

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

Vortrag am 3. Mai 2004  
beim 6. Workshop für Software-Reengineering  
in Bad Honnef

Jens Borchers  
Case Consult Hamburg  
Alter Fischmarkt 5  
D-20457 Hamburg  
Telefon +49 40 323250-0  
Telefax +49 40 323250-50  
jens.borchers@caseconsult.com



# Das Projektbeispiel

Umstellung des Wertpapiersystems bei der SPARDAT in Wien von IBM/370 Assembler nach COBOL für z/OS

- ❖ *Umfang: ca. 400 Programme mit ca. 400.000 LoC (nur Programme)*
  - *alle aus einem Arbeitsgebiet*
  - *alle von einer (kleinen) Gruppe erstellt und gepflegt*
  - *gute Qualität, vor allem in bezug auf Strukturen*
  - *Macros für Standard-Konstrukte*
  - *Datendefinitionen weitestgehend in Copybooks*
- ❖ *Aufgaben:*
  - *Ersetzen alle Assembler-Konstrukte durch logisch äquivalente COBOL-Konstrukte*
  - **Keine** *rein technische 1:1-Umsetzung, sondern eher automatisiertes Redevlopment auf Basis von maschinell erzeugten Basis-COBOL-Sourcen*
  - **Aber:** *Betriebswirtschaft - wie immer - absolut 1:1 !!*
  - *Zusatzaufgaben: u.a. Umstellung von Zugriffsroutinen*



## „Einschub“ - Warum nicht ...?

„Warum Umstellung von Assembler nach COBOL...

- ❖ *und nicht gleich Java ??“*

Antworten:

- ❖ *Warum denn? Wo bringt denn Java hier Vorteile?*
  - *„Keiner will mehr COBOL machen“ gilt - zumindest außerhalb Deutschlands - bekanntlich schon lange nicht mehr!*
- ❖ *SPARDAT hat im Mainframe-Bereich nur noch COBOL*
  - *und weiß auch, warum das so ist - und bleibt!*
- ❖ *Abstand der Sprachen schon so deutlich genug*
  - *Optimisten sei „The Realities of Language Conversions“ von Terekhov/Verhoef empfohlen*
  - *Datentypen sind Hauptproblembereich*

„Geht doch automatisch?“

- ❖ *Ansätze von Ward et. al. (FermaT Workbench) hören sich gut an, sind für Echtssysteme, vor allem im betriebswirtschaftlichen Bereich, aber nicht brauchbar*



# Das Projektvorgehen und die zu schätzenden Aufgabenblöcke

Das Vorgehen entspricht dem üblichen CC „RE-Factory-Ansatz“

- ❖ *Setup-Phase*
  - *Projektinfrastruktur, insbesondere*
    - *Konfigurations-Management*
    - *Testumgebungen*
  - *„Umstellungs“-Kochbuch*
  - *Conversion-Tools anpassen*
- ❖ *Pilotpaket, inkl. Optimieren der Abläufe und Regeln*
- ❖ *Paketorientierte Bearbeitung des Umstellungsstoffes*
  - *Konversion der Source-Komponenten*
  - *Durchführung der Referenzläufe (mit Originalprogrammen)*
  - *Durchführung der Regressionstestläufe, bis 1:1-Funktionalität nachgewiesen ist*
  - *Messen der Qualität (statisch und dynamisch)*
    - *für Assembler-Programme*
    - *für COBOL-Programme*

CC RE-Aufwandsschätzungen



# Wie schätzt man ein solches Projekt?

- ❖ *Rein heuristisch*
  - *CC hatte immerhin schon mehrere Projekte - auch genau diesen Typs (also Assembler nach COBOL) - durchgeführt*
  - *Es gab eine gute Erfahrungsbasis für RE-Projekte insgesamt*
  - *Die Ausgangssituation dieses Projekts war verhältnismäßig homogen und guter Qualität*
- ❖ *Anwendung von metrikbasierten Schätzansätzen*
  - *z.B. COCOMO II auf Basis der LoC und vieler (!) Einflußfaktoren*
- ❖ *Permanente Rekalibrierung der Schätzergebnisse durch die Erfahrung des schon bearbeiteten Projektteils*

## ABER

- ❖ *Alle metrikbasierten Ansätze „streuen“ gewaltig (COCOMO z.B. bis 100% über die vielen Einflußfaktoren)*
- ❖ *Wie weit sind Erfahrungen wirklich übertragbar?*
- ❖ *Der Kunde will aber die Kosten-/Terminsicherheit, BEVOR es losgeht*
  - *Wer ist bereit, wie viel Risiko zu übernehmen?*



# Schätzvorgehen im konkreten Projekt

Auf Basis einer Mischung aus unterschiedlichen Ansätzen

- ❖ *Aufwand für die eigentliche Konversion*
  - *Assessment der Assembler-Sourcen*
  - *Grobe Bewertung auf Basis der Größe und „Komplexität“*
- ❖ *Referenztestdatenerstellung*
  - *auch abhängig von ganz anderen Parametern:*
    - *Anzahl der verwendeten I/O-Bestände eines Programms*
    - *Typ des Programms (Batch/Online)*
  - *„Rule of Thumb“ meist ebenso genau wie detaillierte Assessments*
- ❖ *Regressionstest*
  - *Meist als fester Prozentsatz der eigentlichen Konversion*
  - *Testdeckung kann nachträglich zu weiteren Aufwendungen führen (für Referenz- und Regressionstest)*
- ❖ *Basiskosten, die sich durch aus der Notwendigkeit dauernd erforderlicher Ressourcen ergeben*
  - *verhältnismäßig einfach abhängig nur geschätzten Dauern:*
    - *Mitarbeiteranzahl \* Zeitrahmen \* Anteil*

CC RE-Aufwandsschätzungen

© Case Consult 2004  
CC\_AUF\_D-2004-05 006



The Evolution En@bling Experts



# Ein paar konkrete Projektzahlen

## Team-Größen

- ❖ *Kunde*
  - *Erstellen von Referenztestdaten: 2 Mitarbeiter + ggf. temporär weitere*
  - *Beantwortung von Problem Reports: ca. 1 (im Durchschnitt)*
- ❖ *Case Consult vor Ort und in Büro D: durchschnittlich 2 -3 Mitarbeiter*
  - *Setup*
  - *Betreuung Testumgebungen*
  - *Konfigurationsmanagement*
  - *Problem-Bearbeitung mit Kunde und Backoffice*
- ❖ *Case Consult Backoffice (off-shore): bis zu 20 in der Spitze*
  - *eigentliche Source Conversion*
  - *Durchführung der Regressionstests und Debugging*

## Projektlaufzeit

- ❖ *Setup: 3 Monate*
- ❖ *Eigentliche Umstellung: 13 Monate*

CC RE-Aufwandsschätzungen

© Case Consult 2004  
CC\_AUF\_D-2004-05 007



The Evolution En@bling Experts



# Erste Einschätzung von Schätzung versus Ist-Aufwand

- ❖ *Der beste Schätzansatz für die eigentliche Konversion war anscheinend immer noch derjenige auf purer LoC-Basis*
  - *Produktivität (auf Basis von 500 LoC Assembler) war ziemlich stabil*
  - *„Ausreißer“ nach oben und unten durch Sonderanforderungen bzw. besonders „geniale“ Konstrukte (die man vorher nie erkennt bzw. erfährt!)*
  - *Rekalibrierung hat für spätere Pakete bessere Schätzungen gebracht*
  - *Aber: Viele verfälschende Nebeneinflüsse über die Zeit!*
- ❖ *Referenztest läßt sich am besten auf Basis einer „Rule of Thumb“ (x Tage für Batchprogramm, y Tage für Online-Programm ) schätzen, alles andere ist nur „Pseudo-Genauigkeit“, viele Nebeneinflüsse*
- ❖ *Regressionstest-Aufwand als fester Faktor im Verhältnis zur Konversion am stabilsten abschätzbar und einigermaßen verlässlich*





# Weiteres Vorgehen und „Einladung“

- ❖ *Derzeit werden die Zahlen für die letzten Pakete noch abgewartet*
- ❖ *Innerhalb des Projekts sind - im Sinne von CMM Level 4 - diverse Zahlen erfaßt worden:*
  - *Conversion-Produktivität (z.T. bis auf Einzelprogramm-Ebene)*
  - *Verteilung des Aufwands über die Hauptaktivitäten und Basis-Organisationsaufwendungen*
  - *Zahlen über*
    - *Testfälle, Testbestände, Problemmeldungen, Produktionsfehler, Fehler in Originalprogrammen,*
  - *Metrik-Auswertungen über*
    - *alle Assembler-Sourcen*
    - *alle äquivalenten COBOL-Sourcen*
- ❖ *Traut sich jemand, stabile Korrelationen herzustellen?*
  - *CC ist gern bereit, mit interessierten wissenschaftlichen Einrichtungen zusammenzuarbeiten*

