RAD Opportunity Tree
Perspectives

Barry Boehm, USC
USC-CSE RAD Executive Workshop
March 11, 1998
email: boehm@sunset.usc.edu
http://sunset.usc.edu

Outline

• USC-CSE Welcome
• RAD Context
  – Video
• RAD Opportunity Tree and Strategies
• RAD Issues
  – Biggest Opportunity Areas
• Some Successful RAD Experiences
RAD Motivation

- Earlier ROI
- Market Window
- Technology Half-Life

Biggest Opportunity Areas

- People and teambuilding
- Prepositioning
  - Domain engineering, architecting
  - Reusable everything: plans, specs, class libraries, middleware, tests, manuals
  - Integrated product, process, property, and success models
  - Tools and facilities
  - Preworking asset evolution
Complexity of RAD Improvements

- 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)
RAD Opportunity Tree

- Eliminating Tasks
  - Business process reengineering
  - Reusing assets
  - Applications generation
  - Design-to-schedule

- Reducing time per task
  - Tools and automation
  - Work Streamlining (80-20)
  - Increasing parallelism

- Reducing risks of single-point failures
  - Reducing failures
  - Reducing their effects

- Reducing backtracking
  - Early error elimination
  - Process anchor points
  - Improving process maturity
  - Collaboration technology

- Activity network streamlining
  - Minimizing task dependencies
    - Avoiding high fan-in, fan-out
  - Reducing task variance
  - Removing tasks from critical path

- Increasing effective workweek
  - 24x7 development
  - Nightly builds, testing
  - Weekend warriors

- Better people and incentives

- Transition to learning organization
Design-to-Schedule

“If schedule is your independent variable, then just modulate your functionality to meet schedule.”

- Critical success factors:
  - Prioritized requirements
  - A reasonable “ballpark” schedule estimate
  - Software design for ease of contraction
  - Schedule tracking for midcourse corrections

---

Pareto Analysis of Rework Costs

![Pareto Chart]

- 100% Project B
- 100% Project A

% of Cost to Fix SPRs

Number of Similar Systems

<table>
<thead>
<tr>
<th>Project</th>
<th>Number of Similar Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project A</td>
<td>20</td>
</tr>
<tr>
<td>Project B</td>
<td>60</td>
</tr>
</tbody>
</table>

---
Collaboration and Negotiation Techniques

- Highlighted in 3 of the last 6 ICSE keynote addresses
  - Tom DeMarco, “How the requirements were negotiated is far more important than how the requirements were specified.”
  - Ed Yourdon, “Negotiation is the best way to avoid Death March projects.”
  - Mark Weiser, “Problems with reaching agreement were more critical to my projects’ success than such factors as tools, process maturity, and design methods.”
Reducing Task Variance

- Or, "Where did that week go?"

-- Are these two networks equivalent?

4 Equally likely outcomes

\[EV = \frac{20}{4} = 5 \text{ weeks}\]

Getting Tasks Off the Critical Path

- Decomposition and parallelization
  - Replace Critical Design Review by unit inspections
  - Pre-integrate subsystems
  - Parallelize off-nominal testing
  - Massive beta testing
  - Pre-work unit level acceptance tests

- Pre-positioning facilities, components, tools, experts, data
  - "Overinvesting" on reusable components
“Overinvesting” in Reusable Components

Number of Similar Systems

Cumulative Cost

People and RAD

- Better and fewer people
  - Bright, quick, versatile, adaptable, creative, experienced, focused
- Clear RAD priorities and incentives
- Teambuilding and shared vision
  - All of the stakeholders
- Co-location
- Capitalization
- Learning, metrics, continuous improvement
Outline

- RAD Context
- RAD Opportunity Tree and Strategies
- RAD Issues
  - Biggest Opportunity Areas
- Some Successful RAD Experiences

Biggest Opportunity Areas

- People and teambuilding
- Prepositioning
  - Domain engineering, architecting
  - Reusable everything: plans, specs, class libraries, middleware, tests, manuals
  - Integrated product, process, property, and success models
  - Tools and facilities
  - Preworking asset evolution
Reuse at HP's Queensferry Telecommunication Division

<table>
<thead>
<tr>
<th>Year</th>
<th>86</th>
<th>87</th>
<th>88</th>
<th>89</th>
<th>90</th>
<th>91</th>
<th>92</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to Market (months)</td>
<td>70</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>

- **Non-reuse Project**
- **Reuse project**

Prepositioning Example: TRW CCPDS-R

- **Large IR&D investment**
  - Domain engineering and architecting
  - Infrastructure: UNAS precursor
  - Tools: Rational Ada tools; metrics; documents templates
  - Ada process model
- **People, teambuilding, and incentives**
  - Flowdown of award fee to performers
- **Results**
  - Spectacular 3-week SW Engineering Exercise
  - 1 MLOC within ambitious budget and schedule
RAD with the WinWin Spiral Model

- 11 weeks to architect 15 Library Multimedia archive applications
  - Films/videos, images, manuscripts, urban plans, business reports, etc.
  - Using LCO, LCA anchor point deliverables
  - Using WinWin negotiation system
  - All delivered on schedule; clients highly satisfied

- 11 weeks to develop 6-8 of architected applications
  - Full product-oriented deliverables
  - All delivered on schedule
  - All highly satisfactory to clients except one

- 3-in-1 merge of image archives

---

MISSE Integration Framework

**Success Models**
- Win-Win, IKI/WSI, Business-Case, Mission Models,...

**Process Models**
- Life-Cycle
  - Evolutionary
  - Incremental
  - WW Spiral
  - Anchor Points
  - Risk Mgmt
  - Activities
  - CMM KPA's

**Evaluation Criteria**

**V & V Criteria**

**Product Development**

**A Evolution Process**

**Planning & Control**

**Evaluation & Analysis**

**Property Models**
- Cost & Schedule: Performance: Assurance: Usability,...

*MISSE: Model-Integrated Software System Engineering*
### Elements of Critical Front End Milestones

<table>
<thead>
<tr>
<th>Milestone Element</th>
<th>Life Cycle Objectives (LCO)</th>
<th>Life Cycle Architecture (LCA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition of Operational Concept</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Prototypes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition of System Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition of System and Software Architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition of Life-Cycle Plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feasibility Rationale</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MISSE Model Integration: LCO Stage**

![Diagram showing MISSE Model Integration: LCO Stage](image-url)

**Domain Model**
- WinWin Taxonomy
- Stakeholders, Primary win conditions
- Frequent Risks
- Environment Models
- Basic Concept of Operation
  - WinWin Negotiation Model
  - IKIWIWI Model, Prototypes, Properties Models

**Requirements Description**
- Validate LCO
- Validate Architectural Options
- Update Life Cycle Plan elements

**Updated Concept of Operation**
- Validate LCO Rationale

Biggest Opportunity Areas

* People and teambuilding
* Prepositioning
  - Domain engineering, architecting
  - Reusable everything: plans, specs, class libraries, middleware, tests, manuals
  - Integrated product, process, property, and success models
  - Tools and facilities
  - Preworking asset evolution