

In-the-SPIN

Newsletter of the Boston  SPIN

Issue 23, January, 1999

Editor: Carol Pilch

Editorial

The January edition of *In-the-SPIN* features two articles on metrics. Appearing on this page, the SPIN Perspectives column provides return on investment data for technical reviews as well as some excellent references for technical reviews and measurement. Maxine Crowther is the contributor to this month's SPIN Perspectives column. This month's Feature Article, contributed by David Heimann defines the overwhelming sources of metrics available to projects along with a caution.

While we're on the subject of metrics, did you have the opportunity to attend the December meeting? If not, you can get some insight into what was discussed at the Metrics Roundtable during the December meeting in the Meeting Summary column. You may recall that the format for the December meeting was all roundtables. Other roundtables discussed process improvement, requirements management, and challenges in testing. General feedback for conducting a SPIN meeting with an all-roundtables format was very positive. Thanks to Caroline Starita for defining the format for the roundtables and organizing the sessions. I'm looking forward to future SPIN meetings with this format.

Consistent with the Boston SPIN charter, *In-the-SPIN* is provided by the Boston SPIN as a means of supporting the free and open exchange of software process improvement experiences and ideas. The steering committee encourages feedback on the newsletter as well as broader participation in the content and production of the newsletter. I'd like to hear from you. If you have an article you would like to publish in this newsletter, send it to carol.pilch@gsc.gte.com.

Boston  SPIN
Since January 1993
*Software
Process
Improvement
Network*

SPONSORS:

GTE Government Systems, Needham, MA

We thank the Computer Science department of UMASS- Lowell for providing support and hosting our Web page

SPIN Perspectives

This month's SPIN Perspectives article is contributed by By Maxine Crowther, Corporate Quality Manager, PAGG Corporation. Maxine serves on the Boston SPIN steering committee as secretary.



Benchmarks and Baselines for Technical Reviews



Over the course of the last year, I have spent a good deal of my time implementing a Technical Review program. During that period of time, I found and/or created a lot of benchmarks and baselines. I want to share these with the SPIN community to be used as you see fit. I also have a good bibliography for reference.

Benchmarks – From ASQ Software Quality Magazine, August, 1998

TRW – Inspections second only to unit testing for defect detection

HP – Inspections 3-5 times more efficient in terms of defects/hour than any other type of testing – 10 to 1 ROI

Motorola – as much as 40 to 1 ROI on requirements inspections

AT&T – 14% productivity increase; 10X increase in quality

Bell Northern – inspections 20X more efficient than testing

IBM – 38% reduction in defects.

Also reference this article for a best in class design and

Continued on next page

IN THIS ISSUE . . .

Editorial.....	1
SPIN Perspectives	1
Meeting Summary.....	2
Boston SPIN Calendar.....	4
Feature Article.....	6
Boston SPIN	7

implementation questionnaire for technical review processes.

Baselines – 18 months of experience

Average time spent/reviewer = 95 minutes for both examination and discussion

Average review cost = \$1025 (\$72/hour)

Average defects/review = 39

Average “High” impact defects/review = 7

I also tracked the cost/defect found. Based on the budget of the product validation organization, I was able to calculate that in the same period of time it was costing \$2000 for product validation to find a defect through testing and \$30 to find a defect through a review.

Another interesting graph to portray is the relative cost to remove a defect. Industry standards depict to a log scale graph where fixing a defect at the requirements stage costs virtually nothing all the way up to fixing a defect once product has gone to the customer of \$10,000! I have also verified internally that these numbers really do make sense. For some companies the cost to remove a defect once it is at the customer is more like \$250,000 especially if new media needs to be distributed.

Bibliography

Beizer, Boris, Software Testing Techniques, Van Nostrand Reinhold Company, 1990.

Freedman, Daniel and Weinberg, Gerald, Handbook of Walkthroughs, Inspections, and Technical Reviews, Dorset House Publishing, 1990.

Hetzl, Bill, The Complete Guide to Software Testing, John Wiley, 1988.

Humphrey, Watts, Managing the Software Process, Addison-Wesley Publishing, 1989.

Kit, Edward, Software Testing in the Real World, ACM Press, 1995.

