Software Engineering Economics: Cost-Effectiveness Analysis

CS 566 – Software Management and Economics
Lecture 13 (Chapters 10 – 12, Boehm 1981; Boehm 2005)

Ali Afzal Malik
Software Decision Analysis Techniques
Chapters 10-20 in Software Engineering Economics

- Economics – The study of how people make decisions in resource-limited situations
- Macroeconomics
  - Inflation, taxation, balance of payments
- Microeconomics
  - Make-or-buy, pricing, how much to build
- Software economics decisions
  - Make-or-buy: Software Product
  - How many options to build?
  - Which DP architecture to use?
  - How much testing (prototyping, specifiying) is enough?
  - How much software to re-use?
  - Which new features to add first?
Outline of Chapters

- **10-12. Context; TPS II Example; Cost-Effectiveness Analysis**
  - Models, optimization, production functions, economies of scale, decision criteria

- **13-15. Multiple-Goal Decision Analysis I**
  - Net value, marginal analysis, present value, figures of merit.

- **16-18. Multiple-Goal Decision Analysis II**
  - Goals as constraints, constrained optimization, system analysis, unquantifiable goals

- **19-20. Dealing with Uncertainties**
  - Expected value, utility functions, statistical decision theory, value of information
Master Key to Software Engineering Economics
Decision Analysis Techniques

All decision criteria (DCs) convertible to present $?

Yes → Use standard optimization, net value techniques (chapters 10, 13)

No → All Non-$ DCs expressible as constraints?

Yes → Use standard constrained-optimization techniques (chapter 16)

No → All Non-$ DCs expressed as single “benefit” criterion?

Yes → Use cost-benefit (CB) decision making techniques (chapters 11, 12)

No → All Non-$ DCs quantifiable?

Yes → Use figure-of-merit techniques, CB techniques (chapter 15)

No → Use techniques for reconciling non-quantifiable DCs (chapter 18)

Is outcome of decision highly sensitive to assumptions (chapter 17)?

Yes → Find, use less sensitive solution

No → Use present value techniques to convert future $ to present $ (chapter 14)

When $ are a mix of present and future cash flows

When some DCs involve uncertainties

Use statistical decision theory techniques (Chapters 19, 20)

End
Chapters 10-12
Cost-Effectiveness Analysis

- Introduction
- Example: transaction processing system (TPS II)
- Performance models
- Cost-performance models
- Production functions: economies of scale
- Cost-effectiveness decision criteria
- Summary
TPS II Business Context

- Current TPS inadequate for growing workload
  - Travel reservations: air, rail, car, hotel
  - Top performance: 1000 transactions/second (tr/sec)
  - Need 2000 tr/sec soon
  - Need growth to 4000 tr/sec
- COTS server capability can provide over 2000 tr/sec
  - But can’t achieve 4000 tr/sec
- Consider developing your own server software
TPS II System Architecture

Local Concentrators
- Companies
- Travel agents
- About 10/region

Regional Concentrators

1

Financial DB

DB Server

Services DB

Server
TPS II Concept of Operation

- Clients prepare and send travel-itinerary requests to local concentrator
  - Package of air, rail, car, hotel reservation requests
- Local concentrators validate requests and forward them to regional concentrators at server
  - Usually about 10 local concentrators per region
- N Regional concentrators use DB server to develop best-match travel itinerary package
  - Send back to clients via local concentrators
  - Multiprocessor overhead due to resource contention, coordination
# COTS/New Development Cost Tradeoff Analysis

<table>
<thead>
<tr>
<th></th>
<th>COTS $K</th>
<th>New Development $K</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Software</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>– Cost to acquire</td>
<td>100</td>
<td>606</td>
</tr>
<tr>
<td>– Integrate &amp; test</td>
<td>100</td>
<td>Included</td>
</tr>
<tr>
<td>– Run-time licenses</td>
<td>10N</td>
<td>Not applicable</td>
</tr>
<tr>
<td>– 5-year maintenance</td>
<td>10N</td>
<td>151</td>
</tr>
<tr>
<td></td>
<td>200 + 20N</td>
<td>757</td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td>250 + 20N</td>
<td>250+20N</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>450 + 40N</td>
<td>1007 + 20N</td>
</tr>
</tbody>
</table>
Now, we need to address the benefit tradeoffs.
TPS Decision 1
How Many Regional Concentrators in Server?

Performance Parameters

- N, number of processors
- S, processor speed (KOPS/sec)
- P, processor overhead (KOPS/sec)
- M, multiprocessor overhead factor

\[ \text{overhead} = M(N-1) \text{ KOPS/sec} \]

- T, transaction processing time (KOPS/TR)

N = ?
S = 1000
P = 200
M = 80
T = 1.0

Performance (TR/sec)

\[
E(N) = \frac{\text{KOPS/sec available for processing}}{\text{KOPS/TR required per transaction}}
\]

\[
E(N) = \frac{N[S-P-M(N-1)]}{T}
\]
**TPS Performance, E(N)**

\[
E(N) = \frac{\text{KOPS/sec available for processing}}{\text{KOPS/sec required per transaction}}
\]

