



BEST PRACTICE

Solutions Workshop Notes for

"Building Effective Measurement Solutions"

Date: 4/30/08

Facilitator: Nancy Kastl

Co-Facilitators: Barbara Ainsworth and Sharon Reinhard

Track Host: Luke Musinski

1. *What are the best testing metrics for management?*

Constraints

- Budget/Resources
- Different Data Collection Tools
- Time Constraints

Assumptions

- Data is available
- Audience is upper management
- System Testing completed

Success criteria

- Metrics needs to be reported to management in terms of dollars. Therefore, identifying the Cost of a Defect up front (i.e. \$1 if found in requirements, \$10 if found in testing, \$100 if found in production) would help to tie the metrics such as where found to a dollar amount that management could relate to.

Solution Approaches

- % Complete (Test Case)
- % Complete (over time line)
- Defect Density (per module)
- Find/Fix (Trending)
- Root Cause (where found/when)
- Pass/Fail Rate over Time
- Aging Report
- Avg. time to fix
- Code Coverage
- Reactivation (Retests fail)
- Schedule metrics
- Cost metrics (cost of a defect – more expensive if found in system testing or production)



2. *What is the best way to get the data for measurement?*

Constraints

- Getting accurate data
- Real-Time data
- Time-Frame for data collection
- Budget constraints

Assumptions

- Data is being collected
- There is a need to collect data
- We know the right thing to collect
- We know the right thing to collect

Success Factors

- Timeframes are reasonable and achievable
- Budget to allow the use of automated tools

Solution Approaches

- Training on Std. data collection process
- Training on Std. processes
- Centralized Repository for data
- Having the right tools
- Automate where possible

3. *What are the quality metrics?*

Constraints

- Depends on product, project, service, or process to be measured
- Need a common definition of quality
- Getting accurate data
- Having a sizing measure for software (LOC or Function Points)

Assumptions

- Want to measure the quality of production applications
- Will be able to define the quality characteristics that are important for the production application
 - Lack of bugs
 - Customer satisfaction
 - Meets requirements



- User friendly
- On time
- Within budget
- Software performance
- Maintainability
- Etc.

Success Factors

- Agreement to the definition of quality
- Agreement to relative importance of quality characteristics

Solution Approaches

- Identify if you are measuring the quality of a product, project, process, or service
- Define the characteristics/dimensions of quality for the item
- Rank the relative importance of each quality characteristic/dimension (e.g. is software performance more important than user friendly?)
- Determine the calculation or rating for each quality characteristic (e.g. customer satisfaction is a rating; user friendly is a score; meets requirements is a percentage)
- Determine if the data is available
- Determine the reporting (who, what, when, how)
- Pilot the metrics and refine based on experience

4. How do you create a simple and meaningful balanced scorecard/dashboard?

Constraints

- Data integrity
- Data source automated
- Management support
- Time and funding to develop all requirements for the balanced scorecard/dashboard

Assumptions

- Publish to a wide audience
- Not a Project Life Cycle – the focus is on testing/QA only
- Use prior and current data
- Publish the scorecard/dashboard weekly

Solution Approaches

Quality Perspective:

- Number of defects per project
- Number of defects per phase



- Coverage per project
- Defect causes

Financial Perspective:

- Number of defects post implementation
- QA budget
- SLA penalties

People Perspective:

- Training
- Certifications

Customer Perspective:

- SLAs met
- Interview results
- Post mortems feedback

5. How do you show that QA adds value and reliability?

Constraints

- Culture of finger pointing and blame
- No incentives for upper management to fund tool(s) or go through the QA Process
- Most times cannot prove value and reliability through testing
- No consistency on what to measure (I.e. project standards differ between small and large projects; can be treated differently but can be categorized)

Assumptions

Culture where senior management:

- Jumps to conclusions
- Does not understand
- Does not trust data or test methods
- Does not understand how metrics will help them

Solution Approaches

- Educate senior managers and obtain buy-in
- Establish standards
- Tools for report/data
- Gain consensus on defining what to measure and what to do when/with the data
- Learn and improve the process with data



- Vendor/outsources for best practice

6. *Bar Stool Problem: What are some alternate avenues, other than escalation to management, to ensure service level agreements (SLAs), in regards to fixing defects, are met by applicable parties?*

Solutions

- Ensure that the defined SLAs are reasonable by opening the lines of communication amongst applicable parties. This can help with understanding expectations and defining a common ground to stand on.
 - For example, is it feasible to expect severity 1 defects to be assigned a root cause in 24 hours and fixed in 48 hours? If not, then redefine SLAs with input from all applicable parties.
- Ensure that defects are assigned to a particular person and not just sent to a queue. This may help avoid defects being overlooked in the queue.
- If possible, assign a dedicated person from the applicable party whose sole responsibility is to address and fix defects.
- Promote a positive environment by rewarding resources that meet SLAs. This may encourage others to live into SLAs.
- Produce a daily report that is posted near the project team so everyone can visually account for project defect status. For example, use a chart that is color coded with green, yellow, and red to help identify defects statuses. This will help communicate without having to say anything.
- Have daily defect status reports meetings with applicable parties to address what was done yesterday, what is being done today, and what is slated for tomorrow.

References

Conference – Quality Engineered Software And Testing Conferences - Chicago
April, 30, 2008 (QUEST)

<http://www.qaiquest.org/chicago/>