie. Dies ist oft eine gute Migrations-Strategie, da die bestehende Applikation in gewissen Fällen parallel weiterlaufen und so das Risiko minimiert werden kann.
- Service-Migration: Will man eine Altapplikation oder einen gewissen Funktionssatz davon als einen oder mehrere Services innerhalb einer serviceorientierten Architektur verwenden, wird meistens das sogenannte Wrapping angewendet. Dabei muss die Schnittstelle des Wrappers der gewählten Architektur entsprechen und die entsprechende Umsetzung auf die Altapplikation adaptiert werden.

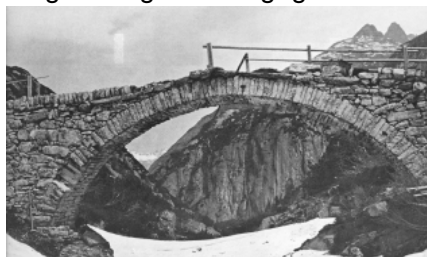
Vorgehen

Bei der Ablösung oder Renovation von Altsystemen und Altapplikationen folgt man in der Regel einem eher konventionellen Entwicklungsprozess einer herkömmlichen IT-Applikation. Da aber das in Angriff genommene System in der ersten Phase zu den betroffenen Umsystemen meistens „kompatibel“ bleiben muss, sollte entsprechend viel Aufmerksamkeit auf Analyse und Planung gelegt werden.

Eine Migrationsstrategie beinhaltet:

- Analyse des Bestehenden
- Architektur des Veränderten oder Neuen
- Organisation und Planung
- Kompatibilitäts- und Systemtests (mit Umsystemen)

Eine Ad-hoc-Lösung zahlt sich selten aus, professionelles und sorgfältiges Vorgehen hingegen schon.



Zudem muss berücksichtigt werden, dass fast jedes System ein Unikat und genau zu analysieren

ist, bevor es renoviert oder ersetzt wird. Der Analysephase ist darum hohe Wichtigkeit beizumessen, da meistens das Wissen mit den Know-how-Trägern wegging. Relevant auch darum, da sichergestellt werden muss, dass spezifische Informationen nicht übersehen werden, welche eine Migration oder Ablösung einfacher realisieren lassen. Ein solcher Schlüsselfaktor sind Meta-Daten, welche man für die automatische Generierung von Programmen (z.B. Schnittstellen) oder zur Verwendung in einem Data-Dictionary einsetzen kann. Die in die Analyse investierte Zeit kann sich später in der Realisierung mehrfach auszahlen.

Es gehört eine gute Portion an Erfahrung, Systematik und damit verbundenes Gespür dazu, wie man diese Schlüsselfaktoren auffinden kann. Zwar gibt es, wie oben erwähnt, verschiedene Muster nach denen ältere Systeme gebaut wurden, was für die Analyse hilfreich ist, doch darf man sich nicht nur darauf verlassen. Gerade später eingebaute Problem-Umgehungen oder Spezial-*Hooks* (Aufhänger) können 10-20% der Funktionalität ausmachen.

Wurden Systeme mit technischen Schnittstellen gebaut oder bereits mit Generatoren entwickelt, kann man von einem höheren Mass an sturer Systematik ausgehen, was für den Einsatz von automatisierten Werkzeugen (in der Regel Software) von grossem Vorteil ist.

Die typischen Prozessschritte sind:

- Festlegung der Systemgrenzen
- Eingehende Analyse des bestehenden Systems
- Erstellen des Ablöse- oder Renovations-Konzepts inklusive Architektur und Grobdesign
- Erstellen der Einführungs- und Migrationspläne
- Einholen von Offerten aufgrund des Ablösekonzepts und Entscheidung durch Kunden für die gewünschte Lösung
- Realisierung: Ausarbeitung der Pläne, frühzeitige Koordination mit den betroffenen Stellen (Kick-off) und Abstimmung für das entsprechende Release
- Architektur und Detail-Design und Ausarbeiten der Testpläne
- Implementation: Generatoren und manueller Code

- Systemtests in produktionsähnlicher Umgebung
- Einführung

Durch die vorgezogene Analyse und dem daraus erstellten Lösungskonzept können viele der möglichen Falltüren identifiziert werden. Auch wenn nie alles vorausgesehen werden kann, zahlen sich die gründlich gemachten Abklärungen aus. Es muss bei Altsoftware berücksichtigt werden, dass zusätzlich zur Renovation oder Ablösung, neue Anforderungen und entsprechende Anpassungen dazu kommen. Dies kann speziell schwierig werden, wenn sich die Renovation über eine längere Zeit hinzieht und zwischenzeitlich mehr als ein Software-Release realisiert wird.

