

„Evaluation of Database Design and Reverse Engineering Tools for a Large Software System“



Anne Thomas

TU Dresden
Dr. B. Demuth

Pre Press GmbH (Dresden)
T. Reuter

Gliederung



- Einleitung
- Vorgehensweise
- Kontext für das Reverse Engineering
- Anforderungen an Reverse Engineering
- Firmenspezifisches Reverse Engineering Vorgehensmodell
- CASE Tools Unterstützung
- Zusammenfassung und Ausblick

Einleitung

■ Situation:

- DSTMMP, großes System für die Herstellung von Telefonbüchern
- Entwicklungsstrategie: Anpassung von DSTMMP für jeden Kunden

■ Vorhanden:

- Zahlreiche CASE Tools auf dem Markt
- Softwaretechnologisches Vorgehen: Reverse Engineering

■ Ziel der Arbeit:

- Entwicklung einer Reverse Engineering Methode am Beispiel eines Moduls des Systems
- Untersuchung von adäquaten Werkzeugen

Vorgehensweise

■ Erfassung des Kontext für das Reverse Engineering

- Gewinnung eines Gefühls für den **Problemumfang**
- Erfassung des **IST-Zustands** mittels systematischer Suche aller Softwareartefakte
- Erfassung **innerbetrieblicher Prozesse, Regeln und Strukturen** mittels Interviews und Beobachtungen
- Erfassung von **Mitarbeiter Profilen und Gewohnheiten** mittels einer Fragebogenaktion

■ Erfassung der **Anforderungen** an Reverse Engineering

■ Entwicklung einer **angepaßten Methode**

■ Erprobung der Methode am Beispielsmodul

Kontext für das Reverse Engineering - IST-Zustand

■ Das DSTMMP System

- C++ (Strukturierte Programmierung) / Oracle Anwendung
- Ca. 50MB Code in ca. 1900 Files, innerhalb 19 Directories
- Ca. 450 Tabellen mit 100 Packages, Triggers...

■ Die Dokumente und die Arbeitsteilungen/Profile

Managementdokumente	Projektmanagement
Analyse- und Entwurfsdokumente	Analyse und Entwurf
Entwicklungsdokumente	Entwicklung
Benutzerdokumente	Training und Dokumentation
Testdokumente	Qualitätssicherung

Kontext für das Reverse Engineering - Software Engineering Probleme

- Entwicklungsstrategie: kein neutrales Kernsystem
- Softwarearchitekturprobleme
- Schlecht und wenig kommentierter Code
- Toter Code/tote Kommentare (früherer Anpassungen)
- Nicht einheitliche und konsistente Begriffsbenutzung
- Inkonsistente, nicht aktuelle, redundante Dokumente
- Keine zentrale und einheitliche Dokumentenverwaltung
- Keine zentrale und einheitliche Projektverwaltung
- Dokumente technisch nicht adäquat
- ...

Anforderungen an Reverse Engineering

- Ansatz von objektorientierter Vorgehensweise
- Benutzung von UML für das Modellieren
- Merkmale von der Anwendung betrachten
- Herstellung von neuen Dokumenten aus
 - Modellen
 - Diagrammen (Use Case, Klassen, Sequenz...)
 - Diagramm-Kommentaren und -Beschreibungen
- Format des Modells
- Format der Dokumente (verbindliche)

Firmenspezifisches Reverse Engineering Vorgehensmodell - 1

- Maßgeschneidertes Vorgehensmodell in 9 Schritten und 4 Aktivitäten.
- Einige Aktivitäten sind einmalig durchzuführen (Reverse Engineering)
- Andere Aktivitäten gehören zur Wartung des Modells und zum Roundtrip Engineering
- Aktivitäten von Werkzeugen unterstützen
- Kann nicht voll automatisiert werden!

Firmenspezifisches Reverse Engineering Vorgehensmodell - 2

- **1. Aktivität** (High Level):
 - Schritt 1: Artefakte sammeln
 - Schritt 2: Repository kreieren und füllen
 - Schritt 3: Use Case Sicht herstellen
 - Schritt 4: Deployment Sicht herstellen
- **2. Aktivität** (Low Level):
 - Schritt 5: Physikalische Sicht herstellen (Automatisches Reverse Engineering von Code und Datenbank)
- **3. Aktivität**
 - Schritt 6: Logische Sicht herstellen (Klassendiagramm)
 - Schritt 7: Dynamische Sicht herstellen
- **4. Aktivität:**
 - Schritt 8: Dokumentation herstellen
 - Schritt 9: Wartung des Repositorys

Firmenspezifisches Reverse Engineering Vorgehensmodell - 3

- Vorgehensmodell ist eine Mischung aus
 - Forward Engineering Vorgehen
 - Objekt Orientiertes Analyse und
 - Objekt Orientiertes Entwurf
 - Reverse Engineering Vorgehen
 - Automatisches Reverse Engineering
- Vorgehensmodell ist mit dem Vorgehensmodell der Firma eng verbunden
- Ergebnisse aus Reverse Engineering sind die Grundlage für eine neue Entwicklung

CASE Tools Unterstützung - Darstellung

- Entwicklungsumgebung:
 - SniFF+ (TakeFive Software)
- Metrik Werkzeug:
 - Crocodile (Univ. Cottbus)
- CASE / CARE Werkzeuge:
 - Together C++ Enterprise Edition (Object International Software)
 - Rational Rose Enterprise Edition (Rational)
 - System Architect 2001 (Popkin Software)
- Data Modeller:
 - Designer 2000 (Oracle)
 - Erwin (Platinum) (nicht selber getestet)
- Client/Server application viewer and manager
 - CAST Workbench 2.1 (CAST)

CASE Tools Unterstützung - Auswertung

- Kein bestes Tool
- Viele Tools werden gebraucht: eins pro Aktivität!
- Tools können zu wenig untereinander kommunizieren
- Kein neutrales Austauschformat
- Keine einheitliche Benutzungsoberfläche
- ...

- Tools sind mittels Skripten anpassungsfähig
- Zusätzliche Mitarbeiter nötig!!!

Zusammenfassung und Ausblick

- Reverse Engineering ist kein Standard
- Reverse Engineering ist eine Aktivität in der Vorgehensweise der Firma
- Reverse Engineering ist nicht automatisierbar
- Unterstützung von Werkzeugen nicht ausreichend
- Vorgehensmodell mit alle Dokumente und Aktivitäten notwendig
- Völlige und einheitliche Werkzeugunterstützung (Art Workbench für Firmenleben)