

# In-the-SPIN

Newsletter of the **BostonSPIN**  
Software Process Improvement Network

Issue 47

April 2002

Editors: Judi Brodman  
Sheila Lynch

## Editorial

Welcome to the April issue of *In-the-SPIN*. April is the month of new beginnings – Red Sox baseball starts, bulbs are blooming, plants are starting to stick their heads above ground, the sap is running, the temperatures are rising – and it is my birthday!

And for us at the Boston SPIN, April is our yearly joint dinner meeting with the ASQ featuring “An Evening With Tom DeMarco”. If you haven’t attended one of these joint meetings, you are missing a great opportunity to network with new and interesting people. See the detailed announcement on Page 10.

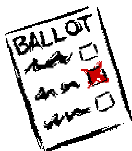
As you can see, Sheila and I are continuing with our changes to the Newsletter. This month I worked with Mary Sayre of Battelle to come up with a Newsletter template that will make our jobs a little easier – thanks Mary and thanks Battelle for allowing Mary to work with us! You can now use the table of contents to go directly to the subject matter of your choice.



Another addition that you see on our web site is the capability of ordering the featured monthly book club book through Amazon.com from our website (<http://www.bostonspin.org/>). Even if you don’t attend the book club Roundtable, add the books to your personal or organization’s library. Barbara Purchia chooses books that can change your professional and personal lives. The Boston SPIN receives a percentage of every purchase made by linking to Amazon.com from our site.



Remember that the SPIN Committee members’ elections are in June! There will be a number of positions opening up and we really need you to volunteer to serve this organization. SPIN continues to offer many services that are vital and/or interesting to our membership and all of them are the ideas and hard work of your Committee. We need “new blood” to serve on the Committee so that we can continue to have a healthy organization that grows terrific leaders and provides what our membership desires. Contact Michael Brother, [mdbrother@attbi.com](mailto:mdbrother@attbi.com) or Rich Green, [rgreen@cirelle.net](mailto:rgreen@cirelle.net) and put your name on the ballot!



Judi Brodman, Co-editor, *In-the-SPIN*,  
email comments to [brodman@logos-intl.com](mailto:brodman@logos-intl.com)

## Letter from the Chair

### Tom DeMarco talks about Slack

This month we have a joint dinner meeting with the ASQ where we invite Tom DeMarco back to the area.

In this era of downsizing, right sizing and layoffs, we all are more stressed and this program is more appropriate than ever. Tom is going to talk about Slack. One of the themes he talks about is the myth of total efficiency.

We are living in that error at this very moment.

There's an anecdote about saving money.

The story goes that a farmer wanted to save money so he starts to substitute sawdust for about 10% of the feed. The donkey seems happy and he's saved 10% so he continues to up the percentage of sawdust until it's at 90% of the feed and the donkey isn't looking so good but he decides to up it all the way and suddenly the donkey dies.

How much sawdust is getting fed in your company every time a layoff happens? How much "efficiency" can your company stand? Can your company continue to afford to live without middle management?

Questions to ponder when you join us for dinner with Tom DeMarco on April 25.

Regards,

Linda McInnis

Chairperson, Boston SPIN

**BostonSPIN** Established January  
1993  
Software Process Improvement Network

## IN THIS ISSUE . . .

Editorial.....	1
Letter from the Chair.....	1
Committee Spotlight.....	2
Speaker Spotlight.....	2
Feature Article.....	4
Dear SPIN Doctor.....	6
March Meeting Synopses.....	6
Upcoming Meetings.....	10
Order Your Book Club Books through SPIN.....	11
SPIN Information.....	11

# Committee Spotlight

## Steering vs. Repeatable

© 2002 Johanna Rothman



“We’re repeatable. We consistently and repeatably do the same stupid things over and over again.” – senior test engineer  
Process improvement staff emphasizes the importance of having a repeatable process. However, I’ve found that managers and others misuse the term “repeatable” to mean “not entirely chaotic.” In process improvement

terminology, “repeatable” means that the company has a documented product development process that includes project management, requirements management, and QA processes. Although you may have a repeatable process in the sense that your organization does the same things repeatedly, it may not be repeatable in the process improvement sense of the word.

Even in organizations that use the term “repeatable” correctly, there’s another problem. Some people view their repeatable process as a substitute for thinking. One project manager asked me this question:

“If we have a process, all we have to do is follow it, right?”

It depends. If you think you “just” have to follow a process, you’re repeatable. Repeatable can work for you, if you always do the same kinds of projects. However, if you work on different kinds of projects, including long R&D projects, small projects with quick deliverables, and iterative delivery projects then reconsider your process.

