In-the-SPIN

Newsletter of the Boston SPIN

Issue 26, April, 1999

Editor: Carol Pilch

Editorial

This month's edition of In-the-SPIN provides three diverse topics. Our SPIN Perspectives column, contributed by Johanna Rothman, gives our readers a glimpse of what the future of software development and maintenance may have in store. Johanna describes work being done by Bob Laddaga of MIT and DARPA to research this topic.

Barbara Purchia contributes a comprehensive report on the March meeting. The subject of the meeting is a tailorng of the SEI's formal CMM based assessment method (i.e., CMM Based Appraisal for Internal Process Improvement or CBA IPI). The approach to tailoring provides and assessment with valid results and requires less time and fewer resources than the formal approach.

Rick Brenner provides this month's feature article that describes five things you can do when you're managing a project and you are faced with a problem or situation that has no obvious solution. Reading this article brought to mind Stephen Covey's "The 7 Habits of Highly Effective People, Powerful Lessons in Personal Change." In particular, the concept of "leader" vs. "manager." Ideas 3, 4 , and 5 in Rick's article are what Covey is alluding to when he describes the characteristics of the "leader."

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 and broader participation in the content and production of the newsletter. If you have an article you would like to publish in this newsletter, send it to carol.pilch@gsc.gte.com.

Boston SPIN *Software Process Improvement Network*

Since January 1993

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 Johanna Rothman, President, Rothman Consulting Group, Inc. Johanna is Vice-chair of the Boston SPIN Steering Committee.

By its nature, software is infinitely malleable. Software is easy to change. And, once people start using a piece of software, they can think of additional features for the application. To make matters worse, it is impossible to completely test a complex software system. Even if we could completely verify a system at one point in its development, the system will change.

So, wouldn't it be great if software could test and correct itself? Bob Laddaga, of MIT and DARPA is researching just this possibility. Bob and the principal investigators are working on creating software that understands:

- what it does
- how it does it
- how to evaluate and change its own behavior
- and how to respond to changing conditions

Continued on next page

IN THIS ISSUE . . .

Editorial 1
SPIN Perspectives 1
Meeting Summary 2
Boston SPIN Calendar 4
Feature Article 5
Boston SPIN 7

For software to do this, it would have to model its behavior and specifications. In order to change itself, the software would have to model its own structure, resources, and processes.

This is extraordinarily difficult to do. At a recent talk, Bob gave an example of this in an image processing application. Right now, the image processing application has some adaptation built in, but not enough to really recognize what's going on. Bob's suggested architecture has a parallel system continuously getting information from the application (testing) and bringing information to the application (updating and adapting). For those of you who remember the Artificial Intelligence push in the mid-1980's, this is a real possibility of using some of the techniques that were initiated then.

Some of the techniques Laddaga and his team are working on are:

- Domain architectures
- Protocol architectures (Open Implementation)
- Data-driven run-time dispatch
- Introspective languages
- Multiple run-time versions of modules
- Agent architectures

One of the questions Bob and his research team are struggling with is how much can they test the initial system and the parallel system responsible for adapting and correcting the original software? There's a real chicken and egg problem here. Some of the problems Bob has identified are:

1. How can the software respond in more than one way to a given set of inputs? How do you manage programs with multiple response capability? Right now, the essence of software is that it is deterministic -- for any given input, there is one, and only one, output. If we change this, do we know how to create or test the software?
2. Is there a low cost way to coordinate updates to multiple component versions?
3. Can we create self-correcting software that meets the performance and high assurance requirements of embedded, real-time applications.?
4. How can we measure and test adaptiveness, or robustness?

If Bob and his group can work out these problems, we can expect some benefits:

1. Swifter response for requested changes
2. Improved system performance.
3. The changes to the system are done with people who are very close to how the system works. In the same way that changes to a work process are best done by those doing the work, self-correcting software is as close to the work as we can get.
4. We can expect changes in the man-machine interface -- we could really talk to our computers!

This may be the first step in developing systems that modify and evolve behavior in cooperation with users. This would completely change how we develop and test software. For more information on self-correcting software, go to http://www.darpa.mil/ito/research/edcs/index.html .

Meeting Summary

Notes from the March Meeting

Contributed by Barbara Purchia, Kronos, Inc.

Topic: "A Tailorable Software Process Mini-Assessment Method"

Speaker: Carol Pilch, GTE Government Systems

One of the benefits of having a defined, repeatable and managed process is the ability to tailor it to meet your needs. Several years ago, GTE embarked on a Corporate Quality Initiative with a goal for software organizations to achieve Level 3 by the end of 1999. Our presenter, Carol Pilch, who has over 4 years experience assessing GTE organizations and is a Software Engineering Institute (SEI)-authorized lead assessor, was called in to help assess the various GTE software organizations using the SEI's Capability Maturity Model (CMM). What she and a co-worker discovered was that groups wanted more frequent assessments and wanted to know what to fix or do better. Although a future goal was identifying a maturity level, the immediate goal was continual improvement.

