

# Architektur-Refactoring



## Ausgangslage

Die Bison Schweiz AG mit Sitz in Sursee ist ein Unternehmen der fenaco Landi Gruppe. Für unsere Kunden innerhalb und ausserhalb der fenaco erstellen wir moderne betriebswirtschaftliche Software-Produkte und -Lösungen, unter anderem für die Agrar- und Retail-Märkte in der Schweiz und in Deutschland. Kerngeschäft der Bison war historisch über viele Jahre die Entwicklung und Vermarktung von ERP-Lösungen. Zum Portfolio gehören daher mehrere ERP-Produkte, die bei zahlreichen Kunden im Einsatz sind und über die dort jedes Jahr Umsätze von vielen Millionen Franken abgewickelt werden. Deshalb ist es für uns und unsere Kunden von entscheidender Bedeutung, dass der Wert dieser klassischen Backend-Systeme erhalten bleibt und ständig und kostengünstig an neue Bedingungen, Gesetze, Märkte und Regularien adaptiert werden kann.

## Zu lösendes Problem

Für diesen Werterhalt ist es notwendig, früher angestrebte Produkt-Eigenschaften auf den Prüfstand zu stellen, ggf. neu zu definieren und daraus resultierende architektonische Umbaumaassnahmen abzuleiten.

Für eines unserer ERP-Produkte wollten wir deshalb die Kern-Geschäftslogik aus einer umfangreichen monolithischen Vererbungshierarchie mit einem zusätzlichen Geflecht an Hilfsklassen neu strukturieren. In der bisherigen sequentiellen Code-Struktur war es eine der grössten

Herausforderungen, bei Wartungs- und Erweiterungsarbeiten die jeweils richtigen Codestellen zu finden und sicherzustellen, dass Berechnungsregeln in der richtigen Reihenfolge ausgeführt werden.

## Lösung

Die Lösung wurde in mehreren Schritten vorgenommen:

1. Durch unser Reengineering-Team wurde das Ziel-Design definiert und die Code-Migration zunächst in einem manuellen Prozess begonnen. Wegen der enorm hohen personellen Aufwände, die dazu notwendig sind, haben wir nach Möglichkeiten gesucht, die Code-Migration von der alten in die neue Architektur zu automatisieren. Kai-Uwe Herrmann (Lead Software Architekt, Bison): «Wir wollen endlich den Weg von der Manufaktur ins industrielle Zeitalter der Software-Herstellung gehen.»
2. Gemeinsam mit Object Engineering GmbH haben wir dann einen Proof of Concept durchgeführt, der das Erkennen bestimmter Code-Muster und das Nachverfolgen von Methoden-Aufrufen im Quellcode aufzeigen konnte. – Die wichtigste Voraussetzung für eine teilautomatisierte Migration war damit erfüllt.
3. In einem gemeinsamen Reengineering-Projekt wurden die im PoC Konzept gewonnenen Erkenntnisse verfeinert und die Analyse-Tools entsprechend adaptiert. Dazu wurden gezielt Generatoren erstellt, die die neuen Code-Artefakte erstellten.
4. Der in Schritt 3 generierte Code wurde dann durch unser Entwicklungs-Team in die Codebasis eingebaut, geprüft, angepasst und schliesslich der Legacy-Code gelöscht.

**SNG**

Member of the  
Solution Network Group

## 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](http://www.objeng.ch)  
[marketing@objeng.ch](mailto:marketing@objeng.ch)

# Architektur-Refactoring



## Vorteile dieser Lösung

Mittels OMAN konnte der zu migrierende Code ermittelt, analysiert und daraus neue Klassen für die Zielarchitektur generiert werden.

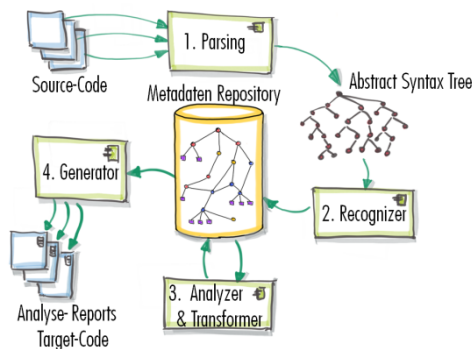
Aufwand und Durchlaufzeit der Code-Migration konnten dank der Automatisierung erheblich gesenkt werden.

Wegen des Generierungsansatzes hatten wir deutlich weniger Fehler bei der Migration, als bei der rein manuellen Umsetzung.

Dank der bereits existierenden OMAN-Plattform konnte Object Engineering die Analyse- und Generator-Bausteine in kurzer Zeit entwickeln, sodass wir schnell davon profitieren konnten.

## Die Reengineering-Plattform

Dieses Verfahren beinhaltet folgende Schritte:



1. **Parsing:** Einlesen der gesamten Code-Basis und Erstellen eines abstrakten Syntax-Baumes (AST) für jede Klasse.
2. **Recognizer:** Durchlaufen des AST mit Erkennung der projekt- und sprachenspezifischen Regeln und Speichern der relevanten Informationen im Meta-Modell.
3. **Analyzer & Transformer:** Durchlaufen des Metadaten-Baumes und Ergänzung zusätzlicher Metainformationen.
4. **Generator:** Durchlaufen des Metadaten-Baumes und Generierung von Reports und neuen Artefakten.

Dank der auf unsere Applikation zugeschnittenen Analyse- und Generator-Bausteine, die in die OMAN-Reengineering-Plattform integriert wurden, beginnt man beim Generator-Bau nicht bei Null. Hinzu kommt die Expertise von Object Engineering, die auf eine lange Historie von Reengineering-Projekten zurückblicken und von deren Beratung wir in diesem Projekt profitieren konnten.

## Einsparungen

Interessant ist ein solches innovatives Automatisierungs-Vorhaben dann, wenn sehr viele ähnliche «Teile» erkannt und daraus sehr viele neue, wiederum ähnliche «Teile» produziert bzw. erzeugt werden müssen, denn natürlich investieren wir nicht ohne betriebswirtschaftliche Erwartungen. Unser Ziel war es, den Aufwand der eigentlichen Code-Migration um 40-50% zu senken. Für dieses erste Projekt haben wir zunächst ein Mengengerüst der zu erstellenden Zielklassen erstellt – ca. 3500. Wenn nun eine solche Zielklasse im Durchschnitt auch nur um wenige Stunden auf die Hälfte reduziert werden kann, dann macht der Multiplikator von 3500 eben ein sehr lohnendes Unterfangen daraus.

Von Beginn an kamen wir auf mehr als 40% Produktivitätsgewinn für die Code-Migration und konnten diesen Prozentsatz später sogar auf über 50% steigern – Aufwand und Durchlaufzeit des Gesamtprojektes konnten dadurch erheblich reduziert werden.

Darüber hinaus sollte nicht unerwähnt bleiben, dass mit dieser Automatisierung eine grosse Arbeit von den Schultern der Entwickler genommen wird. Das akribische, feingranulare Analysieren des Legacy-Codes, um die Code-Teile zu identifizieren, die in eine neue Klasse zusammengefasst werden sollen, entspricht dem Entwickler-Naturell oftmals überhaupt nicht.

SNG

Member of the  
Solution Network Group

## 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](http://www.objeng.ch)

[marketing@objeng.ch](mailto:marketing@objeng.ch)

# Architektur-Refactoring



Dabei muss zusätzlich enorme Denkleistung darauf verwendet werden, dass alle Bedingungen, die zu den relevanten Statements führen, berücksichtigt werden.

Schliesslich müsste der Entwickler auch berücksichtigen, dass die mehrfache Migration derselben Bedingungen und Statements in mehrere neue Klassen vermieden oder zumindest darüber Buch geführt wird, um später ggf. die Wiederverwendung gleicher Code-Teile wieder sicherstellen zu können.

Viel Denken, wenig Kreativität wäre das Resultat. Mit der Automatisierung entfällt ein Grossteil dieser anstrengenden, aber wenig Spass bringenden «Boilerplate»-Arbeit.

## Fazit

Kai-Uwe Herrmann, Lead Software Architekt Bison:

«Wir wollten das bereits gestartete Reengineering-Projekt effizienter gestalten und haben nach Automatisierungsmöglichkeiten gesucht. Bei der Internetrecherche habe ich direkt keine vielversprechenden Ansätze dafür gefunden. Schliesslich stiess ich auf den «Workshop Software Reengineering» in Bad Honnef, zu dem ich mich anmeldete. Ich hoffte, hier Informationen zu bekommen, wie solche Automatisierungen machbar wären. Bei der Konferenz lernte ich Andres Koch von der Object Engineering kennen und hörte seinen Vortrag, der bei mir ins Schwarze traf. Nach meiner Rückkehr habe ich das Thema bei uns intern vorgestellt (unter anderem dem Leiter Entwicklung und dem Chief-PO für die Produktlinie). Begeistert nahm man die Idee auf, und ich konnte das Bison Management für einen Proof of Concept gewinnen. So konnten wir belegen, dass die Idee funktioniert. Während des PoC ist bereits eine Vertrauensbasis mit Object Engineering entstanden, die sich mit Marketing-Slides nicht erklären lässt.

Inzwischen ist die Methode bei Bison etabliert, die Object Engineering hat eine Grundlage geschaffen, auf der sich weitere Generatoren aufsetzen lassen. Sie sind offen für Diskussionen, geben qualifiziertes Feedback, haben Erfahrung, Ideen und sie sind flexibel und schnell in der Umsetzung. Nicht zuletzt vermindert es unsere Entwicklungszeiten ja wirklich signifikant, d.h. der Erfolg gibt ihnen und mir recht.

Unser Management fragt jetzt bei jeder Gelegenheit, ob sich da nicht was automatisieren liesse, und auch wenn das nicht immer möglich ist, so finde ich es doch wichtig und richtig, immer wieder darüber nachzudenken.

Mit Object Engineering haben wir einen Partner, mit dem wir solche Vorhaben umsetzen können. Wir bringen das Knowhow über die Applikation mit, wir haben einen Plan, wie die Ziel-Architektur aussehen soll, wir kennen Mengengerüste und wir wissen, wie der manuelle Migrationsprozess funktioniert und welche Code-Patterns zu erkennen sind, die Transformationsregeln zum neuen Code – und Object Engineering weiss, wie man das automatisiert. »

Marco Persi, Leiter Development Bison:  
«Die Zusammenarbeit mit Object Engineering hat sich für Bison technologisch und betriebswirtschaftlich ausgezahlt. Der Fortschritt respektive die erzielte Beschleunigung in einem technologisch herausfordernden Projekt ist signifikant, weshalb wir uns auf die weitere Zusammenarbeit freuen.»

## Kontakt

Wenn Sie sich für eine unverbindliche Voranalyse für ein System in Ihrem Umfeld interessieren, zögern Sie nicht, uns zu kontaktieren unter:

**Object Engineering GmbH**

**Tel: 044 400 47 00**

**E-Mail: [marketing@objeng.ch](mailto:marketing@objeng.ch)**

**Web: [www.object-engineering.ch](http://www.object-engineering.ch)**

**SNG**

Member of the

Solution Network Group

## 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](http://www.objeng.ch)

[marketing@objeng.ch](mailto:marketing@objeng.ch)