Is your process working for you on the current project? Has your process ever worked for you? Does it meet your needs for this project? I prefer to consider the needs of a particular project, before deciding on a process:

1. Do we know how to gather the requirements for this project? For example, should we use the design to iterate the requirements? Or, do we need to have the users define requirements in pieces, and then we’ll create the product in pieces *a la* an agile project?
2. Is our design process appropriate for this project? Are we going through extra steps because our process tells us to, although the design is as complete as it needs to be for this project? Do we need to iterate on the design this time? Alternatively, is the design complete, and we’re filling in the details with a series of projects?
3. What kinds of testing do we need on this project? How will we start testing at the beginning of this project? Do we need more exploratory testing this time? Do we need more automated testing this time? What’s different about this project?
4. Do our customers want a series of deliverables, or one major deliverable at the end? (Does that change how we deal with our customers?) Does the type of deliverable mean we should choose a different lifecycle for the project?

5. What risks do we have? Do we expect to have to replan this project sometime or several times during its lifecycle? Or, do we have only the Murphy’s Law risks, the ones we can’t anticipate?

When you start analyzing your projects and your process, you can understand what’s happening in your different project. You can then choose what to do for your current project. This continual analysis and understanding of your process is called “steering” by Weinberg in his *Quality Software Management* series of books.

Steering managers plan what they want to have happen, they observe what’s happening, and then take action if the desired effects are not happening. If you’re starting a project, consider the problems you found on previous projects. Do you want to change your project planning to reflect what you’ve learned from previous projects? A repeatable, routine process won’t prevent you from thinking, but consciously choosing a steering mindset can help you make better decisions.

If you’re starting a process improvement initiative, you may find that it’s easier to use your and your team’s problem solving abilities to decide on the appropriate lifecycle and project plan for a few projects, than it is to follow a process cookbook. Watts Humphrey, in *Managing the Software Process*, says, “Software engineering is not a routine activity that can be structured and regimented like a repetitive manufacturing or clerical procedure.” Once you know more about your projects, and how to make them successful, you can make a repeatable process something that’s appropriate for you.

Instead of focusing on being repeatable (as in repetitious), let’s think first, and steer our projects.

## Speaker Spotlight

### The Needle in the Haystack

Copyright © 2001 and 2002 by Tom DeMarco

Reprinted with permission.

The allegorical essay, “The Needle in the Haystack,” is adapted from the final chapter of Tom DeMarco’s new book, *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency* [Random House (Broadway Books Division) 2001 and 2002]. Tom has included this prefatory note to set the context:

One of the themes of *Slack* is a kind of diatribe against efficiency. It argues that efficiency may be nice, but that it often interferes with our ability to change and grow, that companies that are the most efficient are often the least flexible. I hope the book won’t give you the sense that efficiency doesn’t matter at all, only that it doesn’t matter *most* of all.

The fable of “The Needle In the Haystack,” which follows, is meant to provide a balanced perspective on the subject. It suggests that if efficiency wins you one point in The Grand Scheme of Things then a bit of inventiveness, ingenuity, risk taking and insight into human relations has the potential to win you three.

At the beginning of the fable, there is a simple and definite goal. At the end that goal has been achieved. That sounds good but for one tiny fact: the real goal has changed along the way. Will the protagonist, our hero, be agile enough to pick up on the new (much more exciting) goal and work effectively to achieve it? Don't count on it.

— Tom DeMarco  
Camden, Maine

## The Needle in the Haystack

There once was a young tailor who chanced to lose a needle in a haystack. He searched and searched for it, but to no avail. (In fact he searched for quite a long time, but I skip over this part in the interest of brevity.)

Being of a philosophical bent, he considered, as he searched, the abstract of the matter. He looked to see if this might be a single instance of a larger class of problem. Sure enough, he realized almost at once that the heart of his present dilemma lay in the superfluity of an obscuring element: The population of hay is large compared to the population of needles. Of course, there could be a boundless set of such problems, all of them at least abstractly identical to his own.

Let  $n$ , he thought, represent the number of needles, where  $n$  is equal to one in this case. And let  $h$  represent the number of discrete pieces of hay. Now, for  $h$  larger than  $n$ , the difficulty of finding the  $n$  can be arbitrarily high. Looked at another way, the probability,  $P$ , of finding  $n$  goes down as  $h$  increases. In fact, there is no number,  $\delta$  (no matter how small), that is not greater than  $P$  for sufficiently large  $h$ .

In proceeding through this logic, the young man was discovering what our later era would call the Goldblatt Conjecture, essential to the science of fractals and Mandelbrot sets. But of this, he was unaware. He was a tailor. While he had solved the relationship between  $P$ ,  $h$ ,  $n$  and  $\delta$ , he was no closer to solving the underlying problem, since he still couldn't find the needle.

As luck would have it, there happened to be knocking about that very haystack at that very moment a lovely young princess. Normally the tailor would have minded his own business, not even dared to raise his eyes to look her. But then he had an inspiration:

"Say," he said, "I see that you are a beautiful princess, and I know a thing or two about beautiful princesses."

"Yes," she sighed, "it's an occupational hazard. You aren't in the princess line for more than a few moments before everyone is thinking they know all about you."

"I know for example that a princess can feel, even through as many as seven mattresses, that someone has peed in the bed."

