Offering Software Maintenance as an Offshore Service

Keynote Address by
Harry M. Sneed
ANECON GmbH, Wien
Universität Regensburg
Offering Software Maintenance as an Offshore Service

1. What is Software Maintenance
2. The Legacy Software Trap
3. Rationale for outsourcing Maintenance
4. Negotiating a Service Level Agreement
5. Establishing a Maint. Infrastructure
6. Setting up a Maintenance Process
7. Automating the Maintenance Process
8. Servicing Multiple Customers
9. International Competition for Outsourced Software Services
1. What is Software Maintenance

Live Cycle of a Software Product according to N. Zwegintzov
Life Cycle of a Software Product according to Gerish Parikh

1. What is Software Maintenance

Hindu teaching of the Holy Trinity

God of creation

Brahma the developer

God of conservation

Vishnu the maintainer

God of re-use

Shiva the reengineer

Birth

Evolution

Conservation

Death

Reincarnation = Recycling

Redemption
1. What is Software Maintenance

The Software Life Cycle Model

- **Cnception Phase**
  - Birth

- **Development Phase**
  - First Release

- **Evolution Phase**
  - Corrections, changes, Improvements, Enhancements

- **Conservation Phase**
  - Corrections, Changes

- **Redemption Phase**
  - Phase out Displacement

According to Bennett and Rajlik

**Time**
- 5%
- 10%
- 25%
- 55%
- 5%

**Costs**
- 10%
- 20%
- 40%
- 25%
- 5%
1. What is Software Maintenance

Maintenance Activities

according to Lientz and Swanson

- Functional Evolution
- Perfection
- Stabilization
- Change
- Enhancement

Progressive Activities

Conserving Activities
The Legacy Software Trap

• Organization has lost the knowledge contained in the application and young developers do not comprehend the code;
• Organization cannot afford to maintain the existing systems and, at the same time, develop new ones;
• Organization has neither the capacity nor the funds to redevelop the software;
• Scientific community has so far offered no real solution to the problem.
2. The Legacy Software Trap

Software Profile of a Financial Service Provider

- 20% Assembler
- 20% PL/I
- 10% COBOL
- 10% 4GLs
- 10% Java
- 10% C++

48 Million Statements
80 Million LOCs
360 Programmers

ICSM2008
Sneed-7
2. The Legacy Software Trap

Metrics from the Legacy Code Base

- 7,854 PLI Programs
- 13,530 PLI Includes
- 17,151 COBOL Programs
- 24,712 COBOL Copies
- 8,975 Assembler Programs
- 9,597 Assembler Macros
- 3,077 Easytrieve Programs
- 3,353 Pseudo Code Modules
- 3,270 IMS Datenbanken
- 4,416 DB2 Tables
- 31,476 IMS-DC Maps
- 22,090 JCL Procedures
- 2,977 C++ Classes
- 1,977 Java Classes

<table>
<thead>
<tr>
<th>Source Code Quantity Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sources analyzed</td>
</tr>
<tr>
<td>Number of Source Lines in all</td>
</tr>
<tr>
<td>Number of Genuine Code Lines</td>
</tr>
<tr>
<td>Number of Comment Lines</td>
</tr>
<tr>
<td>Number of Modules</td>
</tr>
<tr>
<td>Number of Copy/Includes</td>
</tr>
<tr>
<td>Number of Entry Points</td>
</tr>
<tr>
<td>Number of Exit Points</td>
</tr>
<tr>
<td>Number of Sections/Procedures</td>
</tr>
<tr>
<td>Number of Labels/Paragraphs/Code Blocks</td>
</tr>
<tr>
<td>Number of Reusable Code Blocks</td>
</tr>
<tr>
<td>Number of Data Structures/Objects</td>
</tr>
<tr>
<td>Number of Reusable Data Objects</td>
</tr>
<tr>
<td>Number of Statements</td>
</tr>
<tr>
<td>Number of Macro Statements</td>
</tr>
<tr>
<td>Number of Procedural Statements</td>
</tr>
<tr>
<td>Number of Convertable Statements</td>
</tr>
<tr>
<td>Number of Input Operations</td>
</tr>
<tr>
<td>Number of Output Operations</td>
</tr>
<tr>
<td>Number of Foreign Modules called</td>
</tr>
<tr>
<td>Number of Call Statements</td>
</tr>
<tr>
<td>Number of Perform Statements</td>
</tr>
<tr>
<td>Number of Selections (If &amp; Case)</td>
</tr>
<tr>
<td>Number of Loop Statements (Until/While)</td>
</tr>
<tr>
<td>Number of GOTO Branches</td>
</tr>
<tr>
<td>Number of all Control Statements</td>
</tr>
<tr>
<td>Number of Control Flow Branches</td>
</tr>
<tr>
<td>Number of Literals in Statements</td>
</tr>
<tr>
<td>Number of Constants in Statements</td>
</tr>
<tr>
<td>Number of Test Cases required</td>
</tr>
<tr>
<td>Number of Function-Points</td>
</tr>
</tbody>
</table>
2. The Legacy Software Trap

