Value-Based Feedback in Software/IT Systems

Barry Boehm, USC
FEAST/ProSim/SOCE 2000 Keynote
July 12, 2000

boehm@sunset.usc.edu
http://sunset.usc.edu
Outline

• COCOMO II 4-Cycle feedback model
• Value-based feedback approach
  – DMR Benefits Realization Approach (BRA)
  – Implementing DMR-BRA via MBASE
    • Model-Based (System) Architecting and Software Engineering
• Example application: Y2K
  – Relation to Laws of Evolution anomalies
• Conclusions
COCOMO II Overview

• Re-engineering of 1981 Constructive Cost Model (COCOMO)
  – Focused on 2000’s software projects
  – New approaches to processes, scaling, reuse

• Calibrated to 161 project data points
  – Estimates within 30% of actuals, 75% of time
  – 80% of time when calibrated to organization

• COCOMO II book due 1 August 2000 (Prentice Hall)
  – Includes CD with Freeware model
  – Demo versions of 3 commercial versions
COCOMO II Long Term Vision

System objectives: fcn’y, perf., quality

Corporate parameters: tools, processes, reuse

COCOMO 2.0

Rescope

Ok?

Cost, Sched, Risks

Yes

No
COCOMO II Long Term Vision

System objectives: fn'y, perf., quality

Corporate parameters: tools, processes, reuse

COCOMO 2.0

Rescope

Execute project to next Milestone

Ok?

Cost, Sched, Risks

Milestone plans, resources

Milestone expectations

Revise Milestones, Plans, Resources

Ok?

M/S Results

Revised Expectations

Done?

End

Ok?

Yes

Yes

No

No

Yes

No

Yes

End
COCOMO II Long Term Vision

System objectives: fcn’y, perf., quality

Corporate parameters: tools, processes, reuse

COCOMO 2.0

Rescope

No

Execute project to next Milestone

Ok?

Yes

Milestone plans, resources

M/S Results

Ok?

No

Yes

Revised Expectations

Yes

End

No

Revise Milestones, Plans, Resources

Accumulate COCOMO 2.0 calibration data

Recalibrate COCOMO 2.0

Ok?

Cost, Sched, Risks

Yes

Milestone expectations

Yes

No

Done?
COCOMO II Long Term Vision

System objectives: fcn’y, perf., quality

Corporate parameters: tools, processes, reuse

Evaluate Corporate SW Improvement Strategies

Recalibrate COCOMO 2.0

Accumulate COCOMO 2.0 calibration data

Rescope

Cost, Sched, Risks

Yes

Execute project to next Milestone

M/S Results

Ok?

Milestone plans, resources

No

Revise Milestones, Plans, Resources

Milestone expectations

Yes

Done?

End

No

Yes

Revised Expectations

Improved Corporate Parameters

Cost, Sched, Quality drivers
Earned Value System

- Budgeted Cost of Work Scheduled
- Cost
- Time
- Specs
- Plans
- Analyses
- Prototypes

7/12/00 ©USC-CSE
Earned Value System

Cost

Budgeted Cost of Work Scheduled

Specs

Plans

Analyses

Prototypes

Budgeted Cost of Work Performed

Project Expenditures

Time
Earned Value Feedback Process

- Develop/update plans, BCWS
- Perform to plans
- BCWP ≥ BCWS?
  - Yes: BCWP ≥ cost?
  - No: Determine corrective actions

- BCWS: Budgeted Cost of Work Scheduled
- BCWP: Budgeted Cost of Work Performed
Outline

• COCOMO II 4-Cycle feedback model

• Value-based feedback approach
  – DMR Benefits Realization Approach (BRA)
  – Implementing DMR-BRA via MBASE
    • Model-Based (System) Architecting and Software Engineering

• Example application: Y2K
  – Relation to Laws of Evolution anomalies

• Conclusions
“Earned Value” Tracks Cost, Not Value

- BCWS: Budgeted Cost of Work Scheduled
- BCWP: Budgeted Cost of Work Performed
DMR/BRA Results Chain

Order to delivery time is an important buying criterion

INITIATIVE
Implement a new order entry system

CONTRIBUTION
Reduce time to process order

OUTCOME
Reduced order processing cycle (intermediate outcome)

