Building Process Improvement Business Cases Using Bayesian Belief Networks and Monte Carlo Simulation

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Ben Linders, Affiliate SEI / Ericsson Quality Mgr

SEPG NA 2009, San Jose, CA
Contents / Agenda

- Introduction
- Business Cases
- Quality Factors
- Validate
- Conclusions
Problem Statement

*Quality improvement* needed in many organizations

Business case

- Identification of problem areas
- Selected improvement
- Decision

Quantified

- Costs & benefits
- Lead time to result
Quantification problems

Much time needed to gather data
Difficult to measure things
Hard to keep management commitment
Expensive

Required: Business case, with limited but sufficient measurement effort, to gain management commitment and funding
Affiliate Collaboration

**Ericsson Netherlands: Market Unit Northern Europe & Main R&D Center**

R&D: Value Added Services

- Strategic Product Management
- Product marketing & technical sales support
- Provisioning & total project management
- Development & maintenance
- Customization
- Supply & support

---

**SEI Pittsburgh, PA: Software Engineering Measurement & Analysis**

Modern Measurement Methods

- Goal Driven Measurement
- Analyzing Project management Indicators
- Improving Process Performance using Six Sigma, Designing Products and Processes using Six Sigma
- Understanding CMMI High Maturity Practices
- Client Support & Research
- Training Development & Delivery
Solution

Technologies

- Bayesian Belief Networks (BBN)
- Monte Carlo Simulation
- Root Cause Analysis
- Cost of Quality, Defect Slippage

Six Sigma DMAIC Approach

- Modeling Business Cases
- Research Quality Factors & quantify Quality Improvement
- Validate “Business Case for Quality”
Fault Slip Through = Number of defects detected in integration & customer test that should have been detected earlier

“Should” implies that the defect is more cost effective to find earlier.
Building a business case

BBN

Monte Carlo

Business Cases

Quality

Fault Slip Through

Phase Performance

Historical Project Data

Industry Data

Current Quality Phase Performance

Improved Quality Phase Performance

Subjective Expert Opinion

Quality Factor

Quality Factor

Quality Factor

Quality Factor

Subjective Expert Opinion

Monte Carlo

Software Engineering Institute | Carnegie Mellon

© 2008 Carnegie Mellon University

Ben Linders, Ericsson

Ericsson
Bayes Belief Network (BBN)

- Probabilistic graphical model, to model uncertainty
- Diagnose and explain why an outcome happened
- Predict outcomes based on insight to one or more factors

Used:
- Modeling Quality Factors
- Predicting Quality Phase Performance
- What if Scenario
Monte Carlo Simulation

- Compute a result based on random sampling
- Modeling distributions of data
- Can make uncertainty visible

Used:
- Calculate value of process changes
Quality Phase Performance

**Quality Factor:**
Influencing quality of the delivered product
Management Factors

Management Context for Technical Activities

Direct:
- Project Management
- Process Management

Indirect:
- Strategic & Operational Line Management
Defect Insertion

Technical Activities where defects inserted

- Root Cause Analysis
- Defect Prevention
Defect Detection

Technical Activities where defects detected

• Early Detection
• Economy of Test
• Release Quality

Reduce Defect Slippage
Quality Factors

Purpose

- Predictor of Quality
- Leading indicator

Sources

- Research
- Expert opinion
- Experience
Connect defect data with Quality performance

- Maximum quality factor => Industry best in class
  Published industry data from various sources
- Distribution: Linear (keep it simple)

Extend BBN to calculate remaining defects after each phase

Result: Model for “what if scenario’s”

- Calculate defects in release products, when quality performance improves
- Cost of Quality data to calculate savings
Validate “Business Case for Quality”

- Quality Performance Assessment
  - Determine areas for Improvement

- Pilot: Agile for Requirements
  - Calculate value of process change
  - Run the pilot
  - Evaluate the result
Quality performance assessment

Survey based upon Quality Factors

- 34 respondents from management & technical roles
- 4 management areas & 7 technical areas

2 sub questions for each quality factor:

- How relevant is the factor when we want to improve quality? “little if any,” “moderate,” “substantial,” or “extensive,”

- How well are we doing currently? “poor,” “fair,” “good,” and “excellent.”
Agile for Requirements

Problem areas

• Requirements Stability
• Scope Stability
• Requirement Definition Capability

Agile solution

• Planning Game
• Stand-up meetings
• Architecture team
• Iterations
• Risk Based Testing
**Pilot: Prevent Requirement Defects**

