Metrics and Measurements: The Importance of Integrating the Business View

SEPG 2006

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

JohnTHarding@compuserve.com

How many of your organizations

- **are setting quantitative process objectives based on business needs?**
- **are collecting and analyzing rework effort/cost data?**
- **are setting quantitative effectiveness goals across the organization?**
- **meet on an ongoing basis with the financial organization to discuss financial and business objectives?**

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

Agenda

- Some practical examples to address integrating the business view for common problems
- Two traps to avoid

A Business View

Three important business drivers

- **Growth in the marketplace**
- **Profit margins**
- **Cash flow from operations**

What can positively impact these drivers?

- **Reduction in costs, especially rework costs**
- **Shorten time to market**
- **Improvements in quality**
- **Meeting committed delivery dates consistently**

Note that frequently actions to improve quality also have a positive impact on time to market and reduction in cost.

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

Relate Improvements to the Business Driver

Which driver is most important?

- **Use that to determine implementation priority for Software or Engineering Process Improvements (SPI or EPI)**
- **Do not ignore the other business drivers when implementing the improvements since they may be impacted; possibly in a negative way**
  - **E.g., Time to market and automation**
- **Measure the results and involve the financial organization and senior executives**

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

Let's Work Through Three Examples

Improve profit margins

- **Reduction in rework effort/costs**

Improvement in ability to meet committed delivery dates to avoid cash flow problems

- **More predictable estimates**

Shorten time to market

- **Reduce calendar time for test**

Improve Margins by Reducing Rework
Example 1

- **Many studies have shown rework is 40%-60% of project costs**
  - **This is usually based on test defect data and frequently does not include unit test**
- **Actual data is not always collected for non defect related rework**
  - **At least two CMM Level 5 organizations are collecting it and it can be significant**
- **If improving margins is important rework may be the biggest cost factor available for reduction**

Approach for Reducing Rework

- **Use rigorous inspections to find more defects closer to where they are injected**
  - **Inspections typically take a small percentage of the effort to correct compared to being found later in test**
  - **Effort drives cost**
- **Improve the effectiveness of the inspections**
- **Set organizational level goals which all projects must achieve**
- **Start systematic, data driven, defect prevention**

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

Rigorous Inspections

A review of a work product,

which passes stated entrance criteria,

led by a moderator who is not the author,

that seeks and records defects in that work product,

uses individual preparation and a group meeting,

uses product specific checklists,

uses scenarios and/or other effective reading techniques,

initiates and monitors rework as necessary,

initiates re-inspection based on stated criteria,

passes or fails the work product based on exit criteria,

and adds to the base of historical data.

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

JohnTHarding@compuserve.com

Inspection Effectiveness

Percentage of the defects in the work product being identified during inspections

Teams doing rigorous inspections can typically start finding around 50% of the total defects

Improve the effectiveness; it can go to 95+%

- **Requires collecting and analyzing inspection, test, and customer defect data to identify patterns**
  - **E.g., pareto analysis to find common characteristics of problems and updating checklists/procedures**
  - **Relational databases with "like" searches are useful**
- **Defect density matrices help selection criteria**
- **Developers need to be involved in the analysis**

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

Inspection efficiency is the effort hours per defect.

Effort To Inspect 1kloc of Code

Assume

- **Modification to existing code and uplift factor is 5**
- **Optimum rate is 100 lines of executable code per hour**

Effort required

- **1kloc + 5 kloc = 6 kloc to inspect**
- **Preparation time is 60 hours**
- **Inspection time is 60 hours**
- **Total effort required for EACH participant is 120 hours!**

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

Where Should You Initially Focus?

Focus on effectiveness first

- **Why make a process which is less effective than desired more efficient?**

But where will many managers want to focus first?

Once you achieve desired effectiveness then try to improve efficiency while holding effectiveness constant

Establish an Organizational Goal for All Projects to Improve Inspection Effectiveness

- **Each project needs to quantify the percentage of defects which are found prior to the start of test**
  - **This becomes the baseline for that product or type of product for the next project**
- **Since projects will be different we need an organizational goal that will be fair for all projects**
  - **Each project must improve the percentage of defects found prior to the start of test by a fixed (and common) percentage such as 5% or 10% from the baseline; this is achievable with work for almost every project up to the point where they are at 95%**

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

Improve Estimates to Improve Ability to Meet Committed Delivery Dates

Example 2

Establish an organizational measurement for schedule predictability

- **Must allow for outliers**
- **Suggestion: 80% of projects are not more than X% late**
  - **X should not be outlandish; perhaps 20%**
  - **Based on historical data**
- **Set objectives for a reduction in X%**
- **Each project must identify specific actions to show how they will achieve that improvement**

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

Some Approaches for Reducing Late Delivery

- Improve estimates by understanding if there are patterns for what has been underestimated in past projects
- Understand if contingencies for growth or change have been insufficient and what project characteristics can be used to help establish more appropriate contingencies
- Consider the use of buffers between project committed end date and committed delivery date for business critical projects where the estimates have low confidence

Approach for Improving Estimates

When doing the initial size estimate capture key information from the estimation exercise

- **Assumptions, dependencies, risks, non-standard tasks, non-documented requirements, and component/detailed sizings**

At the notional end of each technical life cycle activity re-estimate the size and capture the reasons for the growth

At the end of the project, and across all projects, look for patterns to explain the growth

Modify the estimating procedures

Reduce Time to Market by Reducing Calendar Time for Test
Example 3

Approach 1

- **Use inspections to shift defect identification and removal from test to development activities**
- **Increase inspection effectiveness as per example 1**

Reduce Time to Market by Reducing Calendar Time for Test

Example 3

Approach 2

- **If testing is spending effort on test problems such as test case in error, no trouble found, duplicates, etc. than set objectives for reducing the number of test problems and effort due to test problems**
  - **Set aggressive targets for each project. Note that this will usually be domain and technology specific so organizational goals or objectives may not be appropriate.**
  - **Use pareto analysis to determine which problem types consume the most effort. Note this may not be the most frequently occurring problem type.**
  - **Also look for problem types that block testing progress**
- **Establish action plans**

Reduce Time to Market by Reducing Calendar Time for Test

Example 3

Approach 3

- **If testing is primarily manual then regression testing is unlikely to be done frequently, regression coverage may be minimal, and reasonable regression testing coverage may take a lot of time**
- **If regression problems are a significant source of defects in test or in customer use then this is another reason to consider automating regression testing**
- **There may be an initial cost and a need to develop some in house tools or in house extensions to commercial tools**
- **Automated regression testing, especially where the execution and analysis are both automated can save time, reduce effort, improve coverage, and permit earlier identification of defects**

Two Traps to Avoid

Management wants to improve productivity by achieving higher LOC/PY

Management wants to reward projects with low numbers of defects

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

"Production Rate" vs. "Productivity Rate"

LOC/PY is a measure of the PRODUCTION RATE

- **It has nothing to do with economic productivity**
- **Maximizing economic productivity would be maximizing reuse; measure amount and type of reuse not LOC/PY**
- **Higher LOC/PY may cause people to maximize code, waste development and test effort, increase number of defects**
  - **Eliminate reuse**
  - **No macros or subroutines**
  - **Use poor coding techniques to maximize LOC count**
- **The historical production rate for a product or type of product can be measured (provided data is collected!)**
- **Can you equate more code from reuse with more New and Changed LOC?**

Software Technology Transition
60 Elm Street
Andover, MA 01810
978-475-5432

Reward Projects with Low Number of Defects?

Do some projects, especially legacy projects, have higher defect density rates (DEF/KLOC) than new projects?

Do projects with large size have more defects than small projects?

- **If so then how could the absolute number of defects be a comparison which shows true business value?**

Might this cause aberrant behavior such as "bundling fixes"?

John (Jack) Harding

JohnTHarding@compuserve.com

www.stt.com