Meeting Summary

Notes from the December Meeting

Compiled by Caroline J. Starita, SPIN Round Table Coordinator

Topic: “Product Development and Process Roundtables: Challenges in Testing, Requirements Management, Metrics, Process Improvement”

REQUIREMENTS MANAGEMENT ROUNDTABLE

Facilitator: Dolores McCarthy, Computer Sciences Corporation

Scribe: Rick Brenner, Chaco Canyon Consulting

The six people at this table called their issues on Requirements Management their “Burning Needs.” After much lively discussion and sharing of experiences, the group concluded that the most important things they learned from each other were the following:

* *Know who and what you are before starting to gather requirements.*

The organization that specifies or shares in the gathering of requirements needs to know what the business needs for the requirements are. Either marketing or technology may drive requirements. For example, the marketing department of one organization may be driving software requirements in hopes of gaining new customers, whereas in another organization, engineering may be driving requirements by developing a prototype to help an existing customer clarify their requirements. These are two very different points of view, and realizing this helps focus the requirements gathering effort in the right direction.

* *Using a tool to track requirements is important, but there is a lot of resistance.*

Good software tools are necessary for tracking requirements through the path from original system requirements to software requirements, design, coding, and changes in requirements in order to prevent unintentional loss or corruption of requirements. It would be very difficult or impossible to do this by hand, especially on large projects. However, there is a lot of resistance to such tools. The apprehension may be due to a lack of appreciation for the tool’s value, the cost of the tool, or time needed for training. The reasons need to be explored and resistance overcome.

* *Metrics help manage requirements volatility.*

It is important to gather metrics on proposed changes to requirements to show managers the impact of changes, that is, how many other requirements may be affected and how much more time and effort will be involved. This information allows managers to negotiate with the customer when costly changes threaten to cause schedule and budget overruns. The impact data allows the customer to see what more they need to fund or allow in schedule flexibility in order to have those changes. Impact analysis tools can help with the effort.

METRICS ROUNDTABLE

Facilitator: Maxine Crowther, PAGG Corporation

Scribe: David Heimann, Converse Network Systems

The Metrics roundtable discussed 3 topics -

- The Metrics Big Picture
- The Metrics-Friendly organization
- Requirements Metrics

* *The metrics big picture*

The metrics big picture contains 2 elements. First of all, you must make sure that there are objectives in place with which to associate your metrics by using a methodology like SEI’s Goal-Question-Metric. Secondly, you need to make sure that your metrics are balanced among:

Continued on next page

- product - the right things - features, defects
- process - doing the right things right - repeatability, schedule, risk
- customer satisfaction - determining the right things - surveys at multiple levels
- organizational capability - the ability to do the right things -
- employee satisfaction, training

This is the "Balanced Scorecard" approach from Harvard Business School. If you do not have these two things then you are measuring for the sake of having metrics. You will find over time that your metrics won't be important enough to analyze or review and the dreaded "So What..." question will keep popping up.

** The metrics friendly organization*

Hand in hand with developing your metrics is the challenge of creating a metrics-friendly culture. Some of the elements include goals that are cross-functionally established, reviews that are process-focused, action plans that are shared, and visible leadership. Only by demonstrating these elements can individuals see for themselves that the metrics are not imposed, that they will not strait-jacket them in performing their tasks, that fingerprinting is not tolerated, and most importantly that individuals will not be singled out based on performance. Metrics should measure processes and results, not people.

** Requirements metrics*

Lastly, we looked at what might be a good set of metrics for requirements. We looked at discovering requirements as being a process where customer reality and supplier technology come together to determine the best set of new features and fixes to existing features for a release or set of releases. Once these are determined then goals can be established. The correctness, non-ambiguity, consistency, usability, testability, traceability etc. of the definitions can be tested through the technical review process. A control board needs to monitor changes over time to ensure that schedules and resources are clearly defined. Choosing the right metrics is based on the objectives for the product or release.

PROCESS IMPROVEMENT ROUNDTABLE

Facilitator: Neil White, Aimware

Scribe: Carol Pilch, GTE

Areas covered by the Process Improvement Roundtable fall into 4 categories:

- Psychology of Process Adoption
- Sequencing Improvement Efforts
- Assessing the Progress of Process Improvement
- Process Improvement for Commercial Software Vendors.

