A Value-Based Software Product Model

Dr. Ray Madachy
USC Center for Software Engineering
Cost Xpert Group

18th International Forum on COCOMO and Software Cost Modeling
October 22, 2003
Agenda

- Model Backround
- Product Model Overview
- Analysis Results
- Summary
Model Background

• **Purpose**: Support software business decision-making by experimenting with product strategies and development practices.

• **Overview**: System dynamics model relates the interactions between product development investments, software quality practices, market share, license retention, pricing and revenue generation for a commercial software enterprise.
Model Assumptions

• COCOMO Reliability cost driver is a proxy for all quality practices
• Resulting quality will modulate the actual sales relative to the highest potential
• Parameterizations:
  – Initial total market size = $64M annual revenue
    • vendor has 15% of market
    • overall market doubles in 5 years
  – A new 80 KSLOC product release can potentially increase market share by 15%-30% (varied in model runs)
Model Diagram

cumulative investment

investment rate

cash flow

cumulative revenue

ROI

revenue generation rate

average license price

reliability

reliability

cumulative effort

development effort rate

project size

schedule compression

active licenses

new license selling rate

license expiration rate

license expiration fraction

market size multiplier

change in perceived quality

perceived quality

current indicator of quality

delay in adjusting perceptions

potential market share

potential market share rate change

potential market share increase due to new product

market share delay

cumulative revenue

revenue generation rate

average license price

reliability

reliability

cumulative effort

development effort rate

project size

schedule compression
Perception of Quality

- Quality reputation quickly lost and takes much longer to regain
- Modeled as asymmetrical information smoothing via negative feedback loop
- The perception modulates sales and resultant market share.
Sales Impact of Quality

<table>
<thead>
<tr>
<th>Reliability Rating</th>
<th>Percent of Potential Sales Captured Relative to Highest Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>50%</td>
</tr>
<tr>
<td>Nominal</td>
<td>80%</td>
</tr>
<tr>
<td>High</td>
<td>95%</td>
</tr>
<tr>
<td>Very High</td>
<td>100%</td>
</tr>
</tbody>
</table>
Sample Run Output
Determining How Much Reliability is Enough

• Use risk exposure framework to find process optimum
• Vary Reliability across runs
• Assess risk consequences of opposing trends: market delays and bad quality losses
• Sum the costs
• Calculate resulting net revenue

\[ RE = P(L) \times S(L) \]

- loss due to unacceptable quality
- loss due to market delay

Sweet Spot
Sample Experiment Results

- 80 KSLOC, schedule 75% of nominal, 3 year revenue timeframe

<table>
<thead>
<tr>
<th></th>
<th>Reliability Rating</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Nominal</td>
<td>High</td>
<td>Very High</td>
<td></td>
</tr>
<tr>
<td>Effort (Person-months)</td>
<td>479</td>
<td>521</td>
<td>573</td>
<td>656</td>
<td></td>
</tr>
<tr>
<td>Schedule (Months)</td>
<td>17.5</td>
<td>17.9</td>
<td>18.5</td>
<td>19.3</td>
<td></td>
</tr>
<tr>
<td>Cost ($M)</td>
<td>10.6</td>
<td>11.6</td>
<td>12.8</td>
<td>14.8</td>
<td></td>
</tr>
<tr>
<td>Revenue ($M)</td>
<td>$44.4</td>
<td>$56.4</td>
<td>$61.1</td>
<td>$62.0</td>
<td></td>
</tr>
<tr>
<td>Maximum Potential Revenue with Same Timing (if highest quality at same schedule)</td>
<td>$65.6</td>
<td>$64.4</td>
<td>$63.2</td>
<td>$62.0</td>
<td></td>
</tr>
<tr>
<td>Market Delay Cost ($M)</td>
<td>$0.0</td>
<td>$1.2</td>
<td>$2.4</td>
<td>$3.6</td>
<td></td>
</tr>
<tr>
<td>Bad Quality Loss ($M)</td>
<td>$21.2</td>
<td>$8.0</td>
<td>$2.1</td>
<td>$0.0</td>
<td></td>
</tr>
<tr>
<td>Total Cost ($M)</td>
<td>$31.8</td>
<td>$20.8</td>
<td>$17.3</td>
<td>$18.4</td>
<td></td>
</tr>
</tbody>
</table>
Resultant Reliability Sweet Spot

- development cost
- market delay loss
- bad quality loss
- total cost

Cost (Millions)

Software Reliability

- Low
- Nominal
- High
- Very High
Profit Maximization View
Sweet Spot Depends on Time Horizon

![Graph showing profit (in millions) vs. software reliability for different time horizons (2 year, 3 year, 5 year). The graph indicates that profit increases with higher software reliability, but the optimal point (sweet spot) depends on the time horizon.](image)
Other Considerations

- Pricing scheme impacts
- Varying market assumptions
- Impact of new releases that increase (or decrease) quality
- Feedback from growing user base to incorporate new features
Summary

• Decision-making can be improved with information gained from simulation experiments
• Risk exposure is a convenient framework for software decision analysis.
• Commercial process sweet spots with respect to reliability are a balance between market delay losses and quality losses
• Business policies operate within a multi-attribute decision space
• Quality impacts the bottom line