Alle Vorbereitungsarbeiten, wie das Erstellen von Konversionsprogrammen oder Metadate-basierte Generatoren, sollten vor dem Anfassender zu renovierenden Software gemacht werden, damit im Notfall auf die bestehende Applikation zurückgegriffen werden kann. Gerade diese Ausweichvarianten dürfen nicht vergessen werden und müssen mit den beteiligten Organisationen genau abgeklärt werden. Zum Beispiel kann es ein "point of no return" geben, wenn für die Umstellung viele Daten konvertiert werden müssen, so dass für das Zurückfahren beim Feststellen eines Fehlers bis zur Produktionsaufnahme keine Zeit mehr bleibt (meistens ein normales oder verlängertes Wochenende).

Fazit

Die Renovation oder Ablösung von Altsystemen ist in der Regel kein „Honigschlecken“, doch hilft die Vogel-Strauss-Methode auch nicht, das laufend wachsende Problem mit den Altsystemen einfach zu ignorieren.

Mit einer eingehenden Analyse, einem cleveren Lösungsansatz (Software Engineering), einer guten Planung und einer gut geführten und qualitativ hochstehenden Realisierung können die meisten Systeme mit vernünftigem Aufwand umgestellt werden. Eine Falle dabei ist, die Sache zu spät und ohne genaue Analyse und Planung anzugehen.

Andres Koch,
dipl. El. Ing. HTL, M. Math (CS)

Amazonas Workflow

Seit diesem Herbst steht nun die neue Version 3.0 von Amazonas Workflow bereit. Neu ist Amazonas-Workflow auf den Java Plattformen 5 und 6 verfügbar. Neben diversen Verbesserungen sind neue Features eingeflossen:

- Das SendMail-Worklet unterstützt nun auch das Versenden von E-Mails mittels SMTP-Authentification-Verfahren.
- AWF-Reporting Funktionalität (Option) basiert neu auf Jasper 2.

Amazonas Workflow V 3.0

Amazonas steht mit dem folgenden Link zum Download bereit:
<http://web06.objeng.ch/support/downloads/awf/workflow-de.shtml>

Für weitere Informationen oder Fragen wenden Sie sich am einfachsten an unseren Verkaufsberater: Tel: 044 400 47 00 oder
E-Mail: sales@objeng.ch

Wettbewerbs-Auflösung aus der InfoBridge Nummer 8:

Der Name dieser alten, malerischen Doppelbrücke ist **Ponte dei Salti** oder auch Römerbrücke genannt. Sie führt bei Lavertezzo über die ja-degrüne Verzasca. Das Verzasca-Tal ist eines der fantastischen Tessinertäler, das zu durchwandern sich lohnt.



Der ausgeloste Gewinner heisst **Erich Stucki**. Er gewinnt Kostproben von einigen typischen Tessiner Köstlichkeiten.

Herzliche Gratulation!

Wir danken für die rege Beteiligung. Wenn es auch dieses Mal nicht geklappt hat, nicht verzagen, es gibt bereits in dieser Nummer den nächsten InfoBridge-Wettbewerb.

Brückenwettbewerb

Die Olympischen Sommerspiele 2008 in Beijing sind vorbei, die vielen Eindrücke bleiben...

In unserem Brückenwettbewerb suchen wir nach dem Namen der abgebildeten Brücke, die sich in dem Land der Spiele über einen See spannt.



Ihre Lösungsantwort senden sie bitte bis zum **31. Dezember 2008** an:

E-Mail: infobridge@objeng.ch
Fax: **044 400 47 07**

Unter den Teilnehmern mit der richtigen Antwort wird eine passende Überraschung ausgelost.

Viel Glück!

Object Engineering GmbH

Birmensdorferstr. 32
CH-8142 Uitikon-Waldegg

Tel: +41 (0) 44-400 47 00
Fax: +41 (0) 44-400 47 07

www.objeng.ch
info@objeng.ch

SNG

Member of the
Solution Network Group

Object Engineering® ist ein eingetragenes
Warenzeichen im Besitz der
Object Engineering GmbH

Ihr Partner für:

- Renovation, Ablösung oder Migration von Altsystemen und Altsoftware
- Integration jeglicher Systeme in ihr Geschäftsumfeld
- Objektorientierte Software-Architektur und Technologie (Java, C++)
- Service Orientierte Architektur