** Psychology of Process Adoption*

Two aspects were explored: Motivation for Change and Resistance to Change. Among the motivators for adoption of Process Improvement initiatives were listed:

- Commercial calamity of some kind or another - "hitting the bottom of the barrel"

- A desire / business need to be more competitive to survive / thrive in the market
- Specific customer requirement
- A visionary leadership where the current market is not the primary driver (idea of Kaizen).

Reasons for Resistance identified included:

- Uncovering inefficiencies in existing processes leading to blame / concerns over recrimination / job losses
- Loss of personal power when individual's "proprietary" knowledge becomes more publicly available
- Lack of belief in the ROI on the part of management.
- Lack of patience/ long range vision on the part of management. Not prepared to stay the course without immediate results.
- Disillusionment on the part of employees re: "another doomed fad".

** Sequencing Improvement Efforts*

The recommended sequence involved:

- Document current workflow - this process evokes differences of opinion and acts as baseline for improvement.....visual graphing preferred.
- Identify target areas where people believe greatest improvement can be made
- Communicate responsibilities clearly for implementing the necessary changes.

** Assessing Progress*

The roundtable discussed the importance of verifying the efforts invested in Process Improvement have justified themselves (also needed as an encouragement!). Then the roundtable listed the various ways in which organizations evaluate progress --

- Informal: Reviews and Inspections, Quality Audits, Metric programs.
- Formal: Audits / Assessments from Customers, internal groups or external 3rd parties.

** Process Improvement for Commercial Software Vendors*

We seemed to agree that this is probably the least forgiving of temporary reductions in efficiency while on route to improvements in process capability. The roundtable explored the argument recently put forward by the Gartner Group that the mean-time to reaching Levels 2 or 3 on the CMM was 2.5 to 3 years, during which time, reorganizations or outsourcing will likely stymie efforts thus ensuring that efforts, for the most part, fail.

Lastly, the roundtable explored the happenings in the GNU / Linux world where the model for improvement is less evolutionary as it is revolutionary. The concept of open source code or "copylefting" vs. quality through informal inspections etc. was discussed. The roundtable and the jury is still out on this one but interesting!!!

CHALLENGES IN TESTING ROUNDTABLE

Facilitator: Caroline Starita, M/A-Com Wireless, A Division of AMP, Inc.

Scribe: Barbara Purchia, Kronos, Inc.

Areas covered during the Roundtable discussion fell into two
Continued on next page

categories: challenges in testing complex, poorly managed projects and challenges in automated testing.

* *Test Challenges in Complex Projects*

Discussion covered a broad spectrum of potential risk factors for testing complex projects. These included (1) multiple development and/or user organizations, with diverse geographic or international locations; (2) escalated schedules with concurrent stages of development, integration and system test; (3) test as a development surrogate; (4) customer mistrust about interim product deliveries during final fixes; (5) replication of dispersed, highly restricted user environment; (6) geographic segregation of development and test teams; (7) test software with many commercial packages (*shifting COTS packages, versions, platforms*); (8) performance testing without documented system requirements; (9) performance verification without well-defined target hardware configuration.

Roundtable Suggestions:

While no definitive solutions to these complex conditions could be provided, the participants made the following recommendations for risk management:

- Set customer expectations
- Treat different aspects of performance as bugs
- Make a bigger effort to set and manage customer expectations
- Make a baseline and track it
- Show progress (or lack thereof) and make it visible

* *Test Challenges in Automated Testing*

Attendees expressed distinct concerns about automated testing:

There is a need for structured test planning and requirements definition. Testing configurations tend to become distorted by the tests that are written versus the quality of the product. There is a need to perform QA on the tests and the test development according to plan. (*Who tests the tests?*) Insufficient attention is paid to efficient and effective automated testing. (*What is vs. what should be tested? What constitutes code coverage? Branch level testing? Testing large systems --- DB, networks?*) A common concern was the misperception of unit test reduction or elimination as an effective approach to reducing cost and schedule. Roundtable participants agreed that there should be tighter integration of the quality and development activities and that support people should be involved with testing. However, there was concern that over-reliance on automated testing might tempt management to push quality personnel into development. Also, the misperception of "too many" quality representatives might tempt developers to be lax in testing their own work.

