Value-Based Software Engineering: Motivation & Key Practices

CS 566 – Software Management and Economics
Lecture 2 (Boehm 2005; Chapter 9, Selby 2007)

Ali Afzal Malik
Outline

- Motivation and definitions
- Understanding sources of value
- Seven key practices
- Conclusions
- References
Software Testing Business Case

- Vendor proposition
  - Our test data generator will cut your test costs in half
  - We’ll provide it to you for 30% of your test costs
  - After you run all your tests for 50% of your original cost, you are 20% ahead

- Any concerns with vendor proposition?
Software Testing Business Case

- Concerns with vendor proposition
  - Test data generator is value-neutral*
  - Every test case, defect is equally important
  - Usually, 20% of test cases cover 80% of business case

* As are most current software engineering techniques
20% of Features Provide 80% of Value: Focus Testing on These (Bullock, 2000)

% of Value for Correct Customer Billing

Automated test generation tool
- all tests have equal value
Value-Based Testing Provides More Net Value

<table>
<thead>
<tr>
<th>% Tests</th>
<th>Test Data Generator</th>
<th>Value-Based Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost</td>
<td>Value</td>
</tr>
<tr>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>35</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>45</td>
<td>30</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>100</td>
</tr>
</tbody>
</table>

Net Value NV

Value-Based Testing

Test Data Generator

Percent of tests run

(30, 58)

(100, 20)
Motivation for VBSE

- Current SE methods are basically value-neutral
  - Every requirement, use case, object, test case, and defect is equally important
  - Object oriented development is a logic exercise
  - “Earned Value” Systems don’t track business value
  - Separation of concerns: SE’s job is to turn requirements into verified code
  - Ethical concerns separated from daily practices

- Value – neutral SE methods are increasingly risky
  - Software decisions increasingly drive system value
  - Corporate adaptability to change achieved via software decisions
  - System value-domain problems are the chief sources of software project failures
The “Separation of Concerns” Legacy

- “The notion of ‘user’ cannot be precisely defined, and therefore has no place in CS or SE.”
  - Edsger Dijkstra, ICSE 4, 1979

- “Analysis and allocation of the system requirements is not the responsibility of the SE group but is a prerequisite for their work”
  - Mark Paulk at al., SEI Software CMM* v.1.1, 1993

*Capability Maturity Model
I wonder when they’ll give us our requirements?
Why Software Projects Fail

352 companies - 8,000 software projects. Source: The Standish Group, 1995
20% of Fires Cause 80% of Property Loss: Focus Fire Dispatching on These?
Need to Consider Fairness and Ethics: Rawls’ Theory of Justice (Rawls, 1971)

- Fair rules of conduct
- Principles of justice
- Participants and obligations
  - Provider (developer)
  - Buyer (acquire)
  - User(s)
  - Penumbra (general public)
- Negotiate mutually satisfactory (win-win) agreements
Rawls’ Theory of Justice - II

- Fair rules of conduct
  - Negotiation among interested parties
  - Veil of ignorance (about what affects whom)
  - Rationality

- Principles
  - Least Advantaged - don’t increase harm to them
    - Harm = probability x magnitude (~risk exposure)
  - Risking Harm - don’t risk increasing harm
    - Don’t use “low-threat” software in “high-threat” context
  - Publicity test - defensible with honor before an informed public
    - Use for difficult cost-benefit tradeoffs
Penumbra Negotiation Example: Fire Dispatching System

- Dispatch to minimize value of property loss
  - Neglect safety, least-advantaged property owners
- English-only dispatcher service
  - Neglect least-advantaged immigrants
- Minimal recordkeeping
  - Reduced accountability
- Tight budget; design for nominal case
  - Neglect reliability, safety, crisis performance
Key Definitions

- Value (from Latin “valere” – to be worth)
  1. A fair or equivalent in goods, services, or money
  2. The monetary worth of something
  3. Relative worth, utility or importance

- Software validation (also from Latin “valere”)
  - Validation: Are we building the right product?
  - Verification: Are we building the product right?
Conclusions So Far

