



Lessons Learned in Project Metrics: Are Your Metrics Dumb or Smart?

February 16th, 2010

Software QA Discussion Group

Mike Griffiths, PMP
Leading Answers Inc.
www.LearningAnswers.com

My Background

- **Project Manager and Trainer**
 - >20 years IT experience on utilities, defense, & finance
 - 8 years PMO Agile-to-Traditional Integration
- **Agile Project Management**
 - Helped create Agile method DSDM in 1994
 - 16 years agile project experience (DSDM, XP, Scrum, FDD)
 - Board director of Agile Alliance and APLN
 - Author, trainer, and presenter Agile Conference 2001-9
- **Traditional Project Management**
 - PMP, PRINCE2 certifications
 - PMBOK v3 contributor and reviewer
 - Trainer for PMI SeminarsWorld 2005-2010
 - Presenter PMI Global Congress 2004-2009



Measurement

Not all observations are useful:

- The sun rises up in the sky in the morning and then falls down again at night
- Planets revolve around the earth
- Stars come out at night
- Heavier objects fall faster than lighter objects

"There are so many possible measures in a software process that some random selection of metrics will not likely turn up something of value" - Watts Humphrey

Not all useful things are easily observable:

- Spouse's mood
- Sponsor Confidence
- Team Commitment

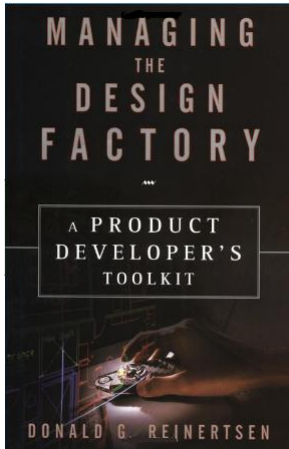
"Not everything that can be counted counts, and not everything that counts can be counted" - Albert Einstein.

The Hawthorne Effect



You **will** influence what you measure...

Design Factory Metrics



Metrics Should be:

1. Simple, self-generating
2. Relevant to the end-goal
3. Ideally, leading, future-focused

Traditional Metrics Examined

Desirable Characteristics:

- ✓ The Hawthorne Effect is positive
- ✓ Simple, self generating
- ✓ Relevant to the end-goal
- ✓ Leading, future-focused

Traditional Metrics:

- **Lines of Code Written** - poor, does not reward simplification, leads to code bloat
- **Function Points Delivered** - poor, effort to generate, not relevant to the end-goal of project
- **Hours Worked** - poor, leads to long hours, burn-out, defects, consumed budgets



Smarter Project Metrics

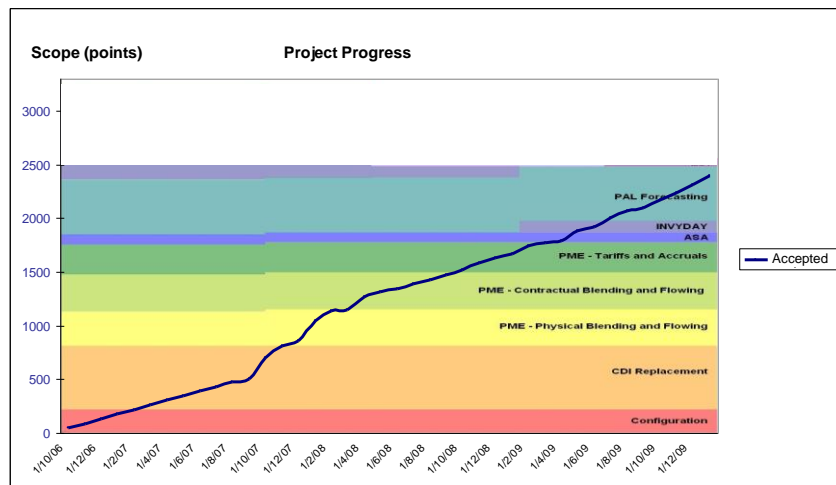
Since you will influence what you measure...