**Monte Carlo Simulation**
- Simulate Current Performance on Defect Insertion & Detection
- Estimate Agile improvement (expert opinion)

**Bayes Belief Network**
- Ericsson data to calibrate current performance
- Predict savings due to less defects inserted

### Current

<table>
<thead>
<tr>
<th>Phase</th>
<th>Quality Factor</th>
<th>Detected defects</th>
<th>Defects left</th>
<th>Detection %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req</td>
<td>4.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arch</td>
<td>5.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impl</td>
<td>5.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inspection</td>
<td>5.3</td>
<td>12</td>
<td>37</td>
<td>24%</td>
</tr>
<tr>
<td>Early Test</td>
<td>5.0</td>
<td>12</td>
<td>25</td>
<td>32%</td>
</tr>
<tr>
<td>Late Test</td>
<td>6.2</td>
<td>11</td>
<td>14</td>
<td>44%</td>
</tr>
<tr>
<td>Customer Test</td>
<td>5.0</td>
<td>5</td>
<td>9</td>
<td>36%</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maint</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Improved

<table>
<thead>
<tr>
<th>Phase</th>
<th>Quality Factor</th>
<th>Detected defects</th>
<th>Defects left</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req</td>
<td>4.7</td>
<td></td>
<td></td>
<td>4%</td>
</tr>
<tr>
<td>Arch</td>
<td>5.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impl</td>
<td>5.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>9%</td>
</tr>
<tr>
<td>Inspection</td>
<td>5.3</td>
<td>11</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Early Test</td>
<td>5.0</td>
<td>11</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Late Test</td>
<td>6.2</td>
<td>10</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Customer Test</td>
<td>5.0</td>
<td>5</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maint</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

© 2008 Carnegie Mellon University

Ben Linders, Ericsson
Results Agile

- Very low number of requirement defects
- Previous projects also had a low number
- Based upon the data no conclusion could be drawn

Root Cause Analysis:

- understanding requirements increased: planning game & stand-up meetings.
- Improvements from retrospectives increased cooperation between development team and product owner.

Requirements quality performance increased!
Evaluation Business Case

Bayesian Belief Networks were successful:

- High level graphical overview
- Focus on improving requirements quality
- Positive results from the pilot
- Limited investment needed for the business case
- We do not know for sure if improvements in another area would have been more beneficial

Monte Carlo Simulation of potential results has had less value:

- Limited data available on requirements
- Used Root Cause Analysis to confirm improvements
Conclusions

Quicker Business Cases with BBN:

- Quality Factors/Performance
- Fault Slip Through

Monte Carlo value Simulation:

- Distribution of cost savings

Benefits

- Quick Improvement Scope
- Better Business Case: Value for Business
- Agile increased requirements quality
- Value based measurement approach
Contact:

Ben Linders  
Ericsson R&D  
The Netherlands

Ben.Linders@Ericsson.com
Increased Value

Modeling Business Cases Decision Aids

Figure 1: Roadmap for research in software engineering economics.
Affiliate Assignment

Joint effort: Ericsson (Ben Linders) and SEI (Bob Stoddard)
- Time, money, materials
- Knowledge & experience

Deliverables Ericsson
- Defect data & benchmarks
- Improved decisions skills
- Business case & Strategy 2007:
  - Early phases: Improvements
  - Late test phases: Reduction

Research contribution
- Apply Six Sigma business cases
- Verify technology (CoQ, RBT, FST, etc)
Six Sigma DMAIC mapping

2006: **DMA**  Project “Improved Test Decisions”:
   - Identify current data used
   - Potential areas for Quality Improvement
   - Define required data & set baseline
   - Propose improvement for 2007

2007/2008: **IC**  Six Sigma based Strategic Improvement program
   - Pilots: Agile, Modeling, Quality Culture
   - Measure benefits
   - Institutionalize improvements
Define

Project Scope

- Problem Statement
- Baseline data
- Goal
- Voice of Customer/Business
- SIPOC
- Quality Roadmap

Establish Project

- Initial Business Case
- Scope & Excluded
- Research Areas
- Assignment roles, costs, planning
Measure

Identify needed data
- Process Model
- GQIM
- Hypothesis & Predictions
- Required data

Obtain Data Set
- Prediction model
- Available data

Evaluate Data Quality
Summarize & Baseline Data
- Benchmarking
Analyze

Explore data

Characterize process & problem

Update improvement project scope & scale
Improve

Pilot Agile Requirements
Measure impact on slipped through defects
Root Cause Analysis
Control

