Agile Requirements

Agile Consortium Benelux, sep 30, 2009

Ben Linders
Contents / Agenda

- Introduction
- Business Cases
- Quality Factors
- Agile Requirements
- 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
Introduction

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)
Two models

Defect Estimation Model
- Data, tuned with expert opinion
- Estimate Fault Slip Through
- Project/Product Quality

Quality Factor Model
- Expert opinion, extend with data
- Quick Quality Scan
- Prediction Fault Slip Through
- Improvement Areas
Measuring quality

Insertion: Where are defects made? How to prevent?

Detection: Where are defects found? Early/economic removal?

Quality: How many defect are left in the product at release?
Process View

Design Process Competence, skills Tools, environment

Test Process Competence, skills Test Capacity Tools, environment

Resident Defects in Design Base Defects Inserted (documentation, code) Defect Density

Defects Detected (Inspection, test) Detection Rate Fault Slip Through Defect Classification

Resident Defects in Delivered Product (Un)happy customers

Defect Level

[Diagram of process view with nodes and arrows indicating flow of defects through design, insertion, detection, and customer satisfaction]
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.
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 - Requirements

Purpose

• Predict Quality
• Leading indicator

Sources

• Research
• Expert opinion
• Experience

Requirements Performance

Process Maturity

Requirement Management Capability

Requirement Stability

Commitment

Roadmap Quality

Scope Stability

Root Cause Analysis

Requirement Definition Capability

Stability

Commitment

Process Maturity

Requirement Management Capability

Requirement Stability

Roadmap Quality

Scope Stability

Root Cause Analysis

Requirement Definition Capability

Stability

Commitment

Process Maturity

Requirement Management Capability

Requirement Stability

Roadmap Quality

Scope Stability

Root Cause Analysis

Requirement Definition Capability

Stability

Commitment
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.”
Findings Requirements

- Process Maturity
- Requirement Management Capability
- Requirement Stability
- Commitment
- Roadmap Quality
- Scope Stability
- Root Cause Analysis
- Requirement Definition Capability

Requirements Performance

Agile Req.
Pilot “Business Case for Quality”

Context:
- Process management
- Quality steering
- Starting with Agile

Pilot: Agile for Requirements
- Calculate value of process change
- Run the pilot
- Evaluate the result
Improve: Requirements Stability

Requirements Stability – Inverse of the amount of requirement changes over time. (The less changes, the higher stability.)

Agile deployment

- Backlog with Prioritized User Stories
- Product manager as Product Owner
- (Pre-) Planning game
- Architecture team
- Stand up meetings
Improve: Scope Stability

Scope Stability – Impact of major changes in projects that are related to changes in the product roadmap, including stability of the products to be developed, development teams involved in projects, and major changes in project funding or delivery dates.

Agile deployment

- Backlog
- Responsibility of Agile teams and Product Owner
- (Pre-) Planning game
- Retrospectives
Improve: Requirement Definition Capability

Requirements Definition Capability – The skill and experience level of the people doing requirements definition (e.g., product managers).

Agile deployment

- (Pre-) Planning game
- Stand up meetings
- Collaborative Culture
- Retrospectives
Steering Agile Quality

- Estimate latent defects after demo (planning game)
- Collect defects during test (after demo).
- Classify defects:
  - "introduction phase"
  - "should have been detected phase"
- Root cause analysis: Prevention
- Decide improvement actions and communicate
- Re-estimate and predict release quality.
Results Agile for Requirements

- 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!
Conclusions

Quicker Business Case:
• Quality Factors/Performance
• Fault Slip Through
• Combining data and expert opinion

Improved Requirements Performance
• Agile increased requirements quality
• Less defects after release
• Increased flexibility and collaboration
More information

Publications:

• *Building Process Improvement Business Cases*
  SEI Technical Note: [http://www.sei.cmu.edu/library/abstracts/reports/09tn017.cfm](http://www.sei.cmu.edu/library/abstracts/reports/09tn017.cfm)

• *Controlling Project Performance by Using the Project Defect Model*
  in proceedings PSQT West Conference 2005

• *The Business Benefit of Root Cause Analysis*
  in proceedings SM/ASM conference 2003

• SPI, the agile way!
  To be presented at the SPIder conference, october 2009
  [www.spiderconferentie.nl](http://www.spiderconferentie.nl)

Contact:

• Email: benlinders@gmail.com

• [http://www.linkedin.com/in/benlinders](http://www.linkedin.com/in/benlinders)
Backup Slides
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”
Building a business case

BBN

Monte Carlo

Business Cases

Quality

Fault Slip Through

Phase Performance

Historical Project Data

Industry Data

Subjective Expert Opinion

Current Quality Phase Performance

Improved Quality Phase Performance

Quality Factor

Quality Factor

Quality Factor

Quality Factor
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 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
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
Quantify Quality Improvement

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
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