Synthesizing Software Quality Metrics for Executive Operations Reviews

Eric Seufert
Digital Chocolate

UKSMA/COSMIC International Conference on Software Metrics & Estimating
October 27 & 28, 2011
London, UK

Why focus on quality at an Operations Review?

- Quality touches other strategic business areas:
  - Brand management / reputation;
  - Product roadmap development;
  - Customer service;
  - Staffing needs.

Why focus on quality at an Operations Review? (2)

- Quality can be used as a proxy for measuring effectiveness of processes:
  - If quality is trending lower, engineering practices may need to be changed.
- Quality can be used as a means of making impediments visible:
  - Operational impediments have a direct effect on product quality.

From what angle is quality usually examined?

- Product Owner's perspective:
  - Inward focus:
    - Granular.
    - What are the factors affecting quality?
    - What can be done to detect and address bugs faster and more thoroughly?
    - Metrics can be specific to the product; no necessity in comparing to other products within the organization.

How does this perspective change for an operations review?

- Executive's perspective:
  - Outward focus:
    - No granularity; enterprise-level focus.
    - What is quality, and how does that affect business concerns?
    - Where should resources be allocated?
    - And, in an organization with many products of different sizes and levels of maturity: are these metrics comparable across the catalog?

How is software quality measured?

- Product metrics
  - Intrinsic product quality measured through number of bugs
    - Absolute bug count, mean time to bug find, mean age of bugs, priority spectrum, etc.
  - Opportunities for Error (OFE) based on Lines of Code (LOC) or Function Points (FP)
    - Function points calculated as Function Counts x Value Adjustment Factor
  - Defect Removal Effectiveness (DRE): Percentage of bugs removed from development cycle in which they were introduced

How is software quality measured? (2)

- Process metrics:
  - Testing thoroughness
    - Number of test cases failed, percentage automated vs. manual.
  - Continuous integration
    - How often are builds committed? How many builds are committed in a release cycle?
- In-Market Metrics
  - Defect Detection Percentage (DDP): Percentage of defects found before release
    - Measures bug leakage into market.

How is software quality measured? (3)

- Enterprise-level metrics
  - Problems per User-Month (PUM):
    - Total Problems (actual bugs + perceived defects, including usability issues, etc.) / total license-months of the software.
    - License-month: install licenses x months in calculation period.
  - Reducing PUM achievable through a variety of business initiatives:
    - Reduce defects;
    - Reduce *perceived* defects by improving documentation and customer service;
    - Sell more licenses (increase PUM denominator).

Measuring quality for the operations review

- Business objective:
  - High-level overview of product quality to drive decision-making.
- Trends, not snapshots:
  - Because the operations review occurs infrequently, snapshots don't provide enough context to be useful.
  - Focus on trends; *is quality improving?*

Measuring quality for the operations review (2)

- *Quality* as an achievable objective:
  - If quality is conceptualized as something that can be achieved to some standard in the short- to medium-term, then the distance (in days / headcount / releases, etc.) from that goal is quantifiable.
- Universal comparability:
  - Only look at metrics that can be compared across the product portfolio. Unique measures of quality have no meaning in the operations review.

What metrics are most important?

- Comparison needed, review-to-review;
- Normalized metrics provide the most context:
  - In-Market: DDP, how many bugs escape to market.
  - Product: Mean age of open bugs, mean time to bug find, Priority breakdown, DRE.
    - *Note*: depends on maturity spectrum of products

What metrics are most important? (2)

- Time-based measures provide strategic insight and can be compared, review-to-review:
  - Based on current velocity, when will the quality standard be reached?
    - Bug-fix velocity needed to achieve quality by next release: Total outstanding bugs / days to next release.

Bug-fix Velocity

- If our bug-fix velocity is positive (more bugs fixed than created for a given period), then the time remaining before all bugs are fixed can be quantified:
  - If it's negative, it's not estimatable.
- *Given the current trajectory, after how many days will the bug count be 0?*
  - This metric provides little insight if:
    - The product has a small bug backlog;
    - The product has a tight bug-fix velocity margin (all bugs fixed almost immediately).
  - Useful mostly for purposes of comparison

How to measure improvement?

- What should be tracked?
  - Did bug-fix velocity change direction?
  - How has bug-fix velocity changed?
  - How has Mean Age of Open Defects changed?
  - How has Mean Time to Fix changed?
- Binary improvement measure (yes / no) more valuable than percentage change:
  - Look for continuous improvement.

Introducing a qualitative Element

- Product Owners should have the opportunity to address impediments and outline strategic quality initiatives:
  - Context is king at the Operations Review;
  - Making impediments visible is of paramount importance.

Resistance to quality metrics

- People don’t like to be measured;
- Implementing process improvements can be time-consuming;
- Hard to make any metric truly universal:
  - Comparing products within the catalog doesn’t always make sense.
- Focus at the operations review level is on bugs:
  - Process elements can affect bug-fix velocity, producing misleading metrics.
- Seen as introducing bureaucracy and “paperwork” into the engineering process.

How to win POs over to quality metrics tracking

- Focus on trending:
  - Trends matter, not absolutes. Emphasize trajectory.
- Make process requirements transparent;
- Don’t measure in the abstract:
  - Use metrics that are easy to grasp;
  - Use actionable metrics;
  - High-level metrics give POs the agency to implement process improvements;

Appendix

- Function Counts: $$FC = \sum_{i=1}^{5} \sum_{j=1}^{3} w_{ij} \times x_{ij}$$
  - w_{ij} are the weighting factors of the five components by complexity level (low, average, high) and x_{ij} are the numbers of each component in the application.
- VAF: $$VAF = 0.65 + 0.01 \sum_{i=1}^{14} c_i$$
  - C_i being the sum of weights (0-5) for a set of 14 characteristics