Escaping the Software Tar Pit: Model Clashes and How to Avoid Them

Barry Boehm, USC
GS AW 99
March 4, 1999
(boehm@sunset.usc.edu)
(http://sunset.usc.edu)
MBASE* Outline

• MBASE Overview and Motivation
  – Model clash taxonomy and examples

• MBASE Framework
  – Relations to Win Win Spiral Model, Anchor Points
  – Relations to Objectory, AT&T/Lucent ARB’s
  – Application to Digital Library projects

• Early Adopters
• Conclusions

* MBASE : Model-Based (System) Architecting and Software Engineering
“No scene from prehistory is quite so vivid as that of the mortal struggles of great beasts in the tar pits.

Large system programming has over the past decade been such a tar pit, and many great and powerful beasts have thrashed violently in it.”

Fred Brooks, 1975
“Everyone seems to have been surprised by the stickiness of the problem, and it is hard to discern the nature of it.

But we must try to understand it if we are to solve it.”

Fred Brooks, 1975
Understanding the Tar Pit: Model Clashes

- **Model (Webster):** A description or analogy used to help visualize or analyze something; a pattern of something to be made
  - Includes product models, process models, property models, success models
- **Model Clash:** An incompatibility among the underlying assumptions of a set of models
  - Produces conflicts, confusion, mistrust, frustration, rework, throwaway systems
- **Model Integration:** Choosing and/or reengineering models to reconcile their underlying assumptions.
Examples of Model Clashes

- Product Model Clashes: structure clashes, traceability clashes, architectural style clashes
- COTS-driven product and Waterfall process
- Risk-based process and spec-based progress payments
- Design-to-cost process and tightly-coupled architecture
- Incremental process and Rayleigh-curve staffing model
- Evolutionary development without life-cycle architecture
- Golden Rule and stakeholder win-win
- Spec-based process and IKIWISI success model
  - I’ll know it when I see it
MBASE Integration Framework

Success Models
Win-Win; IKIWISI; Business-Case; Mission Models;...

Process Models
- Life-Cycle
  - Waterfall;
  - Evolutionary;
  - Incremental;
  - WW Spiral
- Anchor Points
- Risk Mgmt.
- Activities
  - CMM KPA’s

Product Models
- Domain
- Artifacts
  - Rqts.
  - Arch.
  - Code
  - Doc’n
- Packaging
  - Embedded
  - Shrink Wrap
  - Turn Key
- Product Line
  ...

Property Models
Cost & Schedule; Performance; Assurance; Usability;...

Entry/Exit Criteria
V&V Criteria
Product Development Milestone Content;
& Evolution Process Planning & Control
Evaluation & Analysis
Product Line Domain Scope a Function of ROI, Scope of Empowered PC Manager

Return on Investment (ROI)

Breadth of Domain

too few instances to generate payoff

too general to be competitive

Scope of empowered PLM
MBASE Conceptual Framework

- **Stakeholders**: enable satisfying among
- **Success Models**: provide evaluations for
- **Property Models**: impose constraints on
- **Conceptual Product Models**: provide parameters for
- **Domain/Environment Models**: determine the relevance of
- **WinWin Spiral Process Models**: set context for
- **Process Models**: guide progress in selecting, and reifying
- **IPM**: serve and satisfy

**Models**
- **IPM**
  - **IPM**: reified
  - **IPM**: intermediate
  - **IPM**: product

**Frameworks**
- **MBASE Conceptual Framework**: stakeholder, success, property, conceptual, domain, environment, winwin, spiral, process, models, IPM, reified, intermediate, product.
### 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>
MBASE Outline

• MBASE Overview and Motivation
  – Model clash taxonomy and examples

• MBASE Framework
  – Relations to Win Win Spiral Model, Anchor Points
  – Relations to Objectory, AT&T/Lucent ARB’s
  – Application to Digital Library projects

• Early Adopters
• Conclusions
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

Win-Win Extensions

Original Spiral
## Elements of Critical Front End Milestones

(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,  
- Business case analysis for requirements, feasible architectures | • Assurance of consistency among elements above  
- etc. All major risks resolved or covered by risk management plan |

### Objectory Management Checkpoints

<table>
<thead>
<tr>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration 1</td>
<td>Iteration 2</td>
<td>Iteration 3</td>
<td>Iteration 4</td>
</tr>
</tbody>
</table>

#### Major Milestones
- LCO
- LCA
- IOC
- Full Release

**Strategic focus on global concerns of the entire software project**

#### Minor Milestones

**Tactical focus on local concerns of current iteration**

#### Status Assessments

**Periodic synchronization of stakeholder expectations**
MBASE Example I - Digital Library Applications

• The Challenge
• MBASE Approach
• 1996-97 Results
• 1997-98 Results
The Challenge

• 15 Digital Library Applications
  – 2 sentence problem statements
  – Librarian clients

• 86 Graduate Students
  – 30% with industry experience
  – Largely unfamiliar with each other, Library ops.

* Develop LCA packages in 11 weeks

• Re-form teams from 30 continuing students

* Develop IOC packages in 12 more weeks
  – Including 1-week beta test
# Digital Manuscript Archive Home Page

## Digital Manuscript Archive Home Pag

### Antiphonarium

<table>
<thead>
<tr>
<th>Title</th>
<th>Antiphonarium</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Catholic Church</td>
</tr>
<tr>
<td>Date</td>
<td>15th Century</td>
</tr>
<tr>
<td>Type</td>
<td>Liturgical &amp; Ritual</td>
</tr>
<tr>
<td>Style</td>
<td>Not Available</td>
</tr>
<tr>
<td>Physical Characteristics</td>
<td>On vellum; red staves with black Gregorian capitals and rubrication. Dimension of leaves is 57cm x 41cm.</td>
</tr>
</tbody>
</table>
MBASE Laboratory

• 15 software engineering projects/year
  - 5-person USC Digital Library applications

• Rapidly developing successful applications
  - Multimedia, virtual assistants, data acquisition

• Integrating models and tools
  - DARPA-EDCS architecture and WinWin tools
  - Rational Rose, Unified Modeling Language

• Rapidly improving artifact integration
  - 1996 integrated specs, plans: 160 pages
  - 1997 integrated specs, plans: 103 pages

• Higher client satisfaction ratings
  - 4.7 vs 4.4 on scale of 5

• Annual refinement of guidelines
Early Adopters

• Rational, Xerox, FAA
• Air Force C2ISR Center
  – Field initial new C2ISR capabilities in 18 months
  – Determine, support common spiral model
  – General Officers’ Offsite Feb. 17-18, 1999
    • LG’s Kadish, Donahue, Martin
    • MG’s Cliver, Hawley, Carlson, Hess
  – Adopt WinWin Spiral Model as baseline
  – Revise draft AFI 63-123, “Evolutionary Acquisition for C2 Systems”
MBASE Conclusions

• Successfully used on Digital Library projects
  – And CCPDS-R MBASE precursor
• Key to reducing cycle time (USC RAD Workshop)
  – Top people and teambuilding
  – Prepositioning assets (people, tools, architectures, components, models)
• Key to mastering increasingly complex systems
• Complementary to, integrates existing partial models
  – CMM’s, J-STD-016, ISO/IEC 12207, Architecture-based models
• Avoids many current model clashes
  – Due to uncoordinated model-element choices (mandated, legacy, default, arbitrary)
References

(Boehm et al. papers available at http://sunset.usc.edu)


More Information

• CS 577a Software Engineering I Home Page
  http://sunset.usc.edu/classes/cs577a/index.html
• CS 577b Software Engineering II Home Page
  http://sunset.usc.edu/classes/cs577b/index.html
  (contains links to current project teams’ home pages)
• USC Chronicle Article ‘Library Gives Real-World Challenges to Student Software Designers’ by Eric Mankin