Establish measurements on Balanced Scorecard

- Fault Slip Through Root Cause Analysis for continuous improvement
- Defects after Release Both predicted (handshake with product owner) and actual

Quality Steering on all levels

- Inside Project: Planning game, Root Cause Analysis
- Programs: Monthly Project Steering Group
- R&D: Monthly quality meeting with MT members
Two step approach

Quality Factor Model
- Expert opinion, extended with data
- Quick Quality Scan
- Rough Prediction Fault Slip Through
- Improvement Areas

Defect Prediction Model
- Data, tuned with expert opinion
- Detailed Prediction Fault Slip Through
- Improvement Business Case
Example of Bayesian Belief Model

**STEP 1:**
Develop a model that depicts leading indicator nodes and/or root cause nodes for each node to be predicted. In this diagram, the flow is from left to right and thus, each child node (nodes with incoming lines) may be predicted with the parent nodes (nodes sending a line to a child node).

**STEP 2:**
Model the relationship between each “child” node and the associated “parent” nodes for the child using:

Regression & ANOVA with Objective Data Design of Experiments with Subjective Data
Economic Model

Understand costs of defects
Process & project performance
Dialog managers & developers
Use operational data
Manage under uncertainty & incomplete data

Technologies

- Cost of Quality
- Bayesian Belief Networks
- Real Options
- Lean Six Sigma
Quality Prediction

Current Model: Estimation

- Extrapolate past performance
- Based on inserted/detected defects
- Plan & track

Wanted: Prediction

- Causes of defects
- What if Scenarios
- Decision taking

All models are wrong
Some models are useful

Deming
History Defect Modeling

2001
  • Defect Model defined, pilot in first project

2002/2003
  • Improved based on project feedback
  • First release quality estimates
  • Industrialize model/tool, use in all major projects

2004/2005
  • Targets: Project portfolio management
  • Process Performance & Cost of Quality

2006/2007
  • Process Improvement Business Cases
  • SW Engineering Economics, Six Sigma
  • Defect Prediction
Project Defect Model

Why?

• to control quality of the product during development
• improve development/inspection/test processes

Business Value:

➔ Improved Quality
➔ Early risks signals
➔ Better plans & tracking
➔ Lower maintenance
➔ Save time and costs
➔ Happy customers!
Step 2: Defect Causes & Effect

From Estimation to Prediction

- Resident Defects in Design Base
- Defect Density
- Detection Rate
- Fault Slip Through
- Defect Classification
- Resident Defects in Delivered Product
- (Un)happy customers
- Defect Level

Design Process
- Competence, skills
- Tools, environment

Test Process
- Competence, skills
- Test Capacity
- Tools, environment

Process Inputs and Outputs
- Measurement
- Influeencing factors
- Test Capacity
- Tools, environment

Defects Inserted (documentation, code)

Defects Detected (Inspection, test)
Step 2: Defect Prediction

Fault Slip Through
Defect found in a (later) test phase that should have been found earlier

“Should”: More Cost effective (economical)

Predict Defect Reduction

- Determine process impact
- Simulate quality change
- Predict savings

Pilots

- Agile
- Model Driven Development
Process Performance

Project Data

- Insertion Rates
- Detection Rates
- Defect Distribution
- Fault Slip Through
- Post Release Defects

Process View

- Performance of design & test processes
- Benchmarking
- Best Practices & Improvement Areas
Steering Agile Quality

- Estimate amount of latent defects after demo in the planning game.
- Collect all defects during the test phases (after the demo).
- Classify defects:
  - “introduction phase”
  - “should have been detected phase”
- Root cause analysis on defects that should have been found before demo.
- Decide on improvement actions and present to the project team.
- Re-estimate remaining defects and predict release quality.
Monte Carlo: Quality performance

Monte Carlo simulation

- Input from 5 experts
- Estimated chance of occurrence and impact on FST (1-5 scale)
- Simulation done to calculate impact on quality factors
- Result used in BBN model to calculate effect on defect slippage

Expected result:

- Reduced number of requirement defects introduced
- Increased effectiveness of late testing phases
- Less defects in products shipped to customers
- Cost saving:
  - Limited saving in the project
  - Major saving during maintenance
SEI Affiliate

The Software Engineering Institute Affiliate Program provides sponsoring organizations with an opportunity to contribute their best ideas and people to a uniquely collaborative peer group who combine their technical knowledge and experience to help define superior software engineering practices.

Affiliates: [http://www.sei.cmu.edu/collaborating/affiliates/affiliates.html](http://www.sei.cmu.edu/collaborating/affiliates/affiliates.html)