Legacy COBOL Code of Austrian State Pension Fund

```cobol
009210 VERA.   DGREORG
009220 IF ANTWORT NOT = '1' PERFORM UPDGSEQ THRU EXDGSEQ   DGREORG
009230   GO TO AUS-VERA.   DGREORG
010010 IF ZSTORNO NOT = LOW-VALUE GO TO VERA-DSTNR.   DGREORG
010020 IF EDSTNR NOT = ZERO GO TO VERA-FEHLER.   DGREORG
010030 MOVE ESTORNO TO ZSTORNO.   DGREORG
010040 IF ESTORNO = HIGH-VALUE GO TO VERA-STORNO.   DGREORG
010050 IF EAUSL = SPACE GO TO VERA-AUSG.   DGREORG
   IF EAUSL = '4'
   MOVE 'D1' TO ASART
   MOVE '05' TO AVSTR
   MOVE ZERO TO AUNR1
   MOVE EUNR TO AUNR2
   MOVE EAUSL TO AAUSL
   MOVE EGVTR TO AGVTR
   PERFORM UPHVBAND THRU EXHVBAND
   MOVE SPACE TO EAUSL
   GO TO VERA-AUSG.   DGREORG
012190 VERA-DSTNR.   DGREORG
012200 IF EDSTNR = ZERO GO TO VERA-FEHLER.   DGREORG
012210 IF ZSTORNO = HIGH-VALUE ADD 1 TO RFANZ (5)   DGREORG
012220   GO TO AUS-VERA.   DGREORG
012230 IF ESTORNO = HIGH-VALUE ADD 1 TO RFANZ (5)   DGREORG
012240   GO TO AUS-VERA.   DGREORG
013010 VERA-AUSG.   DGREORG
013020 PERFORM UPDGSEQ THRU EXDGSEQ.   DGREORG
013030 AUS-VERA.   DGREORG
013040 PERFORM UPDGALT THRU EXDGALT.   DGREORG
013050 IF EUNR = ZUNR  GO TO VERA.   DGREORG
013060   GO TO GRBEG.   DGREORG
013070 VERA-FEHLER.   DGREORG
013080 DISPLAY 'ORD = ' EUNR EDSTNR UPON CONSOLE.   DGREORG
013090 STOP 'SATZFOLGE FALSCH PROG. WIRD ABGEBROCHEN'   DGREORG
013100 STOP RUN.   DGREORG
EOF-SICHERN.   DGREORG
```

Legacy Software Trap

The Legacy Software Trap

The Legacy COBOL Code of the Austrian State Pension Fund

ICSM2008

Sneed-9
Status of Information Technology in Germany

According to Chairman of German Software Initiative:

• 70% of all business applications are in a legacy language
• Daily more than 30 billion transactions are made with legacy systems
• In Germany 10 billion Euros are spent per year on the maintenance of legacy systems
• The number of COBOL code lines in Germany is estimated to be 240 billion
• To upgrade this code base would cost at least 960 billion Euros, more than the economy can afford
• There is no alternative to maintaining this code.
### Programmer Distribution according to Capers Jones

