

In-the-SPIN

Newsletter of the Boston  **SPIN**

Issue 34, February, 2000

Editor: Carol Pilch

Editorial

SPIN Perspectives

The Boston SPIN would like to welcome its newest sponsor: **Edelman & Associates**. Edelman & Associates is an executive search and technical recruiting firm. Their web site is www.edeltech.com

Donna Johnson contributes this month's SPIN Perspectives column. Donna discusses the two different interpretations of the role of Software Quality Assurance in software organizations. This month's edition also has the summary of the January meeting in case you missed John Abbott's interesting presentation on requirements size units. And if you were unable to participate in a roundtable discussion at last month's meeting, you can find out what the participants had to say in the special Roundtable Report section. Roundtable facilitators, David Heimann, Dolores McCarthy, and Johanna Rothman provide the reports. If you want to know about upcoming meetings check out the Boston SPIN Calendar column by Johanna Rothman, SPIN Program Chair.

If you're a reader of this newsletter, the Boston SPIN would greatly appreciate your feedback. 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.

Boston  SPIN *Software
Process
Improvement
Network*
Since January 1993

SPONSORS:

General Dynamics Communication Systems

Raytheon Company

Edelman & Associates

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

This month's SPIN Perspectives feature is contributed by Donna L. Johnson. Donna is President of LOGOS International, Inc. and is the Boston SPIN's SEI Contact.



Author's Note: Since there have been many discussions at recent SPIN meetings and recaps in the newsletter about software quality, this article is an attempt to expand upon some of the ideas presented and try to put them in perspective.

Software Quality Assurance, or SQA, is a term that has varied connotations within the software community. Many companies have an SQA, or QA, organization, but that organization is more often than not, particularly in a commercial company, a test organization – the final gate through which a software product must pass before it is shipped. To see evidence of this widespread concept of SQA as a testing function, one need look no further than the Boston Globe Sunday Want Ads, wherein 90% of the SQA-related jobs are looking for someone to join a test organization.

Continued on next page

IN THIS ISSUE . . .

Editorial.....	1
SPIN Perspectives	1
Meeting Summary	2
Boston SPIN Calendar.....	3
Roundtable Reports	4
Boston SPIN.....	6

Few people will argue the fact that testing is a very important phase of the software life cycle, but, being the final step in the software development process, it is a point at which problems with the software are very costly to identify and fix. Many of you have seen the statistics that it is 10 times more costly to fix design errors detected in the coding phase rather than in the design phase and 10 times more costly again to fix those same design errors in the test phase. Moreover, there frequently is insufficient time to adequately test the software before delivery to a customer or before releasing a product to the marketplace. Software projects are notoriously late, and when a critical ship date looms on the horizon, it is usually the testing that suffers – by way of too little time spent testing and/or inadequate documentation available on what the product looks like and therefore what to test.

Due to these inadequacies of testing, the software development life cycle needs to include other quality-related activities to support the generation of quality software. A quality practice prevalent in many organizations is the conduct of peer reviews on work products produced during software development. Inspections and other methods of peer review have long been touted as one of the most successful means of reducing defects in a software product, and the conduct of these reviews has often been labeled as the Software Quality Assurance activity in an organization.

A more comprehensive array of quality activities in an organization, however, is found in the Software Engineering Institute's (SEI) Capability Maturity Model® (CMM®) for Software, which provides a framework for introducing quality practices, including peer reviews and testing, into an organization throughout the software development life cycle. Not only does the CMM provide guidance in the way of describing software quality practices, but it also provides a blueprint for expanding and enhancing those quality practices as an organization becomes more disciplined in developing software, i.e., as the organization matures its software development process. SQA is the cornerstone of the CMM's quality activities, and successfully implementing SQA practices on a project is integral to achieving CMM Maturity Level 2. The concept of Software Quality Assurance here, however, is very different from testing or the conduct of peer reviews – SQA is defined as "...reviewing and auditing the software products and activities to verify that they comply with the applicable procedures and standards."

The focus of SQA as the review and audit of software products and activities is also different from that of testing and peer reviews. The focus of both testing and peer reviews is the identification and removal of defects that have been introduced into the software during development. Reviewing and auditing software products and activities, on the other hand, serves to prevent defects from entering the software in the first place. The underlying assumption here is that if an organization consistently follows software development activities and processes that have produced quality software on past projects, the likelihood of producing a quality software product increases on subsequent projects (i.e., a repeatable process). Procedures, checklists, and standards provide guidelines to a developer for producing software based on known, successful methods, such that mistakes performed in

the past can be averted on future projects. SQA's job is to ensure that those processes are followed and that standards are enforced on the products. The SQA audits and reviews are conducted by someone independent of the project not only to ensure an impartial examination of the software activities and products, but also to provide a path independent of the producers of the software for resolving any issues pertaining to the quality of the product.

As we have seen, SQA means different things to different people. No matter what activities are labeled as SQA in an organization, however, a quality program should encompass multiple quality activities, and those activities should take place throughout the software development life cycle, not just at the end of the cycle. Testing, peer reviews, and process and product reviews and audits are all important -- they are integral components of a successful quality program.