Carol began by asking the audience several questions about the CMM and assessments. About 75% of the attendees had knowledge of the CMM. 5-6 people use the CMM. 6 people had CMM assessment teams. 6-8 people were participating in CMM assessment teams. There were no other lead assessors at the meeting.

In order to understand how the assessment method was tailored, Carol first described the complete assessment method. An assessment is NOT an audit. It is a collaborative effort of an assessment team who appraises software practices relative to the CMM. An assessment looks at 4-5 projects in depth and includes other people in the organization. People on the projects fill out a questionnaire and the assessment team conducts interviews with project managers, project leaders, and practitioners. This provides a collective knowledge of the organization and the processes used. KPA goals and maturity levels are rated. The findings are provided identifying strengths and weaknesses as well as non-CMM issues that may have surfaced.

The objective of the assessment is to understand what is going on in the organization related to management and development practices. The assessment provides a framework for action such that detailed action plans can be developed. Full CMM based assessments usually occur every 2 years.

A full assessment requires a lot of preparation, not only from the organization being assessed, but also from the assessment

Continued on next page

team. There are 2 parts to an assessment: Pre-On-Site and On-
Site. During the pre-on-site period, the Lead Assessor works with the sponsor to identify the scope of the assessment, determine the projects, develop an assessment plan, and get concurrence of the sponsor to prepare and train the team. The assessment team consists of 6-10 software professionals, several of whom are from the organization being assessed. This team receives 2 ½ days of standard SEI training about 6-8 weeks before the assessment occurs. This time also helps with team building. The questionnaires are administered to assessment participants, and then the team examines them.

The on-site assessment consists of 11 steps starting with an opening meeting with the sponsor and ending up with a plan to attack the identified weaknesses. The sponsor holds a meeting describing what is happening and why. Then interviews are conducted with project leaders, middle management, and functional area representatives. After each interview session, the information is consolidated. Then draft findings are prepared and presented to the assessment participants to determine if anything was missed. This information is consolidated to create the final findings on strengths and weaknesses along with the KPA ratings to then determine the level. The final findings are presented to the executive sponsor and whomever management invites. A private session is held with the sponsor. The assessment is wrapped up. The assessment team works with the local software process group to help plan how identified weaknesses will be addressed.

GTE wanted to modify this approach to de-emphasize the ratings, to make the process less resource dependent, and to reduce the cost and impact to the organization. However, they still wanted to initiate or continue software process improvement efforts

The tailored assessment approach utilizes the key elements of the CMM Based Appraisal for Internal Process Improvement (CBA-IPI). The team consists of experienced assessment members. The process is tailored based on the organization's experience in process improvement and the sponsor's objectives. Risks are evaluated and mitigated. GTE has developed a set of reusable assessment assets consisting of:

- Mini-assessment planning/preparation checklist
- Mini-assessment schedule
- Document indices (a list of documents that assessors want to review)
- Consolidation tools
- Briefing templates
- Sample exploratory questionnaires (for each interview group type)
- Sample findings

In addition, each mini-assessment provides lessons learned for future improvements to the tailoring method and also have helped with full assessments. One of the key findings was that a full assessment could be done with fewer team members based on the number of key process areas examined.

Mini-assessments have been used at GTE for 4-5 years and have been well received. The process is well defined and is flexible enough to address specific sponsor and organizational objects. The results have been accurate and there has been strong organizational buy-in. Because of the team composition, this technique balances site knowledge and CBA-IPI method experience. This method has helped accelerate software process improvement at GTE with the following benefits:

- Stronger buy in due to frequency and visibility
- Insight into common pitfalls and trends
- Consistency and sharing of best practices
- Pilots to improve CBA-IPI efficiency
- Accurate results in less time that provide input to plans that keep organizations on track.