Roundtable Suggestions:

Participants' discussion provided the following recommendations:

- Treat testing as a project:
Develop and implement an approved plan, with documented requirements traceability matrix, schedule, early definition of verification method: automated testing vs. testing which of its nature must be manual, etc.
- Keep developers accountable for the quality of their work

- Get support people involved in testing
- Shift quality roles from QA to QE (*Engineering*) : to provide big picture (*like QE architect*)
- Foster more testing upstream informally before formal testing
- Foster tighter integration of quality and development effort
- Ensure development and quality review of code before testing
- Provide training for unit testing, via formal courses or customer site visits. (*There may be a fear of not knowing how to unit test.*)

Boston SPIN Calendar

Information about Upcoming Meetings

by Johanna Rothman, Program Chair

January Meeting Announcement

Topic: "Optimum Software Testing"

Speaker: Bill Silver

When: Tuesday, January 19, 1998. 6:30pm-8:30pm

6:30-7:00 Networking and Round Tables

7:00-7:10 Announcements

7:10-8:10 Bill Silver: Optimum Software Testing

8:10-8:30 Questions and Answers

Who: Everyone (Academia, Government, Industry)

Location: GTE, 77 "A" St., Needham MA.

Abstract:

Bill will present a practice-based methodology for assessing true software engineering capability, and how to apply it to software testing. In addition to 12 metrics that measure the overall performance of the software testing life cycle, Bill will discuss how the assessment directly links test capability with test life cycle performance and overall corporate business performance.

This talk will use the assessment methodology to answer a number of questions:

- What is quality?
- How valuable are consistent processes?
- What does process improvement mean?
- What are metrics?
- What is the difference between QA and testing?
- How can we assess the effectiveness of assessment methodologies?

About the Speaker:

Bill Silver, a principal in Quality Fractals, has held executive positions leading software testing in several companies, including system vendors, software vendors, and IS/IT in Financial companies. The former publisher of Software Quality World magazine, Bill has published articles and taught seminars on software testing and metrics. He has conducted numerous test capability assessments and has hands-on experience in all phases of software engineering, including development, testing, and customer service.

Looking for Interesting Speakers

We are always looking for interesting speakers. If you'd like to speak at Boston SPIN, please review these criteria before sending us an abstract:

Speaker Guidelines:

1. Boston SPIN looks for relevant topics facing software groups who want to improve their processes. Particular relevance to recent advances/ changes in this field are particularly welcome.
2. Preference is always given to speakers who present information pertaining to actual experiences in the field as opposed to purely theoretical presentations.
3. Our membership attends hoping to learn how they can enhance their own results. We request proven, practical detail in your presentation.
4. The presentation should be based on the presenter's personal experience.
5. If you are a vendor or a consultant, remember that the most effective presentations are those where you explain your area of expertise and show how to be effective. Please do not use your time at Boston SPIN as a sales pitch.

We developed a speaker checklist so that none of us would have to rely on our short term memories. Please use the checklist to prepare for your SPIN talk.

Speaker checklist:

1. 60 days in advance of meeting: deliver 2 paragraph abstract, one paragraph bio to jr@jrothman.com
2. Within one week of meeting date: If desired, email copy of paper or overheads to heimann@world.std.com so that it is downloadable from the SPIN web page.
3. At the meeting: Speaker provides one copy of overheads to Charlie Ryan for our library.
4. Optional but highly recommended: bring 50-60 copies of overheads to SPIN meeting.

If you have information you'd like us to hear, please send an abstract to Johanna Rothman, jr@jrothman.com. Or, contact Johanna at 781-641-4046.

Monthly Round Tables

What: These are focus group or "birds-of-a-feather" sessions. They provide a professional forum for sharing information and experiences, for learning about other techniques, and for finding out that you are not alone.

Do you need or want to share information about handling thorny situations at work? Do you wonder what metrics are most important? Quality, scheduling effectiveness, time to

market...? Would you like to know how to manage a project that you have just been thrust into in mid stream? Could you benefit from leading edge approaches and innovative solutions for handling current project challenges? In an effort to elevate your organizational ranking from SEI CMM Level 2 to Level 3, are you in search of Lessons Learned from other survivors? Would you like feedback from the diverse backgrounds (Government, commercial, industrial, consultants) on topics related to your projected career moves?