CONTRIBUTION
Reduce time to deliver product

OUTCOME
Increased sales
Benefits Realization Feedback Process

1. Develop/update business case; time-phased cost, benefit flows; plans
2. Perform to plans
3. Benefits being realized?
   - Yes
   - No
4. Assumptions still valid?
   - Yes
   - No
5. Determine corrective actions
Implementing BRA via MBASE

• Use WinWin Spiral Model
  – Negotiated win conditions become goals and plans
  – Situation- and risk-driven process and product plans
  – Risk-, benefit-, and plan-driven feedback control

• Avoid model clashes among success, product, process, property models
  – Via MBASE integration framework, process framework

• Use life cycle commitment anchor points
  – Feasibility validation and stakeholder commitment

• Use electronic process and product guides and templates
  – Web-based, hyperlinked, tailorible
Spiral Model Refinements

- Where do objectives, constraints, alternatives come from?
  - Win Win extensions
- Lack of intermediate milestones
  - Anchor Points: LCO, LCA, IOC
  - Concurrent-engineering spirals between anchor points
- Need to avoid model clashes, provide more specific guidance
  - MBASE

The WinWin Spiral Model

1. Identify next-level Stakeholders
2. Identify Stakeholders' win conditions
3. Reconcile win conditions. Establish next level objectives, constraints, alternatives
4. Evaluate product and process alternatives. Resolve Risks
5. Define next level of product and process - including partitions
6. Validate product and process definitions
7. Review, commitment
WinWin Negotiation Model

Win Condition

Involves

Issue

Addresses

Agreement

covers

Option

adopts

WinWin Equilibrium State
- All Win Conditions covered by Agreements
- No outstanding Issues
Team builds a list of win conditions and organizes win conditions into buckets.
Success Model-Clash Profiles: General

**Users**
- Many features
- Changeable requirements
- Applications compatibility & control
- High levels of Service
- Voice in acquisition
- Flexible contract

**Acquirers**
- Mission cost/effectiveness
- Limited budget, Schedule
- Government standards compliance
- Political correctness
- Development visibility & control
- Rigorous contract

**Maintainers**
- Ease of transition
- Ease of maintenance
- Applications compatibility & control
- Voice in acquisition

**Developers**
- Flexible contract
- Ease of meeting budget & schedule
- Stable requirements
- Freedom of choice: process
- Freedom of choice: team
- Freedom of choice: COTS/reuse

*PC: Process  PD: Product  PP: Property  S: Success*
MBASE Integration Framework

Process models

Product models

Property models

Success models

Life cycle anchor points
Risk management
Key practices

Business case
IKIWISI
Stakeholder win-win

Process entry/exit criteria

Planning and control

Milestone content

Evaluation and analysis

Cost
Schedule
Performance
Reliability

Product evaluation criteria

Domain model
Requirements
Architecture
Code
Documentation
Product Line Domain Scope a Function of ROI, Scope of Empowered PL Manager

Return on Investment (ROI)

Breadth of Domain

too few instances to generate payoff

too general to be competitive

Scope of empowered PLM
MBASE Process Framework

Stakeholders

- Determine the relevance of
  - Stakeholders

Success Models

- Enables satisficing among
  - Success Models

Property Models

- Provide evaluations for
  - Property Models

Domain/Environment Models

- Set context for
  - Domain/Environment Models

Conceptual Product Models

- Are refinements of
  - Conceptual Product Models

- Enable satisficing among
  - Conceptual Product Models

IPM_1

- Intermediate... Product Models

IPM_n

- Reify...

Reified Product Models

- Serve and satisfy
  - Reified Product Models

WinWin Spiral Process

Life Cycle

Architecture Package

Plan in LCA Package
### Success Models Drive Other Model Choices

<table>
<thead>
<tr>
<th>Success Model</th>
<th>Demo agent-based E-commerce system at COMDEX in 9 months</th>
<th>Safe air traffic control system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Stakeholders</td>
<td>Entrepreneurs, venture capitalists, customers</td>
<td>Controllers, Govt. agencies, developers</td>
</tr>
<tr>
<td>Key Property Models</td>
<td>Schedule estimation</td>
<td>Safety models</td>
</tr>
<tr>
<td>Process Model</td>
<td>Design-to-schedule</td>
<td>Initial spiral to risk-manage COTS, etc.; Final waterfall to verify safety provisions</td>
</tr>
<tr>
<td>Product Model</td>
<td>Domain constrained by schedule; architected for ease in dropping features to meet schedule</td>
<td>Architected for fault tolerance, ease of safety verification</td>
</tr>
</tbody>
</table>
RAD Analysis via Process Simulation