- Value considerations are software success-critical
- “Success” is a function of key stakeholder values
  - Risky to exclude key stakeholders
- Values vary by stakeholder role
- Non-monetary values are important
  - Fairness, customer satisfaction, trust
- Value-based approach integrates ethics into daily software engineering practice
The Model-Clash Spider Web: MasterNet

Users
- Many features
- Changeable requirements
- Applications compatibility
- High levels of service
- Voice in acquisition
- Flexible contract
- Early availability

Maintainers
- Ease of transition
- Ease of maintenance
- Applications compatibility
- Voice in acquisition

Acquirers
- Mission cost/effectiveness
- Limited development budget, schedule
- Government standards compliance
- Political correctness
- Development visibility and control
- Rigorous contact

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

PC: Process
PD: Product
PP: Property
S: Success

6/18/2008   6/18/2008
LUMS
Maslow Human Need Hierarchy (Maslow, 1954)

- Physiological (Food and Drink)
- Safety and Security
- Belongingness and love
- Esteem and Autonomy
- Self-Actualization
Maslow Need Hierarchy

• Satisfied needs aren’t motivators
• Unsatisfied lower-level needs dominate higher-level needs
• Management implications
  – Create environment and subculture which satisfies lower-level needs
    • Stability
    • Shared values, community
    • Match to special needs
  – Tailor project objectives, structure to participants’ self-actualization priorities
People Self-Actualize in Different Ways

- Becoming a Better Manager
- Becoming a Better Technologist
- Helping Other Developers
- Helping Users
- Making People Happy
- Making People Unhappy
- Doing New Things
- Increasing Professional Stature
Projecting Yourself Into Others’ Win Situations

Counterexample: The Golden Rule

• Do unto others
  .. As you would have others do unto you

• Computer sciences world (compilers, OS, etc.)
  – Users are programmers

• Applications world
  – Users are pilots, doctors, tellers

■ Build computer systems to serve users and operators
  .. Assuming users and operators like to write programs, and know computer science
VBSE – Seven Key Practices

1. Benefits Realization Analysis
2. Stakeholders’ Value Proposition Elicitation and Reconciliation
3. Business Case Analysis
4. Continuous Risk and Opportunity Management
5. Concurrent System and Software Engineering
6. Value-Based Monitoring and Control
7. Change as Opportunity
Benefits Realization Analysis
DMR/BRA* Results Chain

INITIATIVE
- Implement a new order entry system

OUTCOME
- Reduce time to process order
- Reduced order processing cycle (intermediate outcome)

OUTCOME
- Reduce time to deliver product
- Increased sales

ASSUMPTION
- Order to delivery time is an important buying criterion

*DMR Consulting Group’s Benefits Realization Approach
Stakeholders’ Value Proposition Elicitation and Reconciliation
EasyWinWin OnLine Negotiation Steps

Review and Expand Negotiation Topics (Group Outliner)
Jointly review and define the scope of the negotiation. Identify the negotiation topics for your EasyWinWin activity.

Brainstorm Stakeholder Interests (Electronic Brainstorming)
Collect ideas about Win Conditions for your EasyWinWin activity.

Converge on Win Conditions (Categorizer)
Jointly craft and organize a succinct list of win conditions.

Capture Glossary of Terms (Topic Commenter)
Define important terms of the domain.

Prioritize Win Conditions (Alternative Analysis)
Determine the business importance and the ease of implementation of all win conditions. Reveal issues and constraints.

WinWin Tree (Group Outliner)
Identify Issues and Options. Negotiate Agreements.

Organize Negotiation Results (Categorizer)
Categorize the results using the negotiation topics.
Red cells indicate lack of consensus.

Oral discussion of cell graph reveals unshared information, unnoticed assumptions, hidden issues, constraints, etc.

