# 5 Genauigkeit von Aufwandsschätzungen in Reengineering-Projekten am Beispiel einer großen Sprachumstellung von Assembler nach COBOL

**Jens Borchers**
CC GmbH Wiesbaden
jens.borchers@caseconsult.com

## 5.1 Einführung

Die SPARDAT, die ausgelagerte IT-Service-Organisation der österreichischen Sparkassen, hat sich im Jahre 2002 entschieden, ihre bestehende Wertpapiertransaktionsverarbeitung technisch grundlegend zu renovieren. Als ein Teil dieser umfassenden Renovierung wurde auch die Umstellung von ca. 400 mehr oder weniger komplexen /370 Assembler-Programmen (unter IBM z/OS) erforderlich. Als Zielsprache wurde dabei COBOL (auf Basis des ANSI-Standards von 1985) definiert. Es soll an dieser Stelle nicht weiter darauf eingegangen werden, warum diese Zielsprache - und nicht modernere gewählt wurde, gute Argumente dafür finden sich aber in z.B. [1]. In [2] sind die üblichen Problemfelder eine Assembler-COBOL-Umstellung dargestellt, der dort ebenfalls propagierte Ansatz unter Nutzung der Spezikationssprache WSL hat sich aber für das vorliegende Projekt aber als nicht anwendbar erwiesen; es wurde mit einem „direkten“ Assembler-COBOL-Converter gearbeitet.

Bei dieser Sprachumstellung handelt es sich also um ein gut definiertes und verhältnismäßig homogenes Reengineering-Projekt, das sich daher auch gut abschätzen lassen sollte. Im weiteren Beitrag wird kurz dargestellt, was überhaupt alles zu schätzen war und wie die bisherigen Erfahrungen im Verhältnis zum tatsächlichen Aufwand stehen. Das Projekt steht kurz vor dem Abschluß, alle Zahlenwerke liegen daher im Moment noch nicht vor, werden aber dann Gegenstand weiterer Untersuchungen sein.

## 5.2 Die Projektaufgabe

Bei dem Reengineering-Projekt handelt es sich um eine reine Sprachumstellung, d.h. alle anderen technischen Randbedingungen wie Dateien, Datenbanken, Oberflächen bleiben unverändert. Einzige weitere Änderung ist die Ersetzung von Aufrufen der alten Assembler-orientierten Zugriffsroutinen durch die entsprechenden für COBOL. Außerdem sind alle kurzen Assembler-Namen durch entsprechende COBOL-Namen zu ersetzen (was nicht so trivial ist, wie es sich auf den ersten Blick vielleicht darstellt).

Der Umfang des Projekts stellt sich wie folgt dar:

- 400 Assembler-Programme mit ca. 400.000 Lines of Code
- weit über 1.000 genutzte Macros und Copybooks, über 1 Million LoC

Wie bereits oben beschrieben, werden im Rahmen des Projekts ausschließlich technische Anpassungen durchgeführt, d.h. die betriebswirtschaftliche Funktionalität bleibt unverändert. Dieser Nachweis ist durch entsprechende Regressionstests zu führen. Da es sich um eine Anwendung im Bankenbereich handelt, in der praktisch jedes zweite Feld einen Geldwert repräsentiert, sind die Anforderungen an die Testabdeckung entsprechend hoch. In den Originalprogramme erkannte Fehler, die (nur) im Rahmen jeder Umstellung ans Licht kommen, wurden gesondert behandelt.

## 5.3 Die Projektabwicklung

Das Projekt wurde nach dem von CC entwickelten „Factory“-Ansatz für Reengineering-Projekte abgewickelt, und zwar mit Einsatz von Offshore-Ressourcen. Dieser Ansatz ist ausführlich in [3]und [4] dargestellt und hat sich seitdem mehrfach bewährt und wurde weiter optimiert.