"Mm," she said. "You've got some of the details wrong there, but I guess you've caught the essence: we princesses are a sensitive lot."

Then he explained to her about the problem of the needle in the haystack.

"I think I see what you're getting at," she said. "You want me to lie down on this haystack and help you locate your needle."

"Right."

"I'll do it. We should have your needle in a trice. Now, where did you say you lost it?"

"Of that I am not too sure. But somewhere in this general area." And he waved his hands over an expanse of haystack encompassing very large  $h$ .

"Mm. Well, let's give it a try." With that she plopped down on a likely spot, wiggled slightly into the hay and let her eyes drift. "Nope. Not here. Let's give a try over this way."

She moved to a different part of the haystack and lay down again. Again her eyes wandered dreamily. The tailor felt a thumping in his chest. He realized he had never known *anything* about princesses before, at least not about this princess, not a thing. And now . . . Now his life was transformed, it could never be the same again. Where before there was mending and darning, now there was the possibility of Love. His days might be filled with beauty and enchantment, and dancing, and cuddling. That was the good part. There was a not-so-good part as well: Where before he had had nothing to lose but a dumb needle, now he had everything to lose. He felt a sudden panic. Oh, above all, don't blow it, he thought.

The princess was frowning. *His* princess was frowning. He felt his rosy new future slipping away. "Well, it's not here either," she said. "I can't understand it."

"No matter, really," the tailor said, too quickly. "Really. It's just a needle."

"But I feel terrible. I feel that I've let you down." She stared up at him sadly from the haystack. She was so beautiful.

"You mustn't, you really mustn't feel so bad. Oh please don't. Even the most sensitive person on earth could be overwhelmed by the sheer numbers here. You see, as the number of pieces of hay (we'll call that  $h$ ) increases compared to the number of needles,  $n$ , the probability,  $P$ , of finding  $n$  decreases monotonically without limit, and approaches arbitrarily close to zero."

The princess felt something thumping in her own chest. "Wow," she said. "You must be a poet."

"No, a tailor. But please don't feel bad that you have failed."

She sat up on the hay. "Failed? Not me. I just haven't succeeded *yet*."

"But it's too hard a problem. Please, put it out of your mind. I wouldn't want to be the cause of . . ."

"Nonsense. We'll simply reconstrue. The problem, to find a needle in a haystack, is clearly solvable, even without a princessly backside. All we need to do is to move the  $h$  to one side, and everything that is left is  $n$ . While  $h$  is admittedly large, it is not infinite. The formulation of  $P$ ,  $n$ , and  $h$ , which you put so prettily, is static, as it has no temporal element. If we consider instead the dynamic parameter  $P(t)$ , the probability of finding  $n$  among  $h$  within a time  $t$ , then  $P(t)$  increases monotonically toward 1.0 as  $t$  increases."

"Yes but what a  $t$ . We're talking eons here," the tailor said glumly. They might be ancient before they could ever get on to the Love part.

"So now we reconstrue again." She smiled contentedly at what was about to come, a bit of female logic to finish up the game. "Where you see here a haystack of order  $h$ , I instead

see a needlestack of order  $n$ . Lost in the needlestack are  $h$  hays.”

“But  $n$  is still one and  $h$  is still immense!”

“True, but now suppose we search for the  $h$  instead of the  $n$ . We find the hay lost in the needlestack. Now the numbers are working for us and we’re bound to succeed.”

The tailor breathed a sigh of relief. “Why couldn’t I have seen it that way? What a loser I am.”

The princess ignored him and lay back against the hay. “Hello, I’ve got it,” she said at once. She reached under her waist and produced a perfectly splendid piece of hay. “One  $h$ . I have succeeded after all. We have succeeded. And now we can get on to other things.” She looked up at the tailor, who was rather cute, in addition to being poetic.

But the tailor was staring down at something glittering in the hay. “My needle,” he said. And he picked it up triumphantly. He is a loser, she thought. Can’t find a hay in a needlestack, even when the odds are stacked in his favor. She shrugged and went on her way and he never saw her again.

## Feature Article

### Managing with Mini-Lifecycles

*Copyright © John Brti; John is an At-Large Boston SPIN Steering Committee Member*

When planning a software project, the lifecycle is always a consideration. When using UML, people think in terms of the workflows through the phases: the initiation, elaboration, construction, and transition, and the major iterations in each of these. This “macro”-lifecycle outlines the flow of the total project. Once the macro-lifecycle is established, a valuable next step can be to identify “mini”-lifecycles. Mini-lifecycles represent the flow of processes that repeat over and over during the project. Any repeating process offers the opportunity for establishing repeatable, predictable and optimized mini-lifecycles.

Mini-lifecycles can be found in most development projects - web development projects (where web page development went through a mini-lifecycle), object-oriented development (where each object went through a mini-lifecycle), and database development (where forms, screens, stored procedures, and multiple data migrations each went through their own mini-lifecycle). Identifying and optimizing such mini-lifecycles offer many benefits.