Meeting Summary

Notes from the January Meeting

Contributed by Carol Pilch, General Dynamics

Topic: An Introduction to Requirements Size Units

Speaker: John J. Abbott, Abbott Consulting

John Abbott began his presentation by noting the problems associated with three other approaches used for sizing software projects: lines of code, function points, and testable requirements. Here's a brief summary of these problems.

Lines of Code

Developers are usually unskilled and uncomfortable with LOC estimates which tells us it may be an "unnatural" way to measure size. In addition, it is difficult to measure LOC consistently. There are different definitions for what to count (e.g., comment lines, physical lines, and semi-colons). Given today's development environments, the LOC approach is less meaningful. The approach has the potential to encourage people to write more resulting in inefficiency. Lastly, the LOC approach is not meaningful to customers.

Function Points

Function point estimating requires significant training for both developers and customers and many organizations are not willing to invest in the training. The effort required to count function points is significant. There is difficulty in tracking progress during the project when estimates are based on function points.

Testable Requirements

This method for determining scope requires decomposing requirements to the testable level. This may require projecting

Continued on next page

method assumes a certain level of domain knowledge exists in the organization.

Boston SPIN Calendar

Information about Upcoming Meetings

by Johanna Rothman, Program Chair

February Meeting Announcement

Topic: Using Quality to Drive Project Lifecycles

Speaker: Johanna Rothman, Rothman Consulting Group, Inc.

When: Tuesday, February 15, 2000. 6:30pm-8:30pm

6:30-7:00 Networking and Round Tables

7:00-7:10 Announcements

7:10-8:10 Johanna Rothman

8:10-8:30 Questions and Answers

Who: Everyone (Academia, Government, Industry), no charge

Abstract:

Companies create a variety of products, and different releases of those products, for many reasons. The goals for quality, schedule, and features for a project are intimately related to the reasons for creating the release in the first place. Since the definition of quality is determined by the goals you're trying to achieve, you can't use a cookie cutter approach to the project management process in software development.

Different project lifecycles provide unequal leverage for the three possible top priorities of software product development. We'll discuss how to choose the quality priorities for a project and then choose an appropriate project lifecycle.

About the Speaker:

Johanna Rothman observes and consults on managing high technology product development. She works with her clients to find the leverage points that will increase their effectiveness as organizations and as managers, helping them ship the right product at the right time, and recruit and retain the best people.

Johanna is the founder and principal of Rothman Consulting Group, Inc., and is a member of the clinical faculty of The Gordon Institute at Tufts University, a practical management degree program for engineers.

About Roundtables:

Roundtables are focused group or "birds-of-a-feather" discussions, with a facilitator, to stimulate and moderate discussion. Roundtables are held during the Networking

Continued on next page

what the requirements will be at the end of the project. The definition of "testable" requires an interpretation and using this method requires expertise in software testing. Most customers find this approach too detailed. The method presented, estimating using requirements size units is similar to this method.

Requirements Size Units (RSUs)

Advantages of using RSUs include:

- minimal training effort (as demonstrated during the audience participation exercise!),
- can be counted in a couple of hours
- meaningful to the customer since requirements are at a level the customer understands
- supports discussions of prioritizing requirements
- progress is tracked concurrently with requirements progress.

Method summary

Begin with a list of technical requirements. Technical requirements include "enterprise technical requirements" which are imposed by system architects (e.g., use Unix) and "derived technical requirements" which are determined by the development team. Derived requirements should be decomposed to the level of a requirement that takes one person month of effort to design and code. The requirements are organized into a 3-column table that includes the list of requirements, a column headed "RSUs", and a comment column. Schedule a 2-3 hour meeting for the software developers who best understand the design and code. At the meeting, identify those requirements that do not require code. These are assigned an RSU of zero. Using an RSU of 1 (takes one person month of effort), assign relative sizes to the remaining requirements. Requirements of size zero do require effort and are assessed separately.

John Abbott noted that RSUs do not include relative difficulty. This is not a concern because the method deals with laws of large numbers. Assumptions regarding complexity of tasks average out.

Tracking RSUs

Training involves recording changes to estimates of total RSUs during each project stage. Progress is tracked using percent of RSUs designed, coded, and tested. Tracking RSUs over the course of the project also provides a measure of requirements stability.

Considerations

RSUs are dependent on the skill and technology within the organization or development group. The organization and projects need to calibrate the meaning of 1 RSU by using historical data. The method works well in an organization that deals with similar types of applications. In addition, the

portion of the SPIN meeting. See our web page, <http://www.cs.uml.edu/Boston-SPIN> to see which topics are selected for this SPIN meeting.

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, post the notice on the SPIN web page, and send the cancellation announcement to Channel 7 TV and radio, WRKO AM 680 starting at 3pm.