Dabei sind folgenden Haupt-Phasen und -Aufgaben - auch im Sinne einer Aufwandsschätzung relevant:

- Setup-Phase
  - Aufbau der Projektinfrastruktur (Konfigurations-Management, Testumgebungen)
  - Erstellen des Umstellungs-"Kochbuchs", Anpassen der Conversion Tools
- Umstellung eines ersten Pakets als Pilot, Optimierung der Abläufe und Regelwerke
- Paketorientierte Umstellung der Anwendungsprogramme, insgesamt wurde 7 Pakete (+Pilot) definiert
  - Eigentliche maschinelle und manuelle Konversion der Programme und zugehörigen Komponenten
  - Durchführung von Referenzläufen mit Original-Programmen (ungenau immer als Referenz-Tests"bezeichnet) zur Gewinnung von Testdaten
  - Durchführung der Regressions-Tests, bis zum Nachweis der identischen Funktionalität
  - Statische und dynamische (primär Testabdeckungsgrad) Qualitätsmessungen
- Übergabe der neuen Programme g zu normalen Integrationstests und Produktionsaufnahme

## 5.4 Die Projektschätzansätze

Für die Schätzung von Software-Reengineering-Projekten gibt es diverse Ansätze, die primär auf dem Umfang und der Komplexität der zu bearbeitenden Komponenten basieren (vgl. z.B. [5]). Prominentester Vertreter ist ein abgewandeltes COCOMO II-Verfahren, welches auch von

Sneed beschrieben wurde und bei dem die Zahl der Einflußparameter von den ursprünglich 100 auf 12 reduziert wurde. Trotzdem können auch diese 12 Faktoren die Schätzungen noch signifikante Größenordnungen beeinflussen.

Es ist deshalb in [6] vorgeschlagen worden, während der Durchführung eines Reengineering-Projekts nach jedem abgeschlossenen Arbeitspaket eine Rekalibrierung des Schätzansatzes vorzunehmen. Dieses hat sich auch in diesem Projekt als sinnvoller Ansatz erwiesen, wenngleich es nach unserer Einschätzung nicht mit dem dort vorgeschlagenen und verhältnismäßig aufwendigen Verfahren passieren muß.

Neben den Schätzverfahren, die auf Eigenschaften der umzustellenden Komponenten abheben, gibt es natürlich immer den rein heuristischen Ansatz, bei dem auf der Basis von Erfahrungswerten mit vereinfachten Formeln gerechnet werden kann. Eine entsprechende Erfahrungsbasis vorausgesetzt, kann auch dieser Ansatz zu Schätzungen führen, die nicht wesentlich ungenauer als die oben angesprochenen sind.

Es ist außerdem festzustellen, daß die komponentenorientierten Schätzungen andere wesentliche Blöcke aus dem im vorhergehenden Abschnitt beschriebenen Hauptaufgaben gar nicht berücksichtigen können. So ist z.B. die Erstellung von Referenzdaten in der Praxis von ganz anderen Parametern abhängig als man sie sich z.B. vorstellen könnte (z.B. Zahl der in einer Komponente genutzten Ein-/Ausgabebestände). Hier haben sich „rule of thumbs“ („wir brauchen x Projekttage für ein Batchprogramm und y Projekttage für ein Onlineprogramm“) als völlig ausreichend erwiesen. Sie spiegeln auch keine „Pseudogenauigkeit“ vor wie andere Ansätze.

Der Aufwand für die Regressionstests kann im allgemeinen proportional zum Aufwands für die eigentliche Komponentenbearbeitung geschätzt werden. Detailliertere Schätzansätze (z.B. mit Berücksichtigung der Logik-Komplexität der Zielprogramme) erscheinen nur vordergründig genauer.

Im vorliegenden Projekt wurde folgende Schätzansätze kombiniert:

- Eine detailliertes, toolbasiertes Assessment der Assemblerprogramme, um eine Einordnung in Komplexitätsklassen zu ermöglichen
- Heuristische Schätzung auf Basis der Anzahl, des Umfangs und der Komplexitätsklasse der umzustellenden Komponenten; hier konnte das Projekt auch aus der Erfahrung von 4 vorhergehenden Umstellungen gleicher oder zumindest sehr ähnlicher Art (alle Assembler nach COBOL) schöpfen
- Abschätzung von Referenz- und Regressionstest als abgeleitete Größen der obigen Basisschätzungen
- „Normale“ Schätzansätze für alle Aktivitäten, die primär von der Projektlaufzeit abhängig sind (wie z.B. Problem- und Konfigurationsmanagement) bzw. einmalige Aufgabenblöcke (wie z.B. Setup) darstellen