<table>
<thead>
<tr>
<th>Features</th>
<th>Importance</th>
<th>Ease of Implementation</th>
<th>Total</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Application Capabilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1 W2 Integrate banner ads with email and chat</td>
<td>10.00</td>
<td>6.50</td>
<td>16.50</td>
<td>8.25</td>
</tr>
<tr>
<td>2.2 W3 The banner will provide a link to the university</td>
<td>10.00</td>
<td>10.00</td>
<td>20.00</td>
<td>10.00</td>
</tr>
<tr>
<td>2.3 W4 Interface for advertisers to select their sched</td>
<td>8.67</td>
<td>3.00</td>
<td>11.67</td>
<td>5.83</td>
</tr>
<tr>
<td>2.4 W5 Default banner of bookstore if no other events</td>
<td>8.00</td>
<td>10.00</td>
<td>18.00</td>
<td>9.00</td>
</tr>
<tr>
<td>2.5 W6 The site management must have a website which</td>
<td>10.00</td>
<td>10.00</td>
<td>20.00</td>
<td>10.00</td>
</tr>
<tr>
<td>2.6 W7 Different kinds of advertising, including sales</td>
<td>10.00</td>
<td>10.00</td>
<td>20.00</td>
<td>10.00</td>
</tr>
<tr>
<td>2.7 W8 Flexible text on banners</td>
<td>10.00</td>
<td>5.00</td>
<td>15.00</td>
<td>7.50</td>
</tr>
<tr>
<td>2.8 W9 Display address of the bookstore, a map of it a</td>
<td>4.00</td>
<td>7.50</td>
<td>11.50</td>
<td>5.75</td>
</tr>
<tr>
<td>2.9 W10 Ads must be hyperlinked so that users can clic</td>
<td>7.33</td>
<td>6.00</td>
<td>13.33</td>
<td>6.67</td>
</tr>
<tr>
<td>2.10 W11 Link to bookstore site (incl book's prices)</td>
<td>9.33</td>
<td>10.00</td>
<td>19.33</td>
<td>9.67</td>
</tr>
<tr>
<td>2.11 W12 Web statistics tracking to determine number of</td>
<td>8.00</td>
<td>4.00</td>
<td>12.00</td>
<td>6.00</td>
</tr>
<tr>
<td>2.12 W13 Input of banner contents to admin via email</td>
<td>5.50</td>
<td>10.00</td>
<td>15.50</td>
<td>7.75</td>
</tr>
</tbody>
</table>
Business Case Analysis
Example of Business Case Analysis

ROI = (Benefits - Costs) / Costs

Return on Investment

Option A

Option B

Time
Example of Business Case Analysis

ROI = (Benefits - Costs) / Costs

Return on Investment

Option A

Option B - Rapid

Option B

Time
Continuous Risk and Opportunity Management
Is This A Risk?

- We just started integrating the software
  - and we found out that COTS* products A and B just can’t talk to each other
- We’ve got too much tied into A and B to change
- Our best solution is to build wrappers around A and B to get them to talk via CORBA**
- This will take 3 months and $300K
- It will also delay integration and delivery by at least 3 months

*COTS: Commercial off-the-shelf
**CORBA: Common Object Request Broker Architecture
Is This A Risk?

- We just started integrating the software
  - and we found out that COTS* products A and B just can’t talk to each other
- We’ve got too much tied into A and B to change

No, it is a problem
- Being dealt with reactively

Risks involve uncertainties
- And can be dealt with pro-actively
- Earlier, this problem was a risk
Earlier, This Problem Was A Risk

- A and B are our strongest COTS choices
  - But there is some chance that they can’t talk to each other
  - **Probability of loss** $P(L)$
- If we commit to using A and B
  - And we find out in integration that they can’t talk to each other
  - We’ll add more cost and delay delivery by at least 3 months
  - **Size of loss** $S(L)$
- **We have a risk exposure of**
  \[ RE = P(L) \times S(L) \]
How Can Risk Management Help You Deal With Risks?

- Buying information
- Risk avoidance
- Risk transfer
- Risk reduction
- Risk acceptance
Is Risk Management Fundamentally Negative?

- It usually is, but it shouldn’t be
- As illustrated in the Risk Acceptance strategy, it is equivalent to Opportunity Management
  
  \[ \text{Opportunity Exposure } OE = P(\text{Gain}) \times S(\text{Gain}) = \text{Expected Value} \]

- Buying information and the other Risk Strategies have their Opportunity counterparts
  - \( P(\text{Gain}) \): Are we likely to get there before the competition?
  - \( S(\text{Gain}) \): How big is the market for the solution?
What Else Can Risk Management Help You Do?