The reason mini-lifecycles are ubiquitous is the common characteristic of software development: code is decomposed into multiple components. These components might be web pages, forms or reports on the front end, objects or subroutines in the middle tier, or stored procedures on the back end. When there are many components of a particular type one can usually develop a mini-lifecycle, which can be followed during the development of each component. In some cases each component type can be given its own mini-lifecycle. In other cases it is useful to define a single mini-lifecycle that can be applied to all of the components - although some components may use only a subset of the mini-lifecycle

activities. This second situation is discussed next, as an example.

#### An Example

Let’s take a look at an actual development effort, a rather complex web site. It was a three-tiered architecture: ASP, over COM, over Oracle. There were complex business rules about personalization, permissions, authentication, privacy and security (the site handled money transactions). The team size during the construction phase was thirteen people, and the development period was seven months. There were over 250 identified components: web pages, web page subcomponents (such as reusable lists), middle tier objects, stored procedures, and documentation - yes, the documentation pieces were treated as components and followed the mini lifecycle! To facilitate incremental development and testing, the architecture and the development structure were very UI-page oriented. The mini-lifecycle we employed was in the design-develop-integrate portion of the project. The mini-lifecycle activities used were as follows:

1. Component is specified on a flowboard representation
2. Client approves component on flowboard
3. Wire frame page representation is prepared
4. Client approves wire frame representation
5. Page content is prepared
6. HTML representation is developed
7. Client approves HTML representation
8. ASP representation is developed
9. Code object is designed
10. Code object is developed
11. Stored procedures are developed
12. Database scripts are developed
13. Unit testing is completed
14. Independent code review is performed
15. Functional group passes QA
16. Integration test is passed.

(N.B., this list does not represent the sequence or dependencies among activities. They can be specified by process flow diagrams for each component type.)

As mentioned earlier, the activities in this list do not all apply to all component types. Each component type has a subset of activities applied to it. A middle tier object for example starts with activity #9 (code design) but would then follow the mini-lifecycle through to #16 (integration test passed) while skipping #11 and #12. In contrast, a complex on-screen menu with personalization and security concerns could involve visual representation, active page components, middle tier business logic, stored procedures and data related scripts. Such a component could make use of all of the mini-lifecycle activities.

As mentioned earlier, documents also went through this mini-lifecycle. Documents were handled -- by stretching the metaphor slightly -- passing them through phases #10 (preparation), #13, #14, #15 and #16.

We could have come up with a different mini-lifecycle for each component type, but believe that using a single tailorable mini-lifecycle led to better team understanding of the process flow, and facilitated the visualization of status, which will be discussed next.

Visualization of Project Status

We tied all of this together, the information about all 250-plus components and the mini-lifecycle activities, on a single “page,” a large spreadsheet matrix. A small representative portion of the matrix is shown below in the accompanying figure.

In the matrix, components were listed vertically and the mini-lifecycle activities were listed horizontally. The cells at the intersect between a component and a mini-lifecycle activity were color coded to show applicability and status. If the activity did not apply to that component the cell was shaded gray. For cells where the activity did apply color showed one of several states:

- white – not started
- yellow – in process
- green – complete
- red – failed and sent back

In addition to using shading to show status, cells with activities that were behind schedule were marked with a red asterisk, and activities that had been behind schedule but were now complete were marked with a black asterisk. In some cases where it was important, we also entered the actual date of completion in the cells, but in most cases we only tracked the component completion date, the date on which the component passed independent code review.

Many other pieces of information were stored on the matrix; examples include: component ID, associated file names, the functional owner of the component, the person(s) responsible for the component, dependencies on other components, scheduled completion of the component, estimated effort to complete, etc.

This matrix was updated twice a week. It was sent to a plotter

(it was about 18”x48”) and prominently posted in the team area. This simultaneously gave a color-based intuitive visualization of the high level status of the entire effort, and allowed individuals to see the status, responsibilities, and dependencies of any component down to the mini-lifecycle activity level. As team members completed activities they would update the status, directly on the posted matrix, using felt tip highlighters.

Benefits of Mini-Lifecycles

Some of the benefits of using this approach to mini-lifecycles include:

A one-page visual representation that communicates the tasks and status to the team, and the project manager.

A common well thought out set of mini-lifecycle steps for producing quality components.

A common understanding of the development steps among team members.

Intra-team expectations and flow are standardized, and handoffs are well understood.

Each activity can have completion criteria that eliminated the problem of people trying to skip a step. (Developers actually saw unit testing and independent code reviews as a part of their personal process!!!)