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

Date	Speaker/Topic
Thursday 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 @ General Dynamics	Paul Lanzoni "Technology Planning and Decision Making"
June 20, 2000 @ General Dynamics	Steve Rakitin "Yellow Sticky Method of Project Scheduling"

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 a free drawing.

Roundtable Reports

The following roundtable synopsis is contributed by David Heimann. David is on the Process Quality Assurance staff of Converse Network Systems and an at-large member of the Boston SPIN Steering Committee.

Integrating Testing Into the Project Schedule from the Start

We aimed to identify effective steps that will cause testing to be included in the project schedule at the very start of a project, instead of being left for the end as is "traditional."

Continued on next page

We identified the following four key steps to making this happen:

1. Include requirements related to testing as an essential part of the requirements identification stage of the project. To this end, create and document testing requirements using the same procedures as those used for product requirements, and review the resulting testing requirements by representatives of Marketing, Engineering, Quality Assurance, and Customer Service. At the review meeting (or other mechanism) where the requirements are decided, have testers attend and be involved in the decision making.

2. For each requirement, match a test plan to it. Make sure all requirements are covered by a test plan.

3. Get management support to include testing aspects and requirements early in the project. In addition to other senior management, this especially includes the Development Manager. In view of this, the testing organization needs to routinely cultivate and maintain allies within the organization, especially Development.

4. Develop a test strategy at the start of the project. The strategy includes, at a minimum, the testing requirements (see above), the necessary resources, and the testing schedule (including calendar and staff-hour considerations). Make the test strategy a part of the overall project plan, so that approval of the overall project plan requires the approval of the test strategy.

The following roundtable synopsis is contributed by Dolores McCarthy. Dolores is a Senior Process Engineer with CSC and is Secretary of the Boston SPIN.

Getting to CMM[®] Level 2 ASAP



The subject of this roundtable was suggested at a previous meeting by a member who evidently was in a situation that mandated the achievement of CMM Level 2 as soon as possible (although I do not want to make a false assumption). The subject was intriguing and this roundtable set out to explore the whys and wherefores of reasons that might drive such a mandate and the realities of accomplishing the goal.

Most participants at the table had experience with

the CMM at their companies and had several ideas about what might be behind this goal:

- Management mandate
- Market competition
- Wishful thinking
- Inexperience with CMM
- Gaining more business
- Requirement by customer
- Creating an image.

No matter what the possible reasons, most agreed that anything less than eighteen months or so was an unrealistic timeframe to reach the goal. Two years was a more popular estimate. Even then, there needs to be a champion and sincere management commitment behind the decision. Most people diving in to the challenge do not anticipate the amount of resources, work, learning, and cooperation required over a sustained period of time to carry this out. Without it, the effort will die of malnutrition and neglect. Some at the table had experienced this. A company may wish for the glory of the achievement but not be willing to pay the price.

Adding to the difficulties might be

- Management delegating but not participating
- Inexperience with or reluctance to do planning
- Lack of procedures
- No training in the CMM
- Disappointment with progress toward the goal
- Inability to cope with problems that surface
- Denying needs for improvement
- Resistance to change.

Some at the table had experience with their company (or a part of their company) working on or moving on to Level 3, 4, or 5. The general message for others was that it can be done, but a company must understand and be prepared for what's involved before setting the goal and committing to it.

The following roundtable synopsis is contributed by Johanna Rothman. Johanna is president of Rothman Consulting, Inc. and is Vice-chair and Program Chair of the Boston SPIN.

Managing Software People



We had a diverse set of expectations for this roundtable. Some attendees wanted to talk about managing people strictly in the context of managing projects. Some people wanted to talk about how you manage software people in general. We decided to go with the general issues of managing people: how do you hire, set expectations, measure results, and move careers along. That was too much,

Continued on next page

so we focused a little on how to set expectations and assess results. We could have gone on for at least another few hours!

We started by listing the areas we wanted to discuss:

- Dynamics between Marketing, Sales, Engineering. How to get consensus on priorities.
- Is management like herding cats?
- How do you have requirements conversations with software people? How do you move to a more mature understanding?
- What are approaches to managing, especially in less mature organizations
- What happens in fast-paced, "internet speed" organizations?

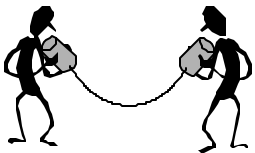
We started to talk about some of the problems we've seen in organizations:

- We can't talk about this. Some problems are not discussed in this company's context.
- The other part of our company knows how to do this, why can't software?
- How to measure/manage performance when the company is moving to software? Management and other departments don't know what we do. We don't understand them. How do we work together?

Some of the solutions:

Communicate (this was the only broad category we talked about), in these forms:

- Project meetings
- How long do people stay stuck?
- Assigning more tasks (inch-pebbles)
- Freedom to talk about capabilities
- Frequent one-on-ones



Our major take-away: The key to managing people successfully is to be clear on what you want done. Help people where they need help. Provide guidelines on how long they could stay stuck. Make sure you and they know what you want done (inch-pebbles). Make time to do one-on-ones for each person, and make project meetings problem solving meetings.

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/>