

In-the-SPIN

Newsletter of the Boston  **SPIN**

Issue 32, December, 1999

Editor: Carol Pilch

Editorial

SPIN Perspectives

In this issue our November Roundtable Facilitators, Rick Brenner, Harry Joiner and Carolyn Starita, report on what the Roundtable participants had to say at the November meeting. For this month's feature article, Johanna Rothman has contributed one of her down-to-earth, common sense project management articles. And, if you missed the November meeting or would like a recap, I've summarized some of James Bach's thought provoking concepts on quality.

While I was working on the November meeting summary, I checked out Bach's web site where I discovered some papers written by Bach and a number of other folks including Johanna. This web site is really worth a visit. You definitely don't want to miss this one: <http://www.satisfice.com>

If you're a reader of this newsletter, the Boston SPIN would greatly appreciate your feedback. 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 Boston SPIN, and in particular the editor, would like to know if the readers' expectations are being met.

The SPIN steering committee also encourages broader participation in the content and production of the newsletter. Send letters-to-the-editor, quips, quotes, anecdotes, articles, offers to participate in the newsletter committee, and general correspondence to Carol Pilch, carol.pilch@GD-CS.COM.

This month's SPIN Perspectives column features summaries of the three Roundtable discussions conducted at our November meeting.

The following synopsis is contributed by Richard Brenner. Rick is Principal of Chaco Canyon Consulting and is an at-large member of the Boston SPIN Steering Committee.

The Impact of Split Assignments on Quality



When we try to split ourselves across two or more tasks, we usually find ourselves overloaded. Why do we so consistently overestimate our abilities for handling split assignments? Our roundtable explored this issue, and explored the causes of split assignments. Here are some of the insights we uncovered.

We can't blame management alone for split assignments.

- Some of us like split assignments—we get bored doing only one thing. Since some of us enjoy variety, split assignments are more fun. We seek them.
- Some of us have expertise that is rare in our organizations. That expertise might be needed on multiple projects. Often, management has tried to recruit others with these skills, but sometimes it's just hard to find such people. And some of us, as a job security strategy, have hoarded our knowledge.

Continued on next page

Boston  SPIN *Software Process Improvement Network*
Since January 1993

SPONSORS:

General Dynamics Communication Systems

Raytheon Company

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

IN THIS ISSUE . . .

Editorial.....	1
SPIN Perspectives-Roundtable Synopses	1
Meeting Summary	3
Boston SPIN Calendar.....	4
Feature Article	5
Boston SPIN.....	6

- Sometimes, it's a good thing to have something completely different to work on for a while. When we do, we can break from one task that has us stumped, go to work on something else for a bit, and return to the original task with a fresh perspective. Split assignments do keep us from getting stale, even though taking time off is often even more effective.

- Information technology also has a role in the increased prevalence of split assignments. Our accounting systems are now able to track complex assignment configurations with no additional cost, which has removed a constraint that kept assignments simpler in the past.

Split assignments have hidden costs, and sometimes those costs are very high.

- It takes between a half-hour and an hour to switch gears from one task to another.
- Sometimes we are constantly distracted by interrupts—we might start on one thing, only to be interrupted by another after a few minutes, only to be interrupted by another. Sometimes this chain can go on for a full hour, in which we accomplish absolutely nothing.
- Management doesn't really understand ramp-up time. People who are working on a task part time take a lot longer to learn what they need to know before they can be useful. There are two reasons for this. First, since they are only working part time on the task, it will take them longer to gather the information they need. Second, since they can't focus fully on the task, they might forget some things before they actually need them. When they finally do need that information, they have to relearn it.
- The cost of training a person who is working part time on a task is about the same as the cost of training a full time person. For example, when we train a person in the use of a procedure, it doesn't matter whether that person is part time or full time on the task. The more people we have per full time equivalent assigned to a task, the higher our training costs per capita.
- There is always the question "Who do I work for?" If I am half time on one task, and half time on another task that is led by my boss, I will tend to work harder on my boss's task. The half time I devote to the first task might not really be half time.