Propose your wish list or questions as a Round Table and get your information from the movers and shakers in the software community. Round Tables are generally informal discussions, with a facilitator, to stimulate and moderate discussion.

A member of the SPIN Steering Committee will assist as Scribe for the discussion. Round Table proposals may be submitted by posting a sign-up sheet with the SPIN Steering Committee Round Table Coordinator, Caroline Starita (staritac@amp.com). Proposed Round Table sessions will be posted for sign-up prior to the monthly meeting in order for attendees to register their interest.

When: 6:30 - 7:00 PM, before SPIN Meetings

For further roundtable information, contact Caroline J. Starita, 978-442-4004 or staritac@amp.com or see the Boston SPIN web site, <http://www.cs.uml.edu/Boston-SPI>.

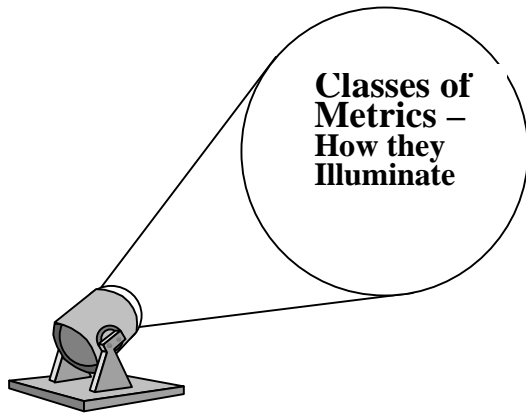
Future Program and Speaker Schedule

Date	Speaker/Topic
Feb. 25, 1999 Joint ASQ dinner at Holiday Inn, Newton (Thursday!)	James Bach "Good Enough Quality"
Mar. 16, 1999 @ GTE	Carol Pilch "A Tailorable Mini-assessment Method"
Apr. 20, 1999 @ GTE	Cem Kaner "Good Enough Testing or Bad Software"
May 18, 1999 @ GTE	Open
June 15, 1999 @ GTE	Open

Happy New Year

Feature Article

This month's Feature Article is contributed by David Heimann. David is a Member of the Technical Staff at Comverse Network Systems and serves on the Boston SPIN steering Committee as an at-large member.



When you drive your car at night, you put your headlights on in order to illuminate the road on which you're travelling, to be sure you're proceeding in the right direction and staying on the road. Likewise, in developing and maintaining software you calculate and use metrics in order to illuminate the landscape, to be sure you're proceeding towards your desired goals and staying on the path. This *SPIN Feature Article* discusses various metrics classes and how they illuminate.

The topic of metrics ranges widely, and many books and papers have been written on metrics (for example, *Software Metrics – A Rigorous and Practical Approach*, by Fenton and Pfleeger, and *Goal-Driven Software Measurement – A Guidebook*, by Park, Geothert, and Florac). Elsewhere in this *In The SPIN*, a summary of last month's metrics roundtable discusses several aspects, such as establishing a metrics-friendly climate and the overall metrics process (e.g., Goal-Question-Metric). In the interest of space and focus, I will focus this article on the measures themselves, particularly the various classes into which they fall.

Metrics tend to group into four categories and ten overall classes:

Product-related metrics classes: Product, Technical and Structural, Requirements

Process-related metrics classes: Project Management, Process, Testing

Quality-related metrics classes: Defects, Reliability

Customer-related metrics classes: Customer, Business

Product

Quantitatively describes the overall product, including features, intended use and intended users.

Examples -- Specifications, architecture, systems engineering, product performance, class of application, intended use, intended customers.

Technical and Structural

Quantitatively describes the size and structure of the software, including source code, object classes, and data sources.

Examples -- Size (lines of code), size (function points), complexity (cyclomatic, essential), coupling, cohesion, depth of inheritance tree, fan-out/fan-in, Halstead, relative database size (e.g., Boehm's DATA), execution-coverage profile.

Requirements

Quantitatively describes the requirements for the software product, their stability and management, and their traceability throughout the development and implementation cycle.

Examples -- Requirements correctness, completeness, unambiguity, testability, requirements stability and change activity, scope creep, requirements traceability.

Project Management

Quantitatively describes the software development as a project to be funded, tracked, completed, and delivered against a schedule and budget using organizational resources.

Examples -- Development risk, time to complete, cost, resources used, developer productivity, number and sizes of project activities, delivered code, delivered documentation, earned value, customer acceptability of delivered code and documentation, resource effort and cost of subsequent maintenance.

Software Process

Quantitatively describes the underlying process by which software is created and maintained, as well as the conformance to this process.

Examples -- Requirements management, requirements stability, requirements traceability, configuration baseline stability, conformance to procedures, number of deviations granted, audit results and closures of findings, reviews/inspections conducted (and passed), percent of training completed, conformance of processes to standards (ISO, CMM, IEEE, etc.).

Testing

Quantitatively describes the level and effectiveness of testing.

Examples -- Test strategy (including human and computer resources), degree of test planning completed, number of test cases, test coverage, percentage test cases passed, defects found, defects closed, phase in which defects found.

Defects

Quantitatively describes the defects in the software, particularly those reaching the user or customer.

Continued on next page

Examples – Defect density, customer-visible defects, defects per KLOC, defect removal effectiveness, defects distributed over components, defects distributed by root cause, defects distributed by phase (requirements, design, coding, testing, installation).

Reliability

Quantitatively describes the frequency with which software installation/operation failures and other incidents (especially shortfalls in performance) occur, as well as the outage, performance, and other impacts of these incidents.

Examples – Number of failures, mean calendar time between failures, mean execution time between failures, number of outages, mean outage length, system availability (i.e., uptime percentage), successful-use probability.

Customer

Quantitatively describes customer support, continuing engineering, and customer satisfaction.

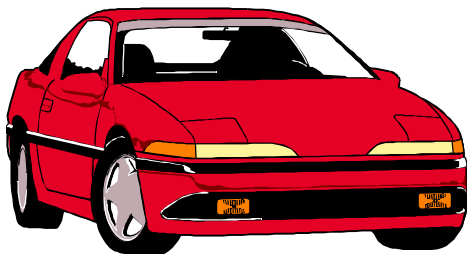
Examples – Problem-free installations and upgrades, number and frequency of support calls, effort and cost of support calls, patch (or maintenance-release) frequency, patch effectiveness, effort and cost of patches, product fitness for use by customer, customer complaints (or compliments!), responses on customer-satisfaction surveys, rate of repeat business.

Business

Quantitatively describes the business climate pertaining to the software product

Examples – Timeliness to market, degree of fit of product with other parts and with mission of the business, degree of fit with current and/or potential customers, overall customer satisfaction, employee satisfaction, organizational capability, R&D spending percentage, product revenue, product net profit (absolute and relative to revenue).

For an effective metrics program, one needs to incorporate enough of these measures so that they fully illuminate the landscape and the road, while also keeping the demand on metrics-related data, data collection, and compilation effort low enough that the battery can continue to deliver power. Happy and safe driving!



The Boston SPIN is a forum for the free and open exchange of software process improvement experiences and ideas. Meetings are usually held on third Tuesdays, September - June. Boston SPIN welcomes volunteers and sponsors.

For more information about our programs and events contact:

Charlie Ryan
ESC/DIB (Building 1704, Room 202)
5 Emlin Street
Hanscom AFB, MA 01731-2116
Telephone: (781) 377-8324
Email: ryan@sei.cmu.edu

For information about SPINs in general including ***HOW TO START A SPIN*** contact:

Dawna Baird of SEI (412) 268-5539,
dbaird@sei.cmu.edu.

IN THE SPIN is available on our Web page.

TO RECEIVE NOTIFICATION OF NEW ISSUES send email addressed to danallen@danallen.com. We have 2 separate email lists: one for this newsletter and one containing announcements that we receive from other process organizations and forward out.

IF YOU WANT TO BE ADDED TO THE ANNOUNCEMENTS LIST send email to ryan@sei.cmu.edu.

Send letter-to-the-editor, quips, quotes, anecdotes, articles, offers to participate in the newsletter committee, and general correspondence to Carol Pilch, carol.pilch@gsc.gte.com.

Send job postings to heimann@world.std.com.

Back issues and other information about Boston SPIN can be found at our WEB HOME PAGE:
<http://www.cs.uml.edu/Boston-SPIN/>