- Determine “How much is enough?” for your products and processes
  - Functionality, documentation, prototyping, COTS evaluation, architecting, testing, formal methods, agility, discipline, ...
  - What’s the risk exposure of doing too much?
  - What’s the risk exposure of doing too little?
- Tailor and adapt your life cycle processes
  - Determine what to do next (specify, prototype, COTS evaluation, business case analysis)
  - Determine how much of it is enough
  - Examples: Risk-driven spiral model and extensions (win-win, anchor points, RUP, MBASE, CeBASE Method)
- Get help from higher management
  - Organize management reviews around top-10 risks
Example Large-System Risk Analysis: How Much Architecting is Enough?

- Large system involves subcontracting to over a dozen software/hardware specialty suppliers
- Early procurement package means early start
  - But later delays due to inadequate architecture
  - And resulting integration rework delays
- Developing thorough architecture specs reduces rework delays
  - But increases subcontractor startup delays
How Soon to Define Subcontractor Interfaces?

Risk exposure RE = Prob(Loss) * Size(Loss)
-Loss due to rework delays

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

Many interface defects: high P(L)
Critical IF defects: high S(L)

Few IF defects: low P(L)
Minor IF defects: low S(L)

Time spent defining & validating architecture
How Soon to Define Subcontractor Interfaces?

- Loss due to rework delays
- Loss due to late subcontract startups

RE = P(L) * S(L)

Time spent defining & validating architecture

Many interface defects: high P(L)
Critical IF defects: high S(L)

Many delays: high P(L)
Long delays: high S(L)

Few delays: low P(L)
Short Delays: low S(L)

Few IF defects: low P(L)
Minor IF defects: low S(L)
How Soon to Define Subcontractor Interfaces?
- Sum of Risk Exposures

\[ \text{RE} = P(L) \times S(L) \]

- Many delays: high P(L)
  - Long delays: high S(L)
- Many interface defects: high P(L)
  - Critical IF defects: high S(L)
- Few interface defects: low P(L)
  - Minor IF defects: low S(L)
- Few delays: low P(L)
  - Short delays: low S(L)

Time spent defining & validating architecture
How Soon to Define Subcontractor Interfaces?

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

- Higher \( P(L) \), \( S(L) \): many more IF’s
- Mainstream Sweet Spot
- Many-Subs Sweet Spot
- Very Many Subcontractors

Time spent defining & validating architecture
How Much Architecting Is Enough?
-A COCOMO II Analysis

Sweet Spot Drivers:
Rapid Change: leftward
High Assurance: rightward
Concurrent System and Software Engineering
Sequential Engineering Neglects Risk

Arch. A: Custom many cache processors
Arch. B: Modified Client-Server

$100M
$50M

Response Time (sec)

Original Spec
After Prototyping

1 2 3 4 5
Value-Based Monitoring and Control
Control Theory

- Adapting to change
- Controlling progress toward SCS Win-Win realization
- Value realized (VBSE) vs. Project progress (Earned value)
Change as Opportunity
Agile Methods

- Continuous customer interaction
- Short value-adding increments
- Tacit interpersonal knowledge
  - Stories, Planning game, pair programming
  - Explicit documented knowledge expensive to change
- Simple design and refactoring
  - Vs. Big Design Up Front
Five Critical Decision Factors (Dimensions)

- Personnel (% Level 1B) (% Level 2&3)
- Criticality (Loss due to impact of defects)
- Dynamism (% Requirements change/month)
- Size (# of personnel)
- Culture (% thriving on chaos vs. order)

Critical Factors (Dimensions):
- Personnel
- Criticality
- Culture
- Size
- Dynamism
Conclusions

- Marketplace trends favor transition to VBSE paradigm
  - Software the major source of product value
  - Software the primary enabler of adaptability

- VBSE involves 7 key elements
  - Benefits Realization Analysis
  - Stakeholders’ Value Proposition Elicitation and Reconciliation
  - Business Case Analysis
  - Continuous Risk and Opportunity Management
  - Concurrent System and Software Engineering
  - Value-Based Monitoring and Control
  - Change as Opportunity

- Processes for implementing VBSE emerging
  - CeBASE Method, CMMI, DMR/BRA, Balanced Scorecard, RUP extensions, Strategic Design, Agile Methods
References