The following synopsis is contributed by Harry Joiner. Harry is with SRC/Titan Corporation.

SQA in Small Organizations



In small organizations (projects under 10-12 people and organizations under about 30 people), Software Quality Assurance (SQA) is not generally a full-time job and it is difficult to establish an independent reporting channel. Also, frequently, the

senior manager is involved on a regular basis with the daily operation of the project and, thus, has first-hand knowledge of any deviations from the standards and procedures.

The following scenario was presented for discussion by the participants:

An MIS project has 10 people working on it and reports directly to the Senior Manager, who works closely with the customer and the project leader. The group is organized informally.

The product is a database application, developed using visual tools. Test planning is performed by the Project Leader and the System Analysts. The product is prototyped, reviewed by the customer, and delivered in a series of versions until the product is essentially complete. The Senior Manager is entirely responsible for establishing and managing the relationship with the customer and for the technical performance of the project (as though this were a separate corporate organization).

The current quality assurance process is informal, but includes the Senior Manager's active involvement in and awareness of the project progress and decision making. He/she works directly with the Project and Software Development Leaders in managing the team's activities and assessing the results. He/she does not participate in all of the technical reviews, but reviews progress regularly with the team and individuals on the team. He/she approves verbally or in writing all significant exceptions to the normal procedures and makes the final decision for shipping the product to the customer. There is no separate SQA group within the overall organization or the project team.

One participant described an organization with a well-defined process that used project metrics to monitor the QA issues. Each of the 17 developers was responsible for following the defined process and reporting the results. There was no separate SQA group. Each of the team members was trained in the process and the metrics used to track progress.

Most of the remaining organizations that were described were struggling to define processes and, in some cases, rationalize the need for the process effort and SQA to the management team and developers. Clearly, one of the primary issues in most small groups was the need to tailor the development process for the project so that the "burden" of following the process did not bury the project in either documentation or red tape.

Caroline Starita contributes the following synopsis. She is M/A-COM Wireless Systems' Product Line Principal Engineer, SQA. Caroline is also the Networking Chair, responsible for the Boston SPIN Roundtable program.

What Should an SQA Group Be Doing, Anyway?

The majority of job postings requiring expertise in "Software Quality Assurance" deal with the performance of independent

Continued on next page

testing and requirements' verification. There are, however, many different combinations of SQA responsibilities. These range from that of objective scribe and secure data collection for formal milestone baselines to that of comprehensive lifecycle risk assessment and mitigation as part of an inter-functional management team.

Experience with Software Quality



The roundtable discussion began with introductions and the participants' experience with SQA functional roles and responsibilities.

Organizational interpretations included the predominant view of SQA as the independent test function, for product performance verification and recording. Another familiar approach was that of providing organizational records and reporting for compliance with industry standards and certification requirements (e.g., ISO 9000 or the SEI CMM status). This approach entailed procedural documentation and auditing, reporting and corrective action tracking until verifiable closure. Data collection was an integral function for performing trend analysis, identification of key metrics and assessing process improvement efforts.

The most comprehensive organizational interpretation of roles and responsibilities was that of SQA from the systems perspective. Implementation encompassed the assessment of software deliverables and reporting on acceptability of the documents (lifecycle and maintenance plans and manuals) and the software (code and coding standards, timing and sizing, etc.). Reporting would address actual performance to approved plans, inter-functional communications and schedule coordination, compliance with contractual and performance requirements, risk assessment and the adequacy of approved mitigation plans, process improvement metrics and progress.

Recommendations for SQA Implementation

There were some contributions that corresponded to successful SQA performance:

- The extent to which SQA organizations perform as the "eyes and ears of management" should be limited to talking to management's "bottom line" --- i.e., providing quantifiable evidence of reductions in resource requirements and/or process complexity --- thereby reducing time, cost and or equipment investment.
- Where personnel transition from software development to software quality responsibilities, there should be clearly defined tasks and working instructions. The new focus shifted from performing development activities to that of evaluation and reporting on

- (1) consistent and compliant performance to documented, approved plans and
- (2) appropriate configuration controls and management.