Choose Smarter Metrics:

- Features Accepted
- Sponsor Confidence
- User Satisfaction
- Defect Cycle Times

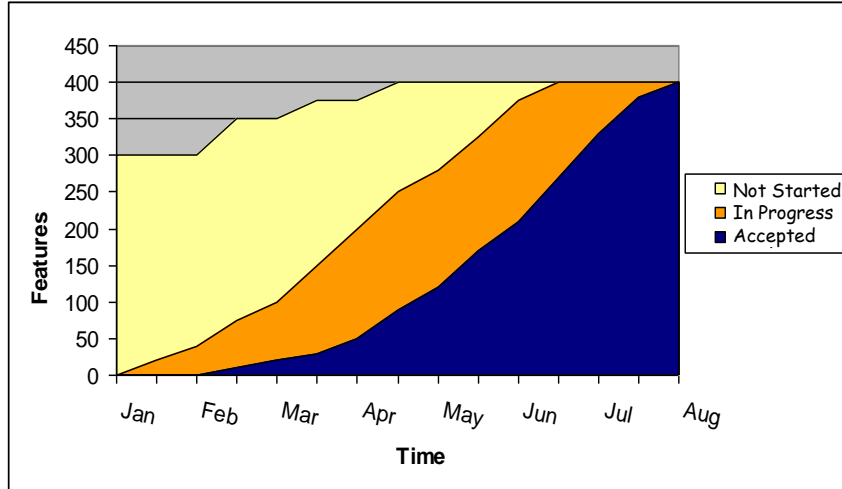
Smarter Metrics

- Features Accepted



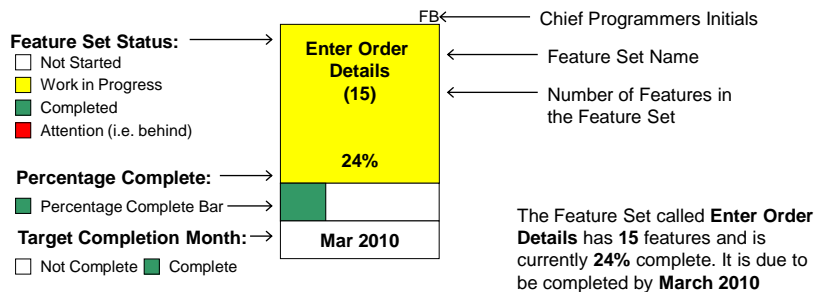
Smarter Metrics

- Features Accepted

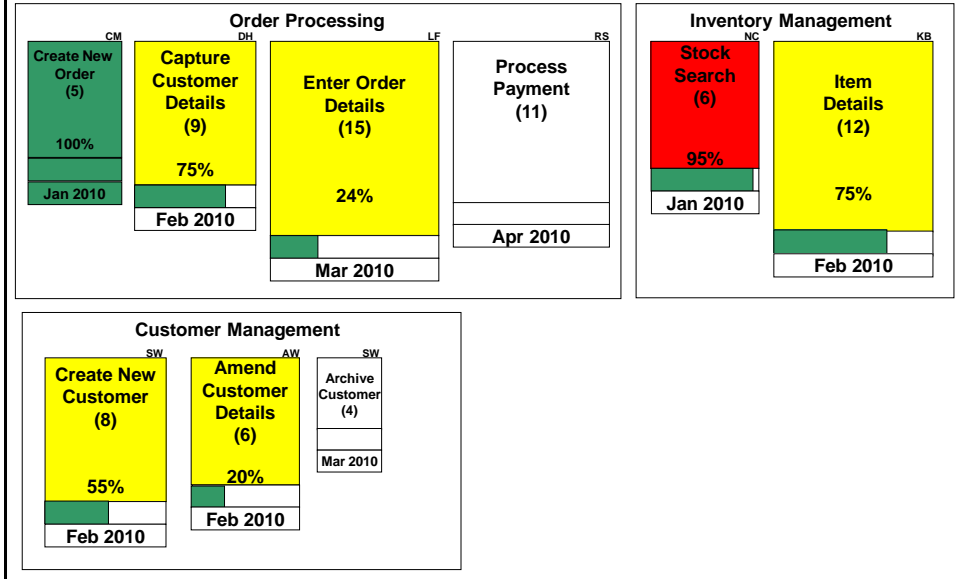


Feature Based Reporting

Example 2: Parking lot Diagrams

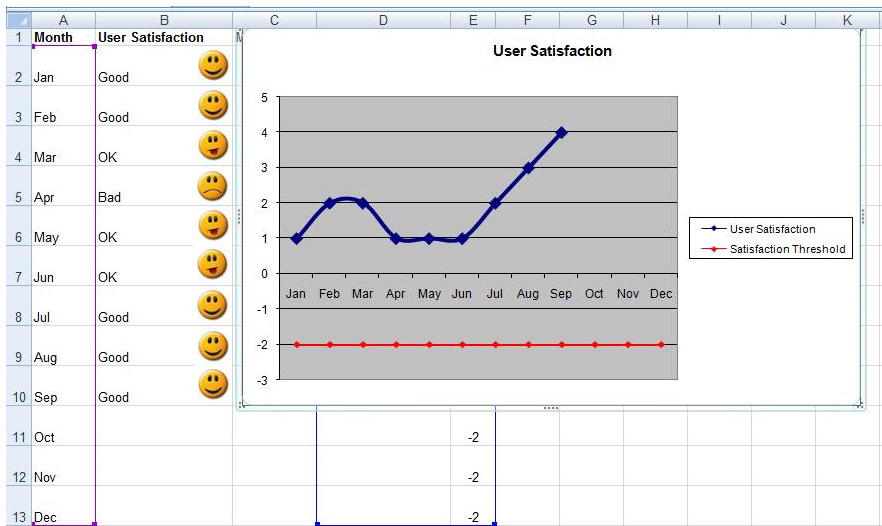


Feature Based Reporting



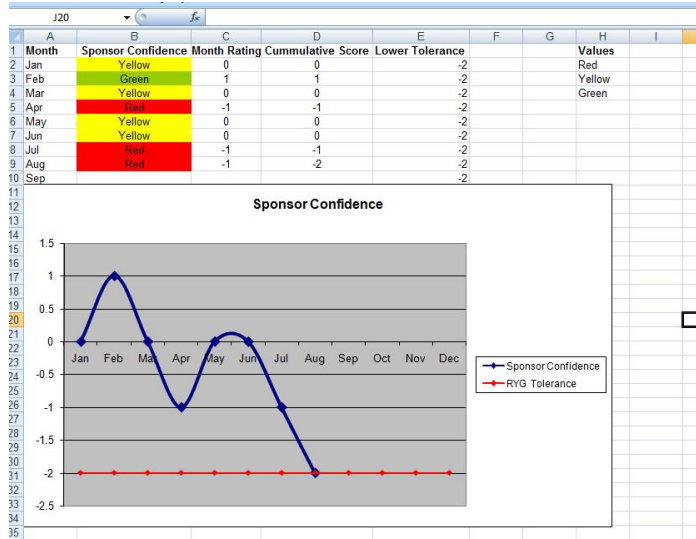
Smarter Metrics

User Satisfaction



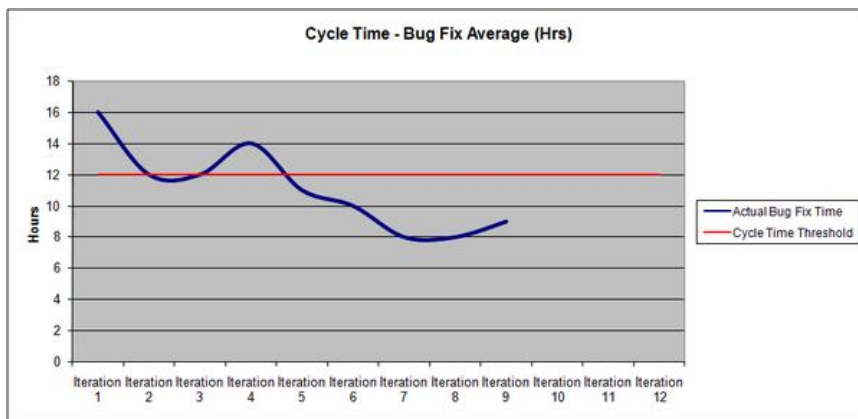
Smarter Metrics

- Sponsor Confidence

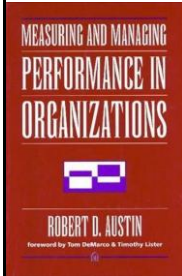


Smarter Metrics

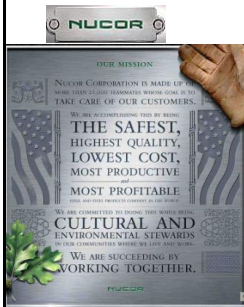
- Defect Cycle Time



Measuring Up



- "You get what you measure"