• Cost: task savings basically map 1:1 into project savings

• Schedule: task savings map 1:1 into project savings only while task is on critical path
  – Complicating factors: scale, dynamism, interdependent tasks
  – System dynamics an attractive analysis approach
Example System Dynamics Analysis (Madachy)

from [Madachy 94]
Life Cycle Anchor Points

• Common System/Software stakeholder commitment points
  – Defined in concert with Government, industry affiliates
  – Coordinated with Rational’s Unified Software Development Process

• Life Cycle Objectives (LCO)
  – Stakeholders’ commitment to support system architecting
  – Like getting engaged

• Life Cycle Architecture (LCA)
  – Stakeholders’ commitment to support full life cycle
  – Like getting married

• Initial Operational Capability (IOC)
  – Stakeholders’ commitment to support operations
  – Like having your first child
# Win Win Spiral Anchor Points

(Risk-driven level of detail for each element)

<table>
<thead>
<tr>
<th>Milestone Element</th>
<th>Life Cycle Objectives (LCO)</th>
<th>Life Cycle Architecture (LCA)</th>
</tr>
</thead>
</table>
| Definition of Operational Concept | • Top-level system objectives and scope  
- System boundary  
- Environment parameters and assumptions  
- Evolution parameters  
- Operational concept  
- Operations and maintenance scenarios and parameters  
- Organizational life-cycle responsibilities (stakeholders) | • Elaboration of system objectives and scope of increment  
• Elaboration of operational concept by increment                                                                                                           |
| System Prototype(s) | • Exercise key usage scenarios  
• Resolve critical risks | • Exercise range of usage scenarios  
• Resolve major outstanding risks                                                                                                                              |
| Definition of System Requirements | • Top-level functions, interfaces, quality attribute levels, including:  
- Growth vectors and priorities  
- Prototypes  
- Stakeholders’ concurrence on essentials | • Elaboration of functions, interfaces, quality attributes, and prototypes by increment  
- Identification of TBD’s (to-be-determined items)  
• Stakeholders’ concurrence on their priority concerns                                                                                                      |
| Definition of System and Software Architecture | • Top-level definition of at least one feasible architecture  
- Physical and logical elements and relationships  
- Choices of COTS and reusable software elements  
- Identification of infeasible architecture options | • Choice of architecture and elaboration by increment  
- Physical and logical components, connectors, configurations, constraints  
- COTS, reuse choices  
- Domain-architecture and architectural style choices  
- Architecture evolution parameters                                                                                                                           |
| Definition of Life-Cycle Plan | • Identification of life-cycle stakeholders  
- Users, customers, developers, maintainers, interoperators, general public, others  
- Identification of life-cycle process model  
- Top-level stages, increments  
- Top-level WWWWWHH* by stage | • Elaboration of WWWWWHH* for Initial Operational Capability (IOC)  
- Partial elaboration, identification of key TBD’s for later increments                                                                                     |
| Feasibility Rationale | • Assurance of consistency among elements above  
- via analysis, measurement, prototyping, simulation, etc.  
- Business case analysis for requirements, feasible architectures | • Assurance of consistency among elements above  
• All major risks resolved or covered by risk management plan                                                                                               |

MBASE Electronic Process Guide (1)

[Image of MBASE 577 Process Guide]

Activities

- MBASE 577 Process
  - Inception Phase
  - Risk Analysis
    - Identify Critical Risk
    - Identify Noncritical Risk
  - Develop Prototypes
  - Evaluate Analytic
    - Tailor Workforce
    - Negotiate System Capabilities
    - Consider Product Line Options
    - Define System Boundary and Interfaces
    - Define Custom Systems and Products
  - Access Analysis
    - Identify Stakeholders
    - Identify Primary Use Condition
    - Create Work Item Agreements
    - Develop Business Case
  - Product Analysis
    - Develop System Requirements
    - Identify Visible Architecture
    - Assess and analyze COTS from LCA Package
  - Process Analysis
    - Identify Life Cycle Strategy
    - Identify Top Level Overview