----------Questions -----

1. How may times have mini assessments been done?
   About 14 mini assessments have occurred.
2. Did a mini-assessment ever NOT include either Carol or the other GTE lead assessor?
   No, however this method could be used by any SEI-authorized lead assessor.
3. How reusable is this approach?
   This is a very reusable approach. Assessing risks is a key part of mini-assessment planning and it helps make the appropriate adjustments to the assessment process.
4. How repeatable are the results of the assessment if a different team was involved? Would you have the same results?
   The results are very consistent. The key is CMM knowledge consistency.
5. How do you bring the SEI process into a small company?
   Get in touch with someone who has tailored the CMM for small companies and organizations. Donna Johnson would be a good resource. (Someone in the audience mentioned that he had done it for a company with 8 people and that it works!)
6. How do you get an action plan that is doable?
   Planning is critical. Generally, you need to show the organization that you are serious about the results and improving. Identify short terms wins to demonstrate success, but also pick long terms goals. GTE has 2-year strategic improvement plans and 1-year tactical improvement plans.
7. Can you automate this process?
   Some organizations have tools. GTE is looking at a database of questions such that they can pull out questions as appropriate. However, they have not had lots of time to invest in many tools.
8. How frequently are full assessments done?
   A full assessment is done every 2-3 years with one mini-assessment in-between. Some groups had 2 mini-assessments in an 18-month period as risk mitigation for a full assessment.
9. How can we find a lead assessor?
   The SEI has a list of lead assessors. Local assessors would be willing to work with companies to tailor the assessment to meet their needs.

Boston SPIN Calendar

Information about Upcoming Meetings
by Johanna Rothman, Program Chair

April Meeting Announcement

Topic: "A Bad Law for Bad Software -- AND What We Can Do About It"

Speaker: Cem Kaner

When: Tuesday, April 20, 1999. 6:30pm-8:30pm

6:30-7:00 Networking and Round Tables
7:00-7:10 Announcements
7:10-8:10 Speaker's Presentation
8:10-8:30 Questions and Answers

Who: Everyone (Academia, Government, Industry)

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

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

Abstract:

A proposal to amend the Uniform Commercial Code with software specific rules will receive (what is supposed to be) final draft consideration this summer. If approved by the National Conference of Commissioners on Uniform State Laws (NCCUSL), the 200+ page proposal could reach state legislatures as early as this October. Because the bill is coming through NCCUSL (a highly influential legislative drafting organization that is largely funded by the state governments), there is a good chance that it will pass in many states even though it is a turkey.

Cem Kaner has been actively involved in the development of Article 2B (the new law for software contracting), as an advocate for customers and for higher quality software. His work on 2B has been discussed widely in the press, including the New York Times, US News & World Reports, ComputerWorld, InfoWorld, PC World, Network World, etc.

Kaner will provide an overview of Article 2B and its most serious problems, and we will discuss what it means for software quality and process professionals. To learn more about 2B before the meeting, check http://www.badsoftware.com (Kaner's site) and http://www.2BGuide.com (a publisher-side site).

About the Speaker:
Dr. Kaner holds a Ph.D. (experimental psychology), a J.D. (law degree), a B.A. (Arts & Sciences: primarily math & philosophy), and a certificate in quality engineering from the American Society for Quality. He is the lead author of TESTING COMPUTER SOFTWARE and of BAD SOFTWARE: WHAT TO DO WHEN SOFTWARE FAILS.

Round Tables - 6:30 - 7:00 PM, before SPIN Meetings

See our web page; http://www.cs.uml.edu/Boston-SPIN to see which topics are selected for this SPIN meeting.

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. GTE is the last building on the right. Enter the parking lot by the GTE 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.

Snow cancellations: We will notify the membership via email to the SPIN distribution list, post the notice on the SPIN web page, and announce the cancellation on Channel 7 TV and radio, WRKO AM 680.

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 is 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.

Continued on next page

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

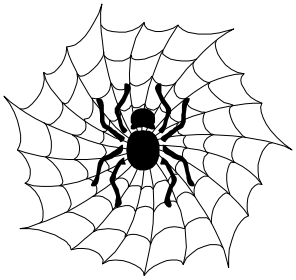
Date	Speaker/Topic
May 18, 1999 @ GTE	Chip Groder "GUI Testing"
June 15, 1999 @ GTE	Paul Lanzoni "Why Good Requirements Are Key to Successful Projects"

Feature Article

This month's Feature Article is contributed by Richard Brenner. Rick is a Principal with Chaco Canyon Consulting.

What To Do When You're Stuck

Copyright (c) Richard Brenner



Sometimes in project work there arises a problem with no obvious solution. When you've tried the usual approaches, and you've put your best people on it, and still nothing has worked, two factors might be responsible—hierarchy and precedent. The status hierarchy in project teams is usually based on competence and seniority. Precedents—the customary ways we do things—ensure that we use "tried and true" methods. Hierarchy and precedent are very helpful for everyday technical work, but they both work against you when you're stuck.

Relying on hierarchy often excludes those of lower status. Of two solutions—one by someone of high status, and one by someone else—we're more likely to choose the one with the author of high status. Similarly, relying on precedent often excludes new approaches. Of two solutions—one novel and one familiar—we're more likely to choose the familiar one. Normally, these are good conservative practices, but when the team is stuck, the usual way of working isn't working. It's time to look for an unusual way. Here are five ideas to help your project get moving again when it's stuck.