<table>
<thead>
<tr>
<th>Year</th>
<th>Programmers in Development</th>
<th>Programmers in Maintenance</th>
<th>% in Maint.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950</td>
<td>90</td>
<td>10</td>
<td>10%</td>
</tr>
<tr>
<td>1960</td>
<td>8,500</td>
<td>1,500</td>
<td>13%</td>
</tr>
<tr>
<td>1970</td>
<td>65,000</td>
<td>35,000</td>
<td>38%</td>
</tr>
<tr>
<td>1980</td>
<td>1,200,000</td>
<td>800,000</td>
<td>40%</td>
</tr>
<tr>
<td>1990</td>
<td>3,000,000</td>
<td>4,000,000</td>
<td>57%</td>
</tr>
<tr>
<td>2000</td>
<td>4,000,000</td>
<td>6,000,000</td>
<td>60%</td>
</tr>
<tr>
<td>2010</td>
<td>5,000,000</td>
<td>9,000,000</td>
<td>64%</td>
</tr>
<tr>
<td>2020</td>
<td>7,000,000</td>
<td>14,000,000</td>
<td>67%</td>
</tr>
</tbody>
</table>
Alternate Solutions to the Maintenance Problem

- Standard Software (ERP)
- Renovation / Reengineering
- Redevelopment
- Outsourcing the Maintenance Operation
- Migration and Wrapping do not solve the Maintenance Problem, they only compound it.
3. Rationale for outsourcing Maintenance

User Department → User Department → User Department → User Department → User Department

Change Requests → Endless Queue Bottleneck

IT-Department

Adding new functions → Changing existing functions

Set of disgruntled old Assembler-, PL/I-, COBOL Programmers

Legacy Systems

centralistic, host-oriented, unflexible Systems

Enterprise data

Overloaded Databases

outdated, complex, monolithic, unstructured, intertwined Programs

Complex, hierarchical, interwoven Data structures
3. Rationale for outsourcing Maintenance

Outsourcing as a long range Solution

User
IT
Operations

Rights
Service Level Agreement
Obligations
Outsourced
Software
Maintenance
Operation
4. Negotiating a Service Level Agreement

Change Management

Request for Change Registration, RfC ID

Estimation
Influence, Consequences Resources, Costs

Authorized Change

Implementation Tests und Quality Assurance necessary

Review after Implementation Expectations satisfied?

Release Management

Release Building and SW Distribution HW Versions with Documentation

Configuration Management

Reports and Configuration Audits for Environment Check

Reports about affected CIs, Documents and Roles

Update of CM Records

Release of SW from DSL UND Update DSL and CM Records

Check of all CM Records after Update

CMDB

Definitive Software Library (DSL)

End

Defining Maintenance Services
Points covered in a Maintenance SLA

- Release Intervals (3, 6, 12 months)
- Error correction times (8, 24, 48 hours)
- System recovery times (15, 30, 60 minutes)
- System availability (22 hours a day)
- Maximum error rates (3 errors per kilo Trans.)
- Charging change requests (by impactdomain)
- Charging additional functionality (by FCPT)
- Help Desk Availability (18 hours per day)
5. Establishing a Maintenance Infrastructure

Software Maintenance Organisation

Common Workflow Components
- Product A
- Product B
- Product C

Common Service Components

Application Areas
- APL A
- APL B
- APL C

User IT Operations

Maintenance Orders
- New Reqs
- Change Reqs
- Error Reports

Projects
- Project A
- Project B
- Project C

Software Architecture
5. Establishing a Maintenance Infrastructure

Structure of a Software Maintenance Shop
Significance of the Software Architecture for Software Maintenance

- Division in application components and technical components
- Framework for communication and data storage
- Separation of data access logic from application logic
- Separation of user interfaces from application logic
- Hierarchical structure of application components
- Collection of common functions in a service level
- Separation of global and local functions
- Implementation of self contained components with a minimum of dependence on other components
- Restricted use of inheritance
6. Setting up a Maintenance Process

Software Maintenance Process

- Problem Reports
- Change Requests
- New Requirements
- Code Analysis
- Software Metrics
- Release Planning
- Release Specification
- Code Documentation
- Architectural Evolution
- Release Implementation
- Data
- Test Cases
- Regression Test
- Release
- Regression Test
- Release Rollout
6. Setting up a Maintenance Process