Artifacts

- Engineering Documents
  - Operational Concept Description
    - Domain Description
    - Proposed System
    - Common Definition Language
  - System and Software Requirement
    - Project Requirements
    - Capability Requirements
    - System Interface Requirement
    - Level of Service Requirements
    - Business Requirements
    - Common Definition Language
  - System and Software Architecture
    - Architectural Analyses
    - System Design
    - Common Definition Language
  - Life Cycle Plan
    - Milestone and Products
    - Responsibilities
    - Approvals
    - Responses
  - Feasibility Rationalization Description
    - Cost Benefit Analysis

Agents

- Participating Agent
  - Customer
  - User
  - Domain Expert
  - Performing Agent
    - Project Manager
    - Analyst
    - System Analyst
    - Designer
    - Developer

[Image of MBASE 577 Process Guide]
Integrating DMR/BRA and MBASE

Stakeholders

Provide value realization feedback to

Negotiate mutually satisfactory

Success Models

Provide measures for

Constrain the realization of

Business Value Assumptions

Business Value Outcomes

Contribute to

Business Initiatives

(Process, product)

Constrain the realization of

Contribute to

Business Outcomes

Constrain the realization of

IT Initiatives

(Process, product)

Contribute to

Potential (model) Actual (performance)

Development Assumptions

Contribute to

Business Assumptions

©USC-CSE
Outline

• COCOMO II 4-Cycle feedback model
• Value-based feedback approach
  – DMR Benefits Realization Approach (BRA)
  – Implementing DMR-BRA via MBASE
    • Model-Based (System) Architecting and Software Engineering

• Example application: Y2K
  – Relation to Laws of Evolution anomalies

• Conclusions
BRA-Based Feedback Process: Y2K Example
– Explaining Laws-of-Evolution Anomalies

- Law IV: Level of activity stays relatively constant
Y2K Example Chronology

• 1994-95: Business case analysis: LCO spiral cycles
  – Initial triage: some phaseouts, outsourcing, upstaffing
  – Resulting BRA plans, LCO milestone artifacts

• 1996: Post-Y2K architecture: LCA spiral cycles
  – Pilot projects in riskier areas
  – Feedback control/update to BRA plans

• 1997-99: Construction and transition spiral cycles
  – Incremental development by system
  – Feedback control/update to BRA plans
1994 Business Case Analysis

- Objectives: No serious Y2K problems; portfolio integration; rapid evolvability; cost-effective support of business strategy
  - Business strategy: Focus on core E-box product line; shorten time-to-market; closed loop sales/service/marketing/development
- Alternatives: New/modified/same portfolio architecture; applications phaseout/migration/modification; fix in-house/upstaff/outsource
- Constraints: Y2K aspects transitioned by mid-1999
- Risks: Scope too ambitious for timeframe; obsolescence due to technology/market trends; underestimate Y2K transition complexity
1995 Example Results Chain: Inventory Control (IC)

Operational units committed to common IC framework

Assumption

Integrated IC/mfg./sales support

Initiative

Operational changes implemented; Staff trained, motivated

Assumption

Initiative

Outcome

Conversion of apps. to common 1/2K IC framework

Contribution

Integrated IC apps. portfolio

Y2K-compliant IC system; rapid evolvability

Contribution

Interoperable IC/mfg./sales IT support

Outcome

Reduced ops. costs; Increased profitability
Individual Business Case Analysis: Control of manufacturing tools inventory

• Relational DBMS and 250 KSLOC of COBOL
• Decoupled from product inventory control, sales
• Analyze Net Present Value (NPV) for main alternatives

\[
\text{NPV} = \sum \text{PV(benefit flows)} - \sum \text{PV (cost flows)}
\]

• Alternatives: current staff/upstaff/outsource
• Costs: COCOMO II maintenance model; outsource bids; Y2K noncompliance costs; training costs
• Benefits: Reduced clerical costs; inventory carrying costs
Manufacturing Tools IC Business Case Summary

- Upstaff has best payoff, least Y2K risk
- Implies non-constant activity levels
Conclusions

• Need to embed IT feedback processes within enterprise feedback processes
  – DMR/BRA and MBASE a good way to do this
  – Organizational evolution and process simulation provide valuable analysis frameworks

• Value based feedback process can help explain anomalies in Laws of Evolution
  – Y2K phenomenon representative of other large isolated events
    • Mergers, acquisitions, downsizing, e-commerce