<table>
<thead>
<tr>
<th>N</th>
<th>E(N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>800</td>
</tr>
<tr>
<td>2</td>
<td>1440</td>
</tr>
<tr>
<td>3</td>
<td>1920</td>
</tr>
<tr>
<td>4</td>
<td>2240</td>
</tr>
<tr>
<td>5</td>
<td>2400</td>
</tr>
<tr>
<td>6</td>
<td>2400</td>
</tr>
<tr>
<td>7</td>
<td>2240</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
1.0 = N \left[ 1000 - 200 - 80(N-1) \right]
\]

\[
= N \left[ 1000 - 200 + 80 - 80N \right]
\]

\[
= 880N - 80N^2
\]

\[
= 80N(11-N)
\]

\[
\frac{dE(N)}{dN} = 880 - 160N^* = 0
\]

\[
160N^* = 880
\]

\[
N^* = 5.5
\]

\[
E(N)^* = 2440
\]
TPS throughput: $E(N)$ versus number of processors, $N$

<table>
<thead>
<tr>
<th>$N$</th>
<th>$E(N)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>800</td>
</tr>
<tr>
<td>2</td>
<td>1440</td>
</tr>
<tr>
<td>3</td>
<td>1920</td>
</tr>
<tr>
<td>4</td>
<td>2240</td>
</tr>
<tr>
<td>5</td>
<td>2400</td>
</tr>
<tr>
<td>6</td>
<td>2400</td>
</tr>
<tr>
<td>7</td>
<td>2240</td>
</tr>
<tr>
<td>8</td>
<td>1960</td>
</tr>
</tbody>
</table>
Production Functions

Achievable output = F (input consumed)
e.g. Output = SLOC; Input = PM

- Assuming only technologically efficient pairs:
  • No higher level of output achievable, using given input
  - PF is nonnegative
  - PF is nondecreasing
Segments of Production Function

Output

Investment | High payoff | Diminishing returns

Input
**TPS Decision 2**  
**Which Operating System?**

<table>
<thead>
<tr>
<th>Option</th>
<th>A (Accept Available OS)</th>
<th>B (Build New OS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost ($k)</td>
<td>450 + 40N</td>
<td>1007 + 20N</td>
</tr>
<tr>
<td>Multiprocessor overhead factor</td>
<td>80</td>
<td>40</td>
</tr>
<tr>
<td>(M)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
O/H = M(N-1)
\]

**Option B:**  
\[
E(N) = \frac{N(1000-200-40(N-1))}{1.0} = 840N - 40N^2 = 40N(21-N)
\]

For \( N = 10 \),  
\[
E(N) = 40(10)(11) = 4400
\]
Cost-Effectiveness Comparison: TPS II Options A, B

Option B

\[ E(N) = 40N(21 - N) \]
\[ C(N) = 1007 + 20N \]
Software Project Economies and Diseconomies of Scale

- **Economies of scale:** factors that make it more efficient to produce large quantities of a product than small quantities
  - e.g. productivity aids (test tools, etc.)
- **Diseconomies of scale:** opposite
  - e.g. M in TPS
  - Human factors: communication overhead, personality conflicts
- The best way to combat diseconomies of scale is to **Reduce the Scale**
  - Prototyping
  - Incremental development
  - Wish-list pruning (avoiding gold-plating)
Software Gold-Plating

- Frequently Gold-Plating
  - Instant response
  - Pinpoint accuracy
  - Unbalanced systems
  - Agents with attitudes
  - Animated displays
  - “everything for everybody”

- Usually Not Gold-Plating
  - Humanized input
  - Humanized output
  - Modularity, info. hiding
  - Measurement, diagnostics

- Sometimes Gold-Plating
  - Highly generalized control, data structures
  - Sophisticated command languages
  - General-purpose utilities
  - Automatic trend analysis
Animated displays
Secondary application functions
User amenities
Main application functions
Tertiary application functions
Natural speech input
Animated displays
Secondary application functions
User amenities
Basic application functions
Data management system
Operating System

Value of software product to organization

Cost of software product

Investment
High-payoff
Diminishing returns

7/7/2008
Modular Transaction Processing System

E(2 x 3) = 2 x E(3) = 3840 vs. 2400

Module 1

Module 2

Processed Transaction out

<table>
<thead>
<tr>
<th>P11</th>
<th>P12</th>
<th>P13</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>P21</th>
<th>P22</th>
<th>P23</th>
</tr>
</thead>
</table>

Trans. in

1

2
Cost-Effectiveness Decision Criteria

1. Maximum available budget
2. Minimum performance requirement
3. Maximum effectiveness/cost ratio
4. Maximum effectiveness – cost difference
5. Return on investment (ROI)
6. Composite alternatives
Max. Budget & Min. Performance

E (tr/sec)

Cost C, $K

4400

2400

450 650 1007 1207

A

B

X

Y

Q

Z

P
Maximum Effectiveness/Cost Ratio

Eff/Cost = 8

Eff/Cost = 3.69
Maximum Effectiveness-Cost Difference

Throughput \( E(\text{TR/sec}) \)

Cost \( C, \text{\$K} \)

- A - Accept available OS
- B - Build new OS

Cost Difference

- 0
- 1000
- 2000
- 3000
- 4000

Throughput

- 500
- 1000
- 1500

Cost

- 2400
- 3193
- 4400

Maximum Throughput

- 2400
- 1750
- 650
- 1207

Maximum Cost

- 4400
- 3193
- 1207
Return on Investment (ROI)

ROI = \frac{E - C}{C}

- E = \text{Expected value}
- C = \text{Cost}

Graph showing the relationship between Cost C ($K) and ROI. Points A and B are marked on the graph with corresponding ROI values of 2.67 and 2.60, respectively.

Cost C ($K):
- 0
- 1
- 2
- 3
- 4
- 5

ROI:
- 0
- 1
- 2
- 3
- 4
- 5
Production Function for TPS Composite Alternative

![Graph showing production function with steps at specific C, SK values: 450, 650, 1007, 1207.]
Summary

- Microeconomic concepts help structure, resolve software decision problems
  - Cost-effectiveness
  - Production functions
  - Economies of scale
  - C-E decision criteria

- No single decision criterion dominates others
  - Each is best for some situations
  - Need to perform sensitivity analysis:
    - Slightly altered situation doesn’t yield bad decision
References