- The team gets into a healthy rhythm represented by the mini-lifecycle.
- The repeatable cycles lead to an accelerated and larger learning curve of improvement.
- It is convenient to have lessons learned between cycles allowing continuous improvement of the mini-lifecycle process.
- The fine granularity of the mini-lifecycle activities allows binary completion gates for individual activities, eliminating debates over what percentage of the work is complete.
- The binary status of activities can be used to generate an accurate aggregate percent complete and earned value

Item	Flow Board	Flow Board Cleint Apprv	Wire Frame	Wire Frame Apprv	Content	HTML	HTML Client Aprv	ASP	Code Design	Code	Stored Prcdr	Script	Unit Test	Indpnt Review	Functional Block QA	Integ Test QA
Pages and Forms																
Home Page	Green	Green	Green	Green	Green	Green	Green	Green	Green	Yellow	Gray	Gray	Red			
Report Suspicious Activity Form	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Yellow*	Yellow*	Yellow*			
Com Objects																
Message Object Server									Green	Green	Green	Green*	Yellow			
Sessions Object																
Documentation																
User Manual										Green				Yellow		
Site Schematic										Yellow						

analyses.

### Not an End to Gantt Charts

This approach to monitoring status did not eliminate the need for Gantt charts. We used Gantt charts (detailed to the component level, not the mini-lifecycle activity level) to track overall dependencies and critical path. Also, the issue arose as to whether to make the mini-lifecycle status matrix available to the customer. Our decision was not to. We limited customer status reports to courser granularity: milestones and major activities on a Gantt chart.

## Dear SPIN Doctor

**This one is too lite, this one is too heavy – this one is just right!**

©2002 Judi Brodman



Dear SPINners:

Recent articles lead us to believe that agile methodologies will help us to shed those stodgy old 'traditional methodologies' and processes, and address the needs of 'the new business environments' with newer,

'lighter' methodologies which somehow conflict with 'traditional process management'. For example, in their article, "Agile Software Development: The Business of Innovation" written in IEEE's Computer, September 2001, James Highsmith and Alistair Cockburn state "the question today is not how to stop change early in a project but how to better handle inevitable changes throughout its life cycle". They then state "Traditional approaches assumed that if we just tried hard enough, we could anticipate the complete set of requirements early and reduce cost by eliminating change". Since the term "traditional approaches" was not defined in the above referenced article, I'm assuming they were referring to the approaches and lifecycles of the 80's and early 90's.

Today, most projects build software incrementally using a spiral life cycle. Features/functions that make up these incremental builds are demonstrated to the customer in order to receive feedback. Feedback is processed, the feature software is updated, and the planning for the next feature(s) increment and demonstration begins. Estimates are computed for how long it will take to generate the software for the next increment and then, based on the feedback, new estimates are generated to cover the updates to the last feature that was demonstrated. Feedback from the customer occurs during the demo days – not "six months" later as Highsmith and Cockburn lead us to believe is happening today with most software projects. Projects are using this incremental (spiral) approach successfully to build e-based software. This process is agile, adaptive, and tailorable and more importantly, meets the needs of the customer and the developer.

In addition, today's software projects have Software Development Plans (SDPs) that contain the following information:

- ② the projects' life cycle,
- ② the planning and tracking process,
- ② the process for the capture and documentation of customer requirements,
- ② the metrics to be collected for each spiral cycle,
- ② the risks involved in the project and mitigation plans for these risks.

These plans are living documents that are updated as needed and are not lost as Highsmith and Cockburn say research shows. Constantly changing schedules and task assignments are maintained separately under commercially available Project Management tools. Revisions to artifacts such as the SDP, schedules and estimates are controlled under a CM tool that performs configuration management easily and maintains an organizational project and process repository.

The important point that articles written today do not make is that YOU need to be agile and adaptable. YOU need to decide what is too 'heavy' or too 'lite' for your project. YOU need to tailor your process to fit your project and your organizational goals.

Continue to embrace solid processes, lifecycles, and methodologies (and they are all different) that are adaptive, tailorable, and handle change. Concentrate on what customers have always wanted – a quality product that meets their needs.

This column is for you; let's make a difference! Send your comments and questions to "Dear SPIN Doctor" at [brodman@LOGOS-Intl.com](mailto:brodman@LOGOS-Intl.com). Sign them or use a "pen-name" - I respect your confidentiality.

"The SPIN Doctor"

## March Meeting Synopses

**"Software Process Improvement and e-Business:**

**An Oxymoron?"**

**Speaker: Donna L. Johnson,**

**President, LOGOS International, Inc. and co-author of the book, *The LOGOS Tailored CMM for Small Businesses, Small Organizations, and Small Projects***

*Dolores McCarthy, Boston SPIN Secretary and Quality Manager at Computer Sciences Corporation, Cambridge, contributed the following synopsis.*

This evening we were privileged to hear from our SEI CMM contact and one of the founders of the Boston SPIN, Donna Johnson, on the results of her interviews with 15 software managers, to find out how much process they use in implementing their e-business projects (e-projects). The managers were in IT organizations developing web pages, web applications, and e-Business applications

Donna explained that e-business can be recognized by its use of the Internet, an Extranet, an Intranet, or the Web for

purposes such as e-commerce, advertising, or communication. Characteristics of e-projects are: fast delivery; research-like, yet with mission critical needs; and the challenge of managing in a turbulent business and technology environment.