1. Focus your effort elsewhere

Sometimes, if you focus effort elsewhere, the problem might go away:

- A fresh viewpoint might be all that's needed. Do something else for a while.
- The problem you're seeing might be a state that arises only as a result of another problem, which, if repaired, makes the first one irrelevant. Taking some time might let you see the real problem.

Continued on next page

- The problem might be a conspiracy of simpler problems, which are more easily resolvable individually. It might just take some time for you to see this.
- Sometimes (too often!) sponsors or customers change their minds about what they want. Giving them time to change their minds could be all you need to do.

2. Replan part of the project so you can do other things for a while

Can you move the problem aside by replanning? If you can, you'll find a number of advantages:

- Replanning to take a problem out of the critical path reduces pressure and lets people think more clearly. It's the pressure of flowing water that keeps logjams jammed.
- Replanning makes it possible to do nothing for a while. See idea #1.
- If you apply effort to another part of the project for a while, you might learn something that could help you with the problem you've replanned around.

3. Generate new ideas deliberately

We all know about brainstorming. Here are four other approaches.

- Play "What Haven't I Told You?"

  Perhaps someone on the team knows exactly what's needed to resolve the difficulty, but doesn't know that they alone have this knowledge. If this happens, it might help to play a simple game, which I call "What Haven't I Told You?" Everyone in a small group (10 or 15) thinks of something that they know, but that they haven't heard anyone else talk about. For each round, everyone takes a turn telling an item to the group. Score the game in some way so as to give credit to truly arcane information.

- Use morphological analysis

  Morphological analysis, invented by Fritz Zwicky, is a way of studying the effects of combinations of factors [Adams 86]. To use it, list the relevant factors you want to consider in a single column along the left edge of a page. Then copy the same list across the top of the page. The intersection of each row and each column thus defines a pair of relevant factors. Next, consider each cell—meditate on it, and see what comes bubbling to the surface. You can apply this technique to the list of items generated in brainstorming or in a game of "What Haven't I Told You?"

- Find similar problems previously solved

  Look to a prior problem, and try a technique similar to whatever resolved it. What provided the key? Did you use brainstorming? Morphological analysis? Did you replan? Perhaps some combination? This sometimes works, but it can be dangerous if you're trying to think differently. If you can avoid that trap, it's possible to mine the solution of one problem for the solution of the current problem.

- Do something different

  To get something new, you might have to do something new. To increase the effectiveness of a brainstorming session or a game of "What Haven't I Told You?", conduct it in an unusual place—Mount Monadnock or a hotel. Or invite a local stand-up comic to facilitate. Declare a casual day (if you need to). If your organization already is casual, declare a Suit Day. Jiggle the status quo. Sounds strange, but it often works.

4. Suspend criticism of new ideas

New ideas are fragile and easily crushed. Here are some of the reasons why.

- If an idea is truly new, it has a small constituency of supporters—perhaps only one.
- The author of the new idea might not be high up in the hierarchy.
- A new idea might be difficult to grasp. Often, a new idea seems to conflict with cherished beliefs when, in fact, it is consistent with them.
- The idea itself might be incomplete, even though it is basically correct.

Since criticizing new ideas crushes them, find ways to suspend criticism of new ideas. If the team is discussing new ideas, try deliberately suspending criticism. Try a rule that if someone wants to comment on an idea, the comment must strengthen or extend it.

5. Distribute information—don't withhold it

Sometimes we withhold information that we think will retard progress, especially if it is bad news regarding availability of resources or some critical component. Withholding information almost always keeps you stuck. Distribute everything you know.

Even if you withhold information, *you* know about it. Do you really think you're so clever that the people around you have no clue that you're hiding something? If you are that clever, you should probably consider a career in espionage, and get out of project work—your talents are wasted. Moreover, in the future, it might come out somehow that although nobody else knew the bad news, you did, *and* you withheld it. What will happen to the atmosphere of trust when this fact becomes known? Hint: nothing good. It's nearly always better to tell the truth, to distribute the bad news.

One Last Thing: Celebrate Success

Lots of organizations hold celebrations when they reach a major milestone, and that's a good thing. But why not hold a mini-celebration when the team achieves a significant breakthrough? It's great for morale, and the next time they're stuck, people will remember that celebration. It could help keep morale up as they deal with the next obstacle. And you can be sure that there *will be* a next obstacle.

Reference

Adams 86 Adams, James L. Conceptual Blockbusting. San Francisco: W.H. Freeman, 1986.

Boston SPIN

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 Eglin 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.

TO ADD YOURSELF 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/