Impact Analysis of Change Requests
6. Setting up a Maintenance Process

Extracting the impacted Code
7. Automating the Maintenance Process

Software Maintenance Workbench

Main-WS

CM

Software Repository

Documentation Tools

Regression Test Tools

Reengineering Tools

Environment

Off-Loaded Maintenance

Main-WS

Main-WS

Main-WS

Main-WS

Main-WS
7. Automating the Maintenance Process

Contents of a Software Maintenance Repository

- Request Analysis
- Analysis
- Effort Estimation
- Maintenance Requests
- Design
- Techn. Spec.
- Retest
- Test Protocols
- Funktional Spec.
- Software Analysis
- Impact Documents
- Maintenance Statistics
- Acceptance
- Acceptance Reports

Sneed-24
ICSM2008
Dash Board for monitoring Software Status

7. Automating the Maintenance Process

- System Availability: 90%
- Error Density: 2
- Change Rate: 4%
- Growth Rate: 10%
- System Availability: 95%
- System Size: 1500
- System Quality: 0.5
- System Complexity: 0.5
- System Volatility: 0.1
- System Stability: Soll vs. Ist
- System Functionality: Fcpts

- Actual
- Planned
- Maintenance Costs
- Corrections
- Errors Reports
- Change Status
- Implemented
- New Requirements
- Requirement Status
- Correctly executed Transactions
- Poor, Yellow Area, Red Area
- Good, Green Area, Product Status
- System Functionality: Soll vs. Ist

ICSM2008
Synergie of a Software Maintenance Shop

- Common Hardware Plattform
- Common Software Tools
- Common Pools of Specialists
- Common Technical Infrastructure
- Common Technical Knowledge Base
- Common Help Desk
- Unified Maintenance Process
9. International Competition for Outsourced Software Services

Indian Offshore Vendors grow and grow
Transaction volume with Software and Services
(in Billions of Dollars)
80 Percent of Indian Software and Services are exported.
Development of the Indian IT-Industry from 2004 to 2008
9. International Competition for Outsourced Software Services

Proportion of IT Services of total Exports

Comparison of competing Countries as of 2004
Export percentage of IT and other business oriented services
9. International Competition for Outsourced Software Services

Rating of Offshore Suppliers by Competence

Rating of Offshore Partners by perceived Competence
(210 German companies asked, in Percent)
Only in regard to innovation is India ahead of China
(210 German companies asked, in Percent)
9. International Competition for Outsourced Software Services

World employment market

Development of the number of employees and the global real gross domestic product within the Industry and Service Areas (2006)

<table>
<thead>
<tr>
<th>Year</th>
<th>Industrial Countries</th>
<th>China</th>
<th>India</th>
<th>Eastern Europe</th>
<th>World Economy global gross domestic product in Dollar</th>
</tr>
</thead>
<tbody>
<tr>
<td>1970</td>
<td>265,5</td>
<td>85,5</td>
<td>62,0</td>
<td>30,4</td>
<td>1.2 Trillion</td>
</tr>
<tr>
<td></td>
<td>Still mostly isolated markets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1990</td>
<td>362,6</td>
<td>85,5</td>
<td>62,0</td>
<td>30,4</td>
<td>26.2 Trillion</td>
</tr>
<tr>
<td></td>
<td>Collapse of the Eastern accelerated Globalization, Pressure of Competition foremost from China</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>416,9</td>
<td>412,5</td>
<td>220,6</td>
<td>41,0</td>
<td>44.4 Trillion</td>
</tr>
<tr>
<td></td>
<td>The global employment Markets are in direct competition</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Keynote Conclusion

- Most money is earned by relieving others of their dire problems.
- Software Maintenance has become for many IT users in the old industrialized countries a dire problem. They cannot cope with it.
- Taking over the maintenance for these users will relieve them of their problem and promise a steady flow of income.
- Users will become dependent on the outsourcing partner for the continuity of their IT service.
- At the same time the outsourcing partner will be able to absorb the knowledge encapsulated in the software.
- The user will be freed of the burden of maintenance and the outsourcing partner will become rich and knowlegable.
- Today’s Servants will become tomorrow’s Masters.