Given these characteristics, those in the e-business environment typically need to be light-footed, fast, and flexible in response to the demands of their clients and rapidly changing technology. Among the companies interviewed, the range of cycles for delivery is 6 weeks to 6 months and requirements are changing constantly. Some would think that the CMM would not apply to such projects, with its emphasis on process and the general perception that it applies to large DoD organizations. However, Donna found that many of the e-Business software managers she interviewed were using a lot of process. In fact, they had to use process in order to survive the demands and constraints of their e-Business. She explained to SPIN how using processes of the CMM could solve many of the problems others were having with their e-Businesses.

For example, e-Business has a worldwide, networked connectivity with a worldwide customer base. A quality problem in an e-Business company's deliverable would have high and rapid exposure to many customers, existing or potential. The effect of this exposure could be a company going out of business entirely, depending on the magnitude and perception of the quality problem. Reducing the probability of such a problem, Donna explained, requires process solutions, to improve product quality, such as the CMM's Software Quality Assurance, Peer Reviews, Software Quality Management, and Defect Prevention.

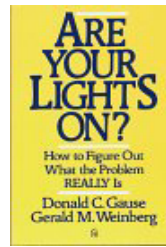
Another example where Donna showed the benefit of CMM process was in the crunch of compressed development cycles, causing insufficient testing, inadequate system analysis, and reduced project functionality. Some process solutions would be improved estimating, training personnel, hiring experienced managers, reduced rework, and increased milestones for tracking progress. All of these would fall under the CMM processes of Software Project Planning, Software Project Tracking, a Training Program, Peer Reviews, and Defect Prevention.

Donna went on to explain some CMM adaptations for e-Business. Projects require more planning up front to manage the shorter project cycles successfully. They need to plan for small, manageable units of work, increased risks, increased interfaces, increased commitments, and changing training needs. For project tracking and oversight, they need a re-evaluation of project tracking metrics, increased tracking of commitments and risks, reduced frequency of plan updates, and increased need for communication.

Donna summarized that CMM practices can improve e-Business software by applying them judiciously (e.g., "lightly") to the needs of e-Business for improved quality, reduced time to market, and increased productivity.

For Donna's complete presentation, link to the SPIN web site: [www.bostonspin.org](http://www.bostonspin.org)

## March Book Club



### Are Your Lights On? How to Figure Out What the Problem REALLY Is

By Donald C. Gause and Gerald M. Weinberg

Facilitated by Barbara Purchia, Rational Software

This book describes four steps to figure out what the problem REALLY is:

1. Identify the true problem
2. Determine the problem's owner
3. Identify the source of the problem
4. Decide whether or not to solve it

The book club attendees reviewed some of the highlights of the book and then tried to practice some of the techniques for each of the steps. We used some real life and hypothetical problems.

For step 1: Identify the true problem, we tried to think of three things that might be wrong with our understanding of the problem. We used a political problem; Find the leader of a terrorist group.

1. The problem is to eliminate terrorism.
2. The problem is to eliminate attacks.
3. The problem is to free the country that the leader has inhabited.

We also tried another technique; once you have a problem statement in words, play with the words until the statement is in everyone's head. We used an example in the book, "Mary had a little lamb," and tried placing emphasis on each word and showing how the meaning could differ. Try it yourself on another great example from the book, "Nothing is too good for our customer."

For step 2: Determine the problem's owner, we brainstormed on all possible owners for the situation many of us are in, people who got laid off.

1. Employees who were laid off
2. Employees who weren't laid off
3. Federal Reserve
4. Economy
5. Local government
6. Shareholders
7. Taxpayers
8. Managers
9. Company president/CEO
10. Direct supervisor
11. Federal government
12. Marketing
13. None of the above
14. All of the above

For step 3: Identify the source of the problem; "The source of the problem is most often within you." In this case we used the example that some big software company releases new software and it won't work in your current environment. We talked about possible solutions to this problem.

1. Don't upgrade and run with the old software. Who knows what was fixed or is now broken.
2. Buy a new computer.
3. Buy another computer to run the new software.
4. Upgrade your computer with a faster CPU or get more memory or disk capacity.
5. Buy competitor's software.

Although we didn't get to the next item, the book has some sage comments.

For step 4: Decide whether or not to solve it; "In spite of appearances, people seldom know what they want until you give them what they ask for." And the wonderful comment, "We never have enough time to consider whether we want it, but we always have enough time to regret it"

It's no wonder this has become a cult classic, its insights are as powerful now as they were when the book was first published in 1982.

## March Roundtables

### Process Improvement Roundtable - How to persuade your sponsor, management chain, peers and subordinates

*Facilitated by Judi Brodman, LOGOS International, Inc.*

We began this Roundtable at 6PM and we were adding new arrivals until 6:30PM. We had a large group participating in the discussions.