## 5.5 Die Projekterfahrungen mit der Schätzung

Das Projekt hat eine Gesamtlaufzeit von 16 Monaten, von denen ca. 3 Monate für die Setup-Phase genutzt wurden und ca. 1 Jahr für die eigentliche paketorientierte Umstellung. Im Mittel waren 20 Mitarbeiter mit dem Projekt beschäftigt, die sich auf drei Standorte (Kunde, CC Deutschland, CC India) verteilt haben.

Von Kundenseite wurden primär über die gesamte Projektlaufzeit die Bereiche Referenzdatenerstellung (2 Mitarbeiter + ztw. zusätzliche Experten) und Problembearbeitung (unklare Umsetzungsregel für spezielle Komponenten, Fehler im Originalprogramm etc.) abgewickelt.

Insgesamt haben sich die ursprünglichen Schätzungen bis zum aktuellen Stand des Projekts als sehr gut erwiesen und liegen deutlich unterhalb der Grenzen, die z.B. in [6] als noch tolerierbare Abweichungen genannt werden. Es wurde jeweils nach Abwicklung der einzelnen Pakete eine Rekalibrierung der Produktivität (basierend auf der Umsetzung von jeweils 500 LoC Assembler) vorgenommen. Diese hat erwartungsgemäß die Schätzgenauigkeit für die letzten Pakete verbessert.

Es hat sich gezeigt, daß Erfahrungswerte und darauf basierende vereinfachte Schätzansätze durchaus mit aufwendigeren Ansätzen mithalten können.

Ganz ausschließen kann man Ausreißer nach oben oder unten ohnehin nie, da sich bestimmte Komplexitäten in bestehenden Programmen erst dann zeigen, wenn man sie konkret bearbeitet. Problematisch ist dabei, daß die üblichen Analysetools derartige Problemzonen nicht finden können und sich selbst die Entwickler dieser Programme dieser nicht mehr bewußt sind.

Außerdem beeinflussen andere Projektereignisse wie z.B. eine zunächst nicht geplante Überarbeitung des Conversion Tools oder überraschende Abgänge von erfahrenden Projektmitarbeitern auch die beste Schätzung.

## 5.6 Das weiteres Vorgehen

Wie bereits oben angemerkt, befindet sich das Projekt derzeit in seiner letzten Phase. Während dieses Projekts sind detaillierte Zahlen für alle möglichen Einflußfaktoren und natürlich auch die tatsächlichen Aufwendungen (bis auf Programmebene) erfaßt worden.

Wir werden (im Sinne von CMMI 4) diese statistisch nach allen Gesichtspunkten auswerten und versuchen, eine Korrelation herzustellen. Inwieweit das wirklich gelingt, wir sich zeigen.

## Literatur


[1] Terekhov, C. Verhoef; *The Realities of Language Conversions*, IEEE Software, November/December 2000

[2] M.P. Ward; *The FermaT Assembler Reengineering Workbench*, ICSM 2001, Florenz, 6.-9. November 2001

[3] J. Borchers, *Erfahrungen mit dem Einsatz einer Reengineering Factory in einem großen Umstellungsprojekt*, HMD Nr. 194, März 1997

[4] J. Borchers, *Software Evolution Enabling - IT-Sicherung auf Basis bestehender Systeme*, 4. Workshop Software-Reengineering, Bad Honnef, 29.-30. April 2002

[5] Andrea De Lucia et. al.,*Empirical Analysis of Massive Maintenance Processes,* CSMR 2002, Budapest, März 2002

[6] M.T. Baldessarre, D. Caivano, G. Vissagio; *Software Renewal Estimation Using Dynamic Calibration*, ICSM 2003, Amsterdam, 22.-26. September 2003

ch gesichertes Zahlenmaterial vorgelegt werden kann.