- "You get only what you measure, nothing else"
- "You tend to lose the things that you can't measure: insight, collaboration, creativity"



- Nucor Steel - From startup in 1970 to North America's largest and most profitable steel company
- Incentive pay based on productivity
- Measure one level above where expected / directly controllable

Measuring Up in Software

- Defects could be traced back to individual developers, but may be environmental
- Rolling up to an entire team and get testers involved earlier to provide more timely and valuable feedback to developers

*"Instead of making sure that people are measured within their span of control. It is more effective to measure people **one level above their span of control**. This is the best way to encourage teamwork, collaboration, and global, rather than local optimization" - Mary Poppendieck*

Leading vs. Lagging Metrics



- For an accountant, a perfect view of the past might be useful
- For a project manager, a perfect or even an imperfect view of the future is far more useful

Pay less attention to Lagging Metrics

- Actual values

Pay more attention to Leading Metrics

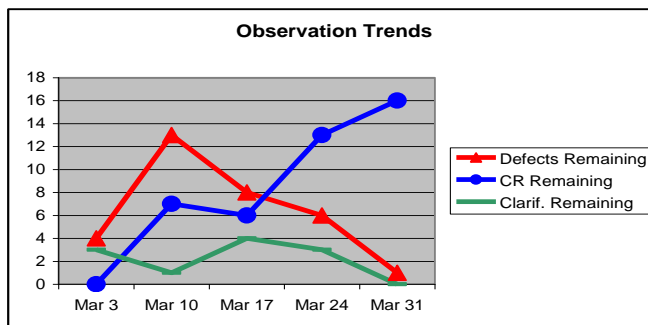
- Trends
- Impacts of projections

Trends

Observations	Mar 3	Mar 10	Mar 17	Mar 24	Mar 31
Defects Opened	5	25	30	20	10
Defects Closed	1	16	35	22	15
Defects Remaining	4	13	8	6	1
CR Opened	0	18	20	23	12
CR Closed	0	11	21	16	9
CR Remaining	0	7	6	13	16
Clarif. Opened	9	12	14	8	2
Clarif. Closed	6	14	11	9	5
Clarif. Remaining	3	1	4	3	0
Total Observations	7	21	18	22	17

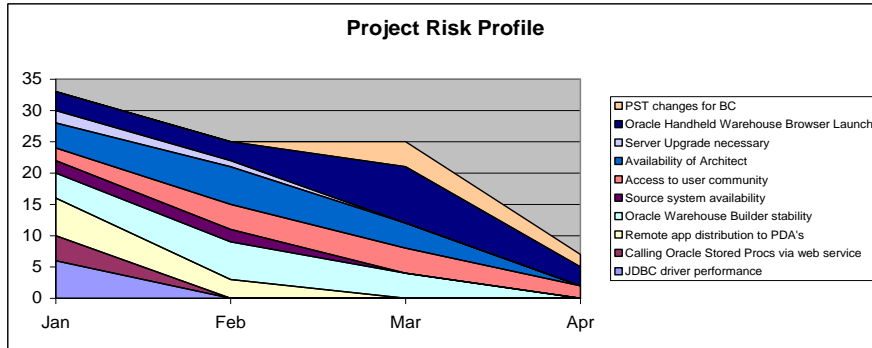
CR = Change Requests

Clarif. = Clarifications

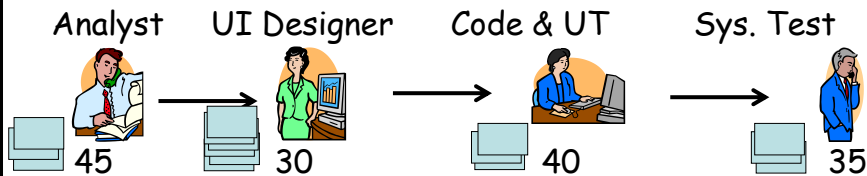


Trends - Risks Reduced

ID	Risk Short Name	Jan			Feb			Mar			Apr		
		Imp	Prob	Sev	Imp	Prob	Sev	Imp	Prob	Sev	Imp	Prob	Sev
1	JDBC driver performance	3	2	6	3	0	0	3	0	0	3	0	0
2	Calling Oracle Stored Procs via web service	2	2	4	2	0	0	2	0	0	2	0	0
3	Remote app distribution to PDA's	3	2	6	3	1	3	3	0	0	3	0	0
4	Oracle Warehouse Builder stability	2	2	4	2	3	6	2	2	4	2	0	0
5	Source system availability	2	1	2	2	1	2	2	0	0	2	0	0
6	Access to user community	2	1	2	2	2	4	2	2	4	2	1	2
7	Availability of Architect	2	2	4	2	3	6	2	2	4	2	0	0
8	Server Upgrade necessary	1	2	2	1	1	1	1	0	0	1	0	0
9	Oracle Handheld Warehouse Browser Launch	3	1	3	3	1	3	3	3	9	3	1	3
10	PST changes for BC	0	0	0	0	0	0	2	2	4	2	1	2
							25			25			7

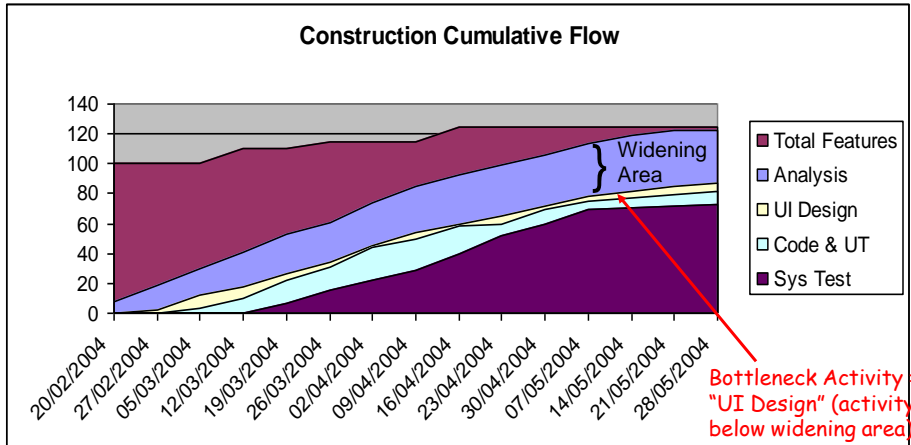


Cycle Times

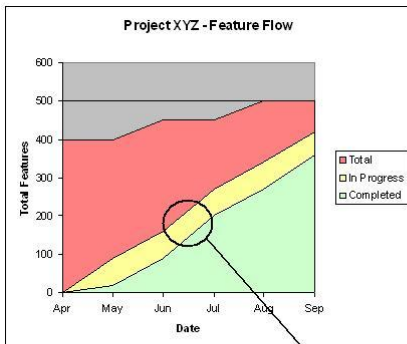


- Identifying constraints
- Buffering
- Managing flow

Cycle Times



Little's Law

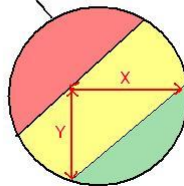


Little's Law:

Cycle times are proportional to queue lengths.

(We can predict completion times based on queue size)

Y = Queue Length (units)
X = Queue Duration (time)



Task Boards & Kanban



Summary



1. Use metrics wisely



2. Leverage the Hawthorne Effect



3. Measure Up



4. Favour Leading Metrics

More info and slides: www.LeadingAnswers.com

Email: Mike@LeadingAnswers.com