We first did a little statistical analysis on the attendees and found out the following:

Number of software developers in the organizations:

<20	20 – 30	30 – 50	50 – 100	100+
18%	0%	18%	27%	36%

Level of maturity of the organizations:

Level 1	Level 2	Level 3	Level 4	Level 5
64%	36%	0%	0%	0%



Everyone took a few minutes to introduce themselves and to state why they were attending this Roundtable. Many of the attendees were thinking of initiating an improvement program in their organization. Other attendees were there to listen and learn. Discussion ensued on a number of topics as described below.

#### Should the organization be using the CMM or the CMMI?

Most of the attendees who were involved in process improvement were using the SW-CMM. They were interested in whether they should transition to the CMMI or stay with the SW-CMM; and when they should think about transitioning to the CMMI. It was agreed that the CMMI should be reviewed and the organization should evaluate if any of the new Key Process Areas (KPA)s or new practices in existing KPA)s would add value to the project and/or would address

improvements that needed to be made in the organization, then those KPA)s should be addressed.

#### Where does ISO fit into process maturity?

We discussed the latest document produced by the SEI that stated that ISO certification mapped to a lower Level 2 maturity.

#### When are SEPGs/PATs useful and what should their makeup be?

SEPGs (Software Engineering Process Groups) were found to be useful even at Level 1 when the process improvement program was just starting up. Members should be well respected by the organization and staff so they can "spread the word". Formal PATs (Process Action Teams) were found to be more effective in the larger organizations than the small organizations.

#### What training is useful?

Larger organizations felt that it was important to have staff members trained in the CMM and assessment methodologies so they can appraise themselves. Small organizations usually didn't have that option. Attendees felt that training in Project Management was important as well as training in Configuration Management and defect tracking and analysis.

#### How do you start an SPI program?

Attendees felt that some sort of an appraisal would be useful to the organization so that the organization could prioritize and focus on software areas needing improvement. A CBA-IPi was thought to be too expensive and too extensive for small organizations to start with.

#### Requirements specification – how much is enough?

How much specification and formalization of requirements was necessary was thought to be dependent on the customer, project size, and whether the requirements are being handed off to someone else to build.

#### Should the Process Roundtable continue?

Attendees agreed that this Roundtable was useful and they elected to meet again and continue to discuss solutions to the problems that attendees brought up. Therefore this Roundtable will meet next in May. Please join us!

#### Job Seekers Roundtable

*Facilitated by Michael Brother*



Anyone who thought unemployment was not really a problem with software professionals or that there are plenty of jobs for the "good" candidates learned differently at the Job Seekers Roundtable. An overflow crowd gathered to discuss their experiences looking for work in this new, new economy. People with impressive

credentials reported being out of work for nine months or more. Everyone felt that recruiters and on-line job boards were not working for them. This is a time to learn new techniques for your job-hunting tool kit.



Not everyone at the roundtable was looking for work. One participant bravely identified herself as a recruiter and gracefully responded to the frustration and complaints from the other participants. We found that recruiters share many of the same problems as the job seekers: a large pool of qualified candidates, a small pool of open positions, and very high standards for obtaining an interview. Recruiters do have one advantage over most job seekers - a strong network that includes hiring managers.

Most people agreed that networking should be part of their job search. Some experts claim that up to 80% of jobs are found by networking. Opinions were mixed on different networking organizations in the area. Some felt that job-hunting networks were a waste of time, just a group of unemployed people complaining to each other. Others found the groups very helpful and supportive. We had a lively discussion about networking and the job-hunting groups with a very good exchange of ideas.

The most amusing exchange was when a participant asked if personal business cards were helpful and was instantly showered with cards from around the table. Everyone agreed that cards were as helpful in the job search as a good resume. You can get good quality cards free on the web, for a nominal fee online, or from the local office supply superstores. Many people print their own on business card stock so they can customize the cards for specific opportunities. Sheets of business card stock for laser and ink jet printers are available at any office supply store. Custom cards lead to custom resumes and cover letters.

Our resident recruiter helped us see the HR perspective and how to increase the chance of getting an interview. Resumes are reviewed very quickly, usually 15 to 20 seconds per resume. You need to quickly show that you have the key skills the job requires, making a custom resume for each job very helpful. Don't stretch the truth, but do highlight the skills requested in the job posting. Cover letters had mixed benefits. Recruiters and HR often ignore the cover letter completely. When the letter is read, it gets the same quick scan as a resume.

Obviously, the Boston Software Process Improvement Network (SPIN) believes in the benefits of professional networking. SPIN launched a Hiring Initiative with this roundtable and a networking session scheduled for April 10. We are exploring ways to help our members find jobs, including having a job seekers roundtable at each meeting as long as people are interested.

Check the Hiring Initiative page at the Boston SPIN web site (<http://www.bostonspin.org>) to learn more.

## **Retaining People in an Uncertain Economy Roundtable**

*Facilitated by Johanna Rothman, Rothman Consulting*

Original description: When the economy is good, it's easy to retain staff with monetary incentives and rewards. However, when money is tight, and salaries are holding even, what do you do to retain staff? Do you change your management style? Do you use non-monetary rewards, such as appreciation dinners, training, and conferences? Do you use something

else? Bring your questions, what you've tried, and we'll discuss how to retain technical staff in an uncertain economy.

We came up with a list of things to remember:

1. Respect is critical
  - a. People want to know that as a manager, you'll listen to their opinions
  - b. People also want to know that you value their ideas, and that you'll use them. (Please don't think that as a manager, you have all the answers).
2. Understand the value of the work people do
3. How does my work contribute to EBIT (Earnings Before Income and Taxes)? (Many technical people, especially those whose organization has been bought or sold recently, keep an eye on earnings, to make sure they're helping the company make money.)
4. Don't be afraid to do formal and informal recognition.
  - a. Informal respect and recognition:
    - i. Keep meeting times
    - ii. Respond to email
    - iii. Don't be stingy with "atta-boys/girls", even if you don't use the software
  - b. Formal recognition:
    - i. Create a "Star" program, with multiple levels. Use credits that can be applied or traded in on corporate clothing or for money or stock.
    - ii. Look into the book: "1001 Ways to Motivate Employees That Don't Cost Much..." (Note: I didn't find that book, but I found this: [1001 Ways to Reward Employees](#) by Bob Nelson.)
5. Make sure your corporate culture believes that retaining people is an asset to the organization, and that you work for an ethical company. Too often, retention is not interesting to management.
6. T-shirts are still useful for recognizing a team's efforts.
7. Beware of solo or team recognition that leaves people out. If you don't recognize some people who've done a great job, public recognition backfires.
8. If someone's done a great job, ask them if they want more challenging work. (Not everyone wants more of a challenge.)
9. Peer pressure (good people help you retain good people.) We had a question: how do you help people enter the system?
10. Tailor recognition to the individual.
11. Make sure the people doing the recognition know who's done what work.
12. GROW THE PEOPLE WHO WANT TO GROW.

# Upcoming Meetings

Thursday, April 25, 2002

## "An Evening With Tom DeMarco"

### Joint Meeting with ASQ-Boston and Boston SPIN

ASQ-Boston and Boston SPIN present an evening of networking and fun. Well-known business and fiction author **Tom DeMarco** will discuss material presented in his latest book: *Slack, Getting Past Burnout, Busywork and the Myth of*



*Total Efficiency*. He will also be available to sign copies (which may be purchased at the meeting). Tom is an engaging speaker with interesting, real-world stories to share. His career began at Bell Telephone Laboratories and his lecturing and consulting experiences

have taken him around the world. His recent interests include project management, change facilitation, and litigation of software-intensive contracts. Tom's credentials include a BSEE degree from Cornell University, an MS from Columbia University, and a diploma from the University of Paris at the Sorbonne. He is a member of the ACM and a Fellow of the IEEE. He makes his home in Camden, Maine.

Please bring your business cards and a friend, if you wish, for a great evening of networking, learning and entertainment!

#### Meeting format:

Hors d'oeuvres buffet will be served during the networking session from 5:30pm to 7:15pm. Tom will begin speaking at 7:30pm. Space is limited -- reserve early!

**Reservations:** Please provide the name of each person requesting a guaranteed reservation, with company affiliation and a contact phone number. Call 617 755-ASQC (755-2772), 24 hours a day, or [make your reservation online](#). The deadline for reservations is **April 22, 2002**. If you must cancel, please do so by the deadline, otherwise you will be billed for the dinner. We welcome non-members.

**Cost:** \$25.00 (\$15.00 for students, and those unemployed or retired)

**Directions:** The Sheraton Four Points Hotel is located in Waltham at 420 Totten Pond Road. From I-95/Route 128, take Exit 27A. The hotel is just East of the highway. The hotel's phone number is (781) 890-0100.

## May 21, 2002 –

### The Politics and Process of Software Estimating; Why Are We Always Late?

#### Tim Lister

Software estimation has proven to be "challenging," but to be frank, software estimation has proven to be a nightmare. Most organizations that develop software have lost all credibility with their clients when it comes to simple questions like "When will you be done?" and "How much is it going to cost?" In this fast-evolving industry, one thing is clear: Time hasn't given us a chance to improve our estimating skills.

In this presentation, Tim Lister investigates the estimation issue, offers some suggestions, and promises to come up with some surprising answers to the title question.

Tim Lister is a principal of the Atlantic Systems Guild, Inc. He is a member of the Airlie Software Council, a group of industry consultants advising the DoD on best practices for software development and acquisition. He's also a member of the Cutter IT Trends Council and co-author -- with Tom DeMarco -- of many popular course sequences, as well as *Peopleware: Productive Projects and Teams*. Their tentatively titled book, *Software Risk Management*, is scheduled to print in 2002.



Tim is a member of the IEEE and the A.C.M. He also serves as a panelