

Toolbasierte Software-Migration nach Plan

Christian Becker, Uwe Kaiser

pro et con Innovative Informatikanwendungen GmbH, Dittesstraße 15, 09126 Chemnitz

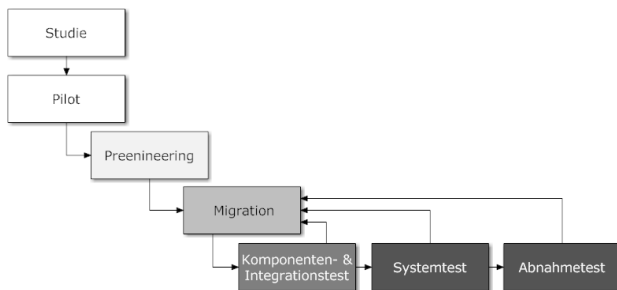
{christian.becker, uwe.kaiser}@proetcon.de

Abstract

Die Firma proetcon hat aus aktuellen und in der Vergangenheit realisierten Migrationsprojekten Erfahrungen zu deren Planung und Ablauf gesammelt. Der vorliegende Beitrag beschreibt einige dieser Erfahrungen und skizziert den Ablauf von toolbasierten Migrationsprojekten.

1 Projektphasen

Die Realisierung eines Migrationsprojektes erfolgt analog zu konventionellen Softwareprojekten in verschiedenen Phasen. Die nachfolgende Grafik stellt den favorisierten Ablauf eines Migrationsprojektes verallgemeinert dar:



Die Phasen müssen für jedes konkrete Migrationsprojekt und für jeden Kunden entsprechend der dann vorliegenden Anforderungen angepasst werden. Wesentliche Aspekte wie die Erstellung einer Studie, die Realisierung eines Piloten oder die paketweise, parallele und iterative Durchführung der Migration und des Tests sind jedoch für jedes Migrationsprojekt gegeben.

2 Studie

Die erste Phase des Projektes beinhaltet das Anfertigen einer Studie. Wesentlicher Bestandteil ist ein Überblick und ein Mengengerüst über alle Artefakte (Masken, Programme, Jobs, Dateien und Tabellen) und Schnittstellen des Basissystems. Bei diesem Reverse Engineering leisten Analysewerkzeuge einen wesentlichen Beitrag. Sie basieren auf Metawerkzeugen, welche projektspezifische Anpassungen der Analysewerkzeuge mit überschaubarem Aufwand ermöglichen. Ein weiterer Bestandteil der Studie ist die eindeutige Definition des Zielsystems und der Migrationspfade, d.h., einer Abbildungsvorschrift zur Umsetzung aller Komponenten aus dem Basissystem in das Zielsystem. Daraus lässt sich der zu leistende Aufwand ableiten, um eine existierende Toolbox für die Software-Migration an die kundenspezifischen Projektanforderungen anzupassen. Das Spektrum reicht dabei vom „Nichtstun“ bis zur Neuentwicklung, wobei Erfahrungen zeigen, dass es „fertige“ Werkzeuge nicht gibt. Ein gewisser Entwicklungsaufwand ist immer zu leisten und muss natürlich in die Aufwandabschätzungen für das Gesamtprojekt einfließen. Mit den o.g. Analysewerkzeugen lassen sich auch fein-

granulare Statistiken zum Auftreten und zur Häufigkeit von Codefragmenten ermitteln, welche für die Migration kritisch sind. Neben diesen teilautomatisierten Analysen sind zielgerichtete und individuelle Interviews mit den Entwicklern sowie die Prüfung der bestehenden Dokumentation ein unverzichtbarer Bestandteil der Analysephase. Aus diesen Informationen lässt sich eine Abschätzung des personellen und zeitlichen Aufwandes vornehmen unter der Voraussetzung, dass die vorhandenen Technologien und Werkzeuge zum Einsatz kommen. In dieser Studie wird dem Kunden also eine „Hausnummer“ genannt, was das Projekt kostet. Aus diesem Grund ist die Erstellung der Studie unverzichtbar, auch wenn bereits alternative Auswertungen zum Basissystem, z.B. durch einen Mitbewerber, existieren. In dessen Abschätzung des Aufwandes kann die Aufwandsparnis, welche durch die Nutzung der existierenden Migrationstools erreicht wird, nicht enthalten sein.

3 Pilot

Die nächste Phase beinhaltet die Realisierung eines Piloten. Die Ergebnisse der Studie werden an einem Ausschnitt des Basissystems in Form eines vertikalen Durchstichs prototypisch evaluiert. Beim Piloten handelt es sich um die Migration eines oder mehrerer vollständiger Geschäftsprozesse des Basissystems. Alle Artefakte (Masken, Programme, Jobs und Daten) des Basissystems sind enthalten und können deshalb eigenständig getestet werden. In dieser Phase werden bereits vorhandene Lösungsansätze evaluiert und an die kundenspezifischen Anforderungen angepasst. Aufgrund der frühzeitigen Durchführung dieser Phase werden ggf. Teile des Piloten manuell umgesetzt, da zu diesem Zeitpunkt noch keine ausreichende Werkzeugunterstützung existiert. Das Ergebnis ist neben einem „proof of concept“ auch ein technischer Fahrplan, der für die gesamte Migration gilt. Der Pilot gibt dem Kunden einen präzisen Ausblick auf das zukünftige, vollständig migrierte Zielsystem und ist somit ein wichtiger Bestandteil der Risikoabschätzung. Anhand dieser Informationen kann der Kunde aus technischer Sicht die Entscheidung über die Weiterführung des Migrationsprojektes treffen.

4 Preengineering

Eine wesentliche Phase eines Migrationsprojektes betrifft das so genannte *Preengineering*. Dieser Begriff fokussiert auf *Reengineering*-Maßnahmen vor der eigentlichen Migration im Basissystem (*Sanieren vor Migrieren*). Dazu gehören z.B. das Aussortieren ungenutzter Programme und Anpassungen am Sourcecode. Typische Preengineering-Maßnahmen bei einer Sprachmigration von COBOL nach Java sind z.B. das Ersetzen von nicht migrierbaren „GO

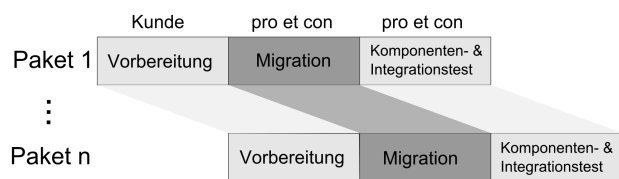
TO“-Anweisungen durch äquivalenten COBOL-Code, die Eliminierung von Klonen in COBOL-Datenstrukturen [1] oder die Bereinigung der Speicherform COMP/COMP-3 für numerische Variablen in den COBOL-Programmen. Dieses Preengineering setzt eine vollständige Datenflussanalyse der jeweiligen COBOL-Sourcen voraus. Dabei kommen Analyse- und Konvertierungswerkzeuge zum Einsatz. In Abhängigkeit von der Art und Häufigkeit der zu sanierenden Artefakte kann demnach auch eine automatisierte Sanierung vorgenommen werden.

Eine weitere Aufgabe ist die Aufteilung des Basissystems in zu migrierende Pakete. Diese Aufteilung wird auf Basis der toolbasierten Analyse von *vertikalen* Geschäftsprozessen (zusammenhängende Komponenten von Jobs, Programmen und Daten) und dem Fachwissen des Kunden vorgenommen. Die Größe der einzelnen Pakete ergibt sich aus den verfügbaren personellen und zeitlichen Ressourcen. Es hat sich bewährt, bei der Planung ein zusätzliches Paket für *vergessene* oder *unerwartet aufgetauchte* Programme einzuplanen. Dieses Paket dient gleichzeitig als zeitlicher Puffer für zusätzlich anfallende Entwicklungsaufgaben, welche sich während der Migration ergeben (können).

Parallel dazu erfolgen notwendige Arbeiten wie der Aufbau der Betriebsumgebung sowie die Realisierung von Build-, Deployment- und Testprozessen. Alle in der Preengineering-Phase durchgeführten Maßnahmen dienen der Minimierung des Aufwandes folgender Projektphasen.

5 Migration und Integrationstest

Die favorisierte Strategie ist die iterative Migration von Paketen. Jedes Paket besteht aus einer Menge zusammenhängender und vollständiger Geschäftsprozesse. Die Anzahl der Iterationen wird mindestens durch die Anzahl der Pakete bestimmt, wobei sich die Bearbeitung eines Pakets aus folgenden Schritten zusammensetzt: Vorbereitung, Migration, Komponenten- und Integrationstest. Die folgende Grafik zeigt die Parallelisierung dieser Schritte:



Im Vorbereitungsschritt erfolgen je nach Projekt und Umfang abschließende Sanierungsarbeiten und die Bereitstellung der zu migrierenden Pakete durch den Kunden. Anschließend beginnen die Migrationsarbeiten. Dabei benötigt die Datenmigration im Vergleich zur Programm- und JCL-Migration zeitlichen Vorlauf. Auf Grund des umfangreichen Datenbestandes ist eine große Varianz in der Datenqualität zu erwarten. Deshalb, und da ein migrierter Datenbestand Voraussetzung für nachfolgende Tests ist, wird die vollständige Datenmigration als erstes Paket realisiert. Durch die Werkzeugunterstützung wird ein Automatisierungsgrad von mehr als 90% bei der Job- und Programmkonvertierung erreicht. Erfahrungen besagen, dass

Jobs und Programme einen Konvertierungsaufwand gemäß der 80:20-Regel erfordern (20% des Aufwands entfallen auf 80% der Arbeiten). Manuelle Arbeiten entfallen z.B. auf die Anbindung neuer Schnittstellen und Datenbanken, auf lokale Performanceoptimierungen sowie etwaige Neuentwicklungen von Programmen, welche im Basissystem so „schlecht“ programmiert wurden, dass eine Migration nicht lohnt. Nach der Migration durchläuft jedes Paket den Komponenten- und Integrationstest bei proetcon sowie einen weiteren Integrationstest beim Kunden. Lokalisierte Fehler bei diesen Tests führen unmittelbar zur Korrektur der Werkzeuge und zur erneuten Migration. Davon profitieren alle nachfolgenden Pakete. Vor jeder Auslieferung eines neuen Pakets wird ein Gesamttest aller bisher migrierten Pakete mit Hilfe von automatisierten Testfällen für Batchprogramme und Tests für Onlineprogramme auf Basis normierter Messages durchgeführt [2]. Das iterative Vorgehen bei der paketweisen Migration minimiert Codefreezes für den Kunden, da bei Notwendigkeit Programme in späteren Paketen aufgrund von Fehlerbeseitigung, Optimierung oder Weiterentwicklung beliebig oft konvertiert werden können. Grundvoraussetzung hierfür ist ein hohes Maß an Automatisierung in der Programmkonvertierung und die parallele, paketweise Migration. Auf dieses Ziel wird seit der Pilotphase hingearbeitet.

6 System- und Abnahmetest

Es finden die üblichen Testphasen Anwendung. Nach Abschluss der Migrationsarbeiten folgen System- und Abnahmetests durch den Kunden in eigenständigen Phasen, welche durch besonders kurze Fehlerbehebungs- und Auslieferungszyklen unterstützt werden. Es wird eine erneute, vollständige Datenmigration zur Aktualisierung des Datenbestandes vor Beginn des Abnahmetests vorgeschlagen. Die Inbetriebnahme erfordert ausgereifte Build- und Deploymentprozesse inkl. einer Fallbackstrategie.

7 Zusammenfassung

Mehrere, erfolgreich realisierte Migrationsprojekte der Vergangenheit führten zur Herausbildung der in diesem Beitrag dokumentierten Vorgehensweise. Zusätzlich ist eine Software-Migration auch immer als gemeinsames Projekt zwischen allen Projektpartnern zu planen und zu realisieren. Auch für den Kunden bedeutet ein solches Projekt einen nicht unerheblichen Aufwand, welcher entsprechend zu planen ist. Dieser Aufwand wird jedoch durch eine umfangreiche Werkzeugunterstützung reduziert.

Literaturverzeichnis

- [1] Becker, C.; Kaiser, U.: Applikationswissen in der Sprachkonvertierung am Beispiel des COBOL-Java-Converters CoJaC. GI-Softwaretechnik-Trends, Band 34, Heft 2.
- [2] Uhlig, D.: Testautomatisierung am Beispiel des COBOL-to-Java-Converters CoJaC. GI-Softwaretechnik-Trends, Band 34, Heft 2.