



## SUCCESS STORY

# Erfolgreiche BS2000-Migration bei ITZBund

Die Wirkung von kühlem Bier oder anderen geistreichen Getränken ist allgemein bekannt. Wir durften im vergangenen Jahr ganz neue Erfahrungen auf diesem Gebiet sammeln. Allerdings ging es für uns nicht um den Genuss, sondern um die toolgestützte Migration eines komplexen Softwareprojektes zur Verwaltung von Alkoholika von BS2000 nach Linux. Wie üblich gab es dabei wieder eine Reihe neuer Herausforderungen. Dieser Bericht vermittelt Erkenntnisse und Erfahrungen aus der Zusammenarbeit zwischen dem Informationstechnikzentrum Bund (ITZBund) und uns, welche in diesem Projekt gewonnen wurden.



**Informations  
Technik  
Zentrum Bund**

## Neue BIBER-und ZEBRA-Gehege

ITZBund ist der IT-Dienstleister des Bundes. 2.400 Mitarbeiter(innen) beraten, unterstützen und sichern den laufenden Betrieb sowie die Versorgung mit Informationstechnik und speziell entwickelten IT-Lösungen. Im Projekt sollten zwei IT-Lösungen (sog. Verfahren) zur Festsetzung und Erhebung von Bier- bzw. Alkoholsteuern migriert werden. Die Verfahren BIBER (Biersteuerberechnung) und ZEBRA (Zentralisierung des Brennens unter Abfindung) sind historisch gewachsen und seit mehr als 30 Jahren im Einsatz. Sie erzeugen ca. 220.000 Bescheide mit einem Steueraufkommen von 740 Millionen Euro pro Jahr. Sie wurden auf BS2000 betrieben und es kamen typische BS2000-Werkzeuge zum Einsatz. Technisch bestehen beide Verfahren aus 60 interaktiven Online-Programmen (IFG-Masken) unter der Steuerung des Transaktionsmonitors UTM und aus 300 Batchprogrammen (COBOL), zu deren Steuerung und Verwaltung ca. 200 SDF-Jobs eingesetzt werden. Die Daten werden in ca. 220 SAM-, ISAM- und LEASY-Dateien verwaltet.

## Was Du heute kannst besorgen...

Wir verfügen über eine firmeneigene Toolbox namens pecBOX (pro et con – Toolbox für die Software-Migration). Mit dieser konnten alle notwendigen Migrationspfade des Projektes unterstützt werden. Die folgende Tabelle dokumentiert diese und die eingesetzten Migrationswerkzeuge.

Legacy-System (BS2000)	Zielsystem (Linux)	Migrationswerkzeug
COBOL	Java	CoJaC
JCL (SDF)	Perl	S2P
Masken (IFG)	HTML, JavaScript	MaTriX
Dateien (SAM)	Linux-Files	FiRe
Dateien (ISAM/LEASY)	Oracle	FiRe
Middleware (UTM)	MidaS	-



Dass qualitativ gute Migrationswerkzeuge die Projektlaufzeit verkürzen und Budget schonen, ist nicht neu. Das Projekt konnte nur so im vorgegebenen, engen Zeitrahmen realisiert werden. Aber auch die Erfahrung: „Fertige Werkzeuge gibt es nicht!“ wurde bestätigt. Konkret mussten z. B. CoJaC und FiRe das „Rückwärtslesen“ von LEASY-Dateien lernen. Bei diesen Anpassungen favorisieren wir die Erweiterung unserer Tools gegenüber einer manuellen Anpassung des Sourcecodes. In frühen Projektstadien kann nicht eingeschätzt werden, warum und wie häufig schon konvertierte Programme erneut konvertiert werden müssen. Die manuelle Anpassung müsste dann jedes Mal wiederholt werden. Da alle Werkzeuge auf deskriptiven Modellen und Metawerkzeugen basieren, ist deren Anpassung in einem akzeptablen Zeitrahmen möglich. pecBOX enthält darüber hinaus auch Systemkomponenten für das Zielsystem. Ein Beispiel dafür ist MidaS (Middleware as a Service). Dieses von uns entwickelte Tool ersetzt den BS2000-Transaktionsmonitor UTM im Zielsystem. Durch einen frühzeitigen Lasttest konnte nachgewiesen werden, dass die Performance von MidaS den praktischen Anforderungen genügt und deshalb auf den Einsatz von kostenpflichtiger Standardsoftware verzichtet werden kann.

## Wissen ist Macht

Die erste Phase eines Migrationsprojektes beinhaltet ein toolbasiertes Reverse Engineering mit unserem Werkzeug Flow Graph Manipulator (FGM). Dabei wird ein spezielles Augenmerk auf jene Aspekte geworfen, die Risiken für die Migration darstellen. Im vorliegenden Fall lagen diese vor allem in der Datenhaltung. Laut definiertem Migrationspfad sollten ISAM- und LEASY-Dateien durch FiRe in Oracle-Tabellen migriert werden. In der Analyse wurde eine Besonderheit identifiziert: Die dynamischen Inhalte aus den Dateien werden bei der Programmausführung durch statische Inhalte (Konstanten) aus den Programmen angereichert und danach über eine Druckerschnittstelle ausgegeben. Schon vor der Migration musste deshalb das Encoding der Daten und Quelltexte (inkl. Konstanten) vereinheitlicht werden. Dieser Punkt betraf also nicht nur eine Anpassung von FiRe, sondern erstreckte sich auch auf weitere Aspekte der Migration. Je früher solche vermeintlichen Kleinigkeiten im Projekt erkannt werden, desto geringer ist der Aufwand ihrer Behebung, da eventuell schon eine Konfiguration der Migrationswerkzeuge ausreicht.

## Versuch macht klug

Nach einer detaillierten Analyse und ausgestattet mit einem vollen Werkzeugkasten sollte man davon ausgehen können, dass die eigentliche Migration direkt in der Breite gestartet werden kann. Unsere Erfahrungen belegen etwas Anderes. In der erstmaligen Kombination von Werkzeugen und Kundenanforderungen stecken unbekannte Herausforderungen. Aus diesem Grund wurde vor dem Gesamtprojekt ein Pilot migriert, welcher einen vertikalen Schnitt durch die Anwendung (Maske(n), Programm(e), Jobs, Daten) geboten hat. Dadurch wurden alle Migrationspfade einmal durchlaufen und die Tools und Technologien (zumindest zum Teil) genutzt. Dieses Vorgehen erwies sich als nützlich, wurden doch wesentliche Erkenntnisse gewonnen. So erfolgte z. B. die Entscheidung, das Drucken in ein separates Tool auszulagern und nicht über die in den Laufzeitsystemen existierenden Lösungen abzubilden. Diese hätten den kundenspezifischen Anforderungen nicht genügt. Aufgrund dieses Vorgehens wurden zusätzliche Anforderungen erkannt und vor der eigentlichen Migration implementiert. Das reduzierte das Risiko, die Werkzeuganpassungen neben den Aufgaben der Migration vornehmen zu müssen.

## Divide et impera

Bei einem Migrationsprojekt, welches alle Facetten von Legacy-Systemen widerspiegelt, besteht die Herausforderung, die Komplexität der Migrations- und Testprozesse zu reduzieren. Im vorliegenden Projekt wurde dem Rechnung getragen, indem unser Partner ITZBund die Anwendung in einzelne Pakete aufteilte. Die Basis für die Aufteilung waren die (Geschäfts-)Prozesse, welche sinnvoll in Pakete zusammengefasst wurden. Die dadurch mögliche Parallelisierung von Migration und Test zeigte im Projekt viele Vorteile. Ein wesentlicher Punkt war auch die Kommunikation zu den Fachleuten von



ITZBund. Diese kennen ihre Prozesse und konnten valide Testdaten bereitstellen. Diese Daten und die definierten Prozesse boten auch die Möglichkeit der Automatisierung. Dazu wurden die Prozesse und die passenden Daten normiert und in Testfälle überführt, was den Aufwand für den Test enorm verminderte. Durch die Aufteilung in einzelne Pakete blieben Migration und Test zu jeder Zeit beherrschbar und es existierte stets ein Überblick über den Projektfortschritt.

## Man lernt nie aus

Am 22.08.2016 wurde das Projekt produktiv geschaltet. Erneut wurde mit diesem Projekt bewiesen, dass der von uns verfolgte, toolbasierte Ansatz funktioniert. Das gilt insbesondere für die automatische Sprachkonvertierung von COBOL nach Java mit unserem Werkzeug CoJaC. Technologien und Werkzeuge können noch so erprobt und ausgereift sein, jedes neue Migrationsprojekt bietet neue Herausforderungen. Dabei entfällt auch immer ein nicht unwesentlicher Aufwand auf unsere Projektpartner. Diesen hat ITZBund durch Fachwissen und aktive Mitarbeit vorbildlich geleistet. Nur durch eine so enge Zusammenarbeit können Migrationsprojekte erfolgreich verlaufen.

Unter [www.proetcon.de](http://www.proetcon.de) können weitere Informationen zu Technologien und Werkzeugen für die Software-Migration eingesehen werden.

Informationen zu ITZBund sind unter [www.itzbund.de](http://www.itzbund.de) zu finden.