- When newly assigned to SQA management, the most important activity while becoming familiar with the new organization and policies is to identify and prioritize top management's "worst case scenario" risks. Focus the SQA efforts on implementation of risk mitigation plans, including the identification and collection of key metrics.

Consistent with many software quality meetings, the time allotted for discussion produced new data and greater understanding as well as identifying several items meriting further investigation. Most of the participants indicated interest in a follow-up Roundtable discussion.

Meeting Summary

Notes from the November Meeting

Contributed by Carol Pilch, General Dynamics

Topic: Good Enough Quality/Testing

Speaker: James Bach, Consultant

James Bach began his presentation by stating in his dynamic and energetic style: "Good enough quality is a controversial subject. I am a quality fanatic." At this point, if you have not read some of the many articles that James Bach has written on this subject or had the experience of hearing him speak, you might be wondering "what exactly is James Bach saying?" Using "good enough" as a modifier for "quality" just doesn't seem to flow given the usage of "good enough" in most contexts.

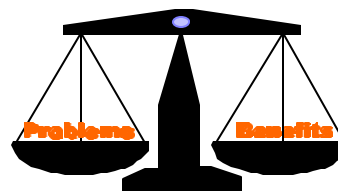
To appreciate the essence of James Bach's presentation, we really need to understand what he means by "good enough." Here's a quote from "Good Enough Quality: Beyond the Buzzword," James Bach, IEEE Computer, August 1997 that might help put this initial statement into perspective:

"Good Enough has nothing to do with mediocrity. It has to do with rational choices, as opposed to compulsive behavior."

At the recent SPIN meeting, James Bach went on stating that "quality is multidimensional." We need to think about "where do we have to be." We need to be somewhere between perfect and awful. Somewhere between these two extremes, there is a "good enough quality bar." The "good enough quality bar" floats. Bach asserts that being better than good enough is wasting resources.

Bach presented a heuristic for good enough:

In order to say that something (X) is good enough all conditions must apply:



- X has sufficient benefits
- X has no critical problems
- Benefits of X sufficiently outweigh problems

Continued on next page

In the present situation, and all things considered, improving X would be more harmful than helpful.

Bach wants the “good enough” standard to be a high quality standard. He **very** enthusiastically considers himself a “quality fanatic.” He encourages us to use the heuristic to create a dialogue about that which cannot be easily quantified.

Star Model Heuristics

Bach has defined the Star Model Heuristics to analyze a test process. With the test process at the center of the star, the five points of the star include: mission, requirements, test lab, test team, development. The mission is at the top of the star. The goal is to satisfy whatever the mission is. In this context, Bach advocates using checklists and asking questions to reveal information about the mission. Bach defines the mission of testing as:

- > Find important problems
- > Assess quality
- > Certify to standard
- > Fulfill process mandates
- > Satisfy stakeholders
- > Assure accountability
- > Advise about QA
- > Advise about testing
- > Advise about quality
- > Maximize efficiency
- > Minimize time
- > Minimize cost.

But, according to Bach, the answer is not this list. It’s situation dependent. The quality of testing depends on which of these possible missions matter and how they relate. According to Bach, many debates about the goodness of testing are really debates over missions and givens for the situation at hand.

Bach concluded with: “What I’m trying to get at with good enough quality is a means to persuade people to work better. I know it’s not good enough, but have you really thought about what good enough means?”

James Bach has a really excellent web site with lots of interesting and informative papers (by Bach and others) on this subject and related subjects: <http://www.satisfice.com>

Boston SPIN Calendar

Information about Upcoming Meetings
by Johanna Rothman, Program Chair

December Meeting Announcement

Topic: Product Development and Process Roundtables
Speaker: All of you
When: Tuesday, Dec. 21, 1999. 6:30pm-8:30pm
 6:30-7:00 Networking
 7:00-7:10 Announcements and Topic Selection

7:10-8:10 Roundtable Discussions
 8:10-8:30 Each roundtable reports back to the group
Who: Everyone (Academia, Government, Industry), no charge

Abstract:
 Do you need information about handling thorny situations at work? Could you benefit from leading edge approaches and innovative solutions for handling current project challenges? Are you in search of Lessons Learned from others? Would you like feedback from our diverse backgrounds (Government, commercial, industrial, consultant) on topics related to your projected career moves?

Propose your wish list of issues, volunteer as a facilitator, or just come ready to participate. We will select topics, engage the roundtables, and then report back to the group. To propose an issue, or facilitate a roundtable, please contact Caroline Starita at staritac@tycoelectronics.com, 978-442-4004.

Pre-meeting roundtables will resume in January.

Location: General Dynamics, 77 "A" St., Needham MA.
Directions: From Route 128 in Needham, take exit 19A onto Highland Avenue East. Take your first right by the Ground Round and take your second left onto "A" Street. General Dynamics is the last building on the right. Enter the parking lot by the General Dynamics sign and come into the building by the cafeteria entrance, which is located to the left of the main entrance. There will be a security guard at the entrance.

Info: See our web page, <http://www.cs.uml.edu/Boston-SPIN>
 For SPIN info, contact Johanna Rothman, 781-641-4046, or jr@jrothman.com

Cancellations (including weather cancellations): We will notify the membership via email to the SPIN distribution list by 3pm, post the notice on the SPIN web page, and announce the cancellation on Channel 7 TV and radio, WRKO AM 680.

***SPIN '99-'00 Program and Speaker Schedule
as of 12/08/99***

Date	Speaker/Topic
Dec. 21, 1999 @ General Dynamics	Audience Participation Roundtables
Jan, 18, 2000 @ General Dynamics	John Abbot “Requirements Sizing”
Feb. 15, 2000 @ General Dynamics	Johanna Rothman “Using Quality to Drive Project Lifecycles”
Mar. 16, 2000 Joint meeting with ASQ	Jim Driscoll “Ensuring Clients Achieve Superior Value in the Digital Economy”
Apr. 18, 2000 @ General Dynamics	Dolores McCarthy, Carol Pilch, Panel: “Process Maturity: Things that Work” Moderator: Donna Johnson
May 16, 2000	Paul Lanzoni

@ General Dynamics	"Technology Planning and Decision Making"
June 20, 2000 @ General Dynamics	TBD

Feature Article

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 Criteria:

1. The topic must be timely, an issue of concern to our membership.
2. We want to hear about real-world topics. If you have an academic topic, we're interested in how it applies to the real world.
3. If you are interested in creating a panel, please write an abstract, and suggest at least one panelist. We can work with you to find other panelists.
4. The topic should either explain how to *do* something, or bend our brains in another direction.
5. The presenter should be intimately involved with the "hows" of the thing that got done.
6. We are not interested in sales pitches.

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.

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, and picture 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 desired: Send a copy of overheads, paper, etc. for our web page as of the day of the meeting. If possible, provide 50-60 copies of overheads at the SPIN meeting. (The attendees really appreciate this.)
5. Optional: If you've written a book and are willing to donate it to SPIN, we'd be happy to offer the book as a door prize by conducting free drawing.



This month's Feature Article is contributed by Johanna Rothman. Johanna is president of Rothman Consulting, Inc. and is Vice-chair and Program Chair of the Boston SPIN.

It's Just the First Slip...

© 1999, Johanna Rothman



I recently read an article by a well-known author. He claimed that your first project slip isn't so bad; the third or fourth project slips are the bad ones.

In my mind, red flags went up. I flipped the bozo bit on the author. I completely disagree with his conclusion.

The first slip is your initial indication something is wrong. Don't expect you can make up time in your project. You can't. You can use the first slip to take a step back and observe what's going on vs. what you'd like to have going on. When you hit the third or fourth slip, you've lost the schedule battle.

When software projects start to slip, they're talking to you, the project manager. The first slip is a whisper: "Your expectation is not matching my reality. Listen to me, I can tell you my reality." If you ignore the first slip, the second slip is a murmur: "Things aren't quite right. Don't you want to know what's going on?"

At the third slip, the project says: "Knock-knock. Are you there? Don't you want to know what's going on?" At the fourth slip, the project yells: "Hey, you! You didn't listen to me when you could act. Now, get out the Tums. You'll pay for this."

I prefer to have projects whisper to me. (Otherwise people think I'm crazy when I yell back at my projects.) If you and your project agree on reality at early stages, you can make small adjustments with big results.

I recently worked with a company just before they planned to ship a Beta release. They were having trouble getting the software ready, and they wanted help getting the work done, so they could meet their Beta date.

I was ready with questions about the schedule, defect data, the testing, how the developers integrated the code. Luckily, we talked about schedule first. "Oh, we planned the schedule six months ago. We haven't changed it." I asked if they had met their milestone dates. "Well, not really. We missed the first

Continued on next page

deadline. The requirements weren't done, but we had to get started, so we started designing without knowing all the requirements." This is risky, but not a Terrible Thing, especially if they planned to manage the risks. I asked about the other milestones.

"Well, since the requirements weren't done, we couldn't finish the design on time. Since the design wasn't done, the coding was a little late." The first slip cascaded into slips for every other milestone. Then I asked what turned out to be the key question: "When did the testers know what to test, if the requirements, design, and implementation were a little late?" The answer I got was "Last week."

Uh oh. I asked one more question: "How much testing did you plan for this project?" They looked at each other, and said "Oh we planned to do about 6 weeks worth, but I guess we won't get to that now, will we?"

It's important to emphasize that these people weren't stupid. They had a simple problem with a huge cascading effect: the first slip led to more slips. Then they had trouble hearing the reality of their project. They started with a small slip, but because they kept going, the small slip magnified the effect of later slips.

If they had stopped at the time of the first slip, and replanned their work, or replanned the schedule, they might have been able to meet their hoped-for Beta date. Now their only option was to extend the schedule.

Let's assume that at the first slip, the project team had gotten together and said, "Hmm, the requirements aren't complete. Can we start design yet?" Sometimes the answer is yes, more frequently, the answer is a qualified "Yes, and then we have to review the requirements and the design together." Sometimes, if you haven't defined the requirements, the design work can't start, because you just don't know enough.

In this case, the project team could have started some design work, as long as they planned time for architecture and design reviews. They would have had to replan the schedule to make Beta, but it was achievable.

Maybe your project is different: you were able to define the requirements, and you started the design, *and* started coding and test development before the design was complete. This is common in concurrent engineering projects. How do you avoid cascading slips causing a missed date?

- Be ready to replan at any time. Use the original project schedule as a guide, but don't be afraid to change it when it no longer works for you.
- Replan when you first realize something is not going according to plan. Think creatively about what the project team can accomplish and when. You, as the project manager, may need to come up with some ideas, but involve the whole project team in reworking the schedule.
- If you know in time, you can choose not to include some functionality, if making the date is critically important.

If you can't effectively replan, you will have to slip the schedule or remove functionality. Slips tell you valuable information about your project. *Something is not going according to plan.* Before the something turns into lots of things, have a heart-to-heart discussion with your project.



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. There is no charge to attend the meetings.

For more information about our programs and events contact:

Charlie Ryan
ESC/DIJ (Building 1624, Room 2NE15)
5 Eglin Street
Hanscom AFB, MA 01731-2100
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, <http://www.sei.cmu.edu/collaborating/spins/spins.start.html>.

IN THE SPIN is available on our Web page: <http://www.cs.uml.edu/Boston-SPIN>.

TO RECEIVE NOTIFICATION OF NEW IN-THE-SPIN ISSUES and Boston SPIN specific notices 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 ADD YOURSELF TO THE ANNOUNCEMENTS LIST send email to ryan@sei.cmu.edu.

Send letters-to-the-editor, and general correspondence to Carol Pilch, carol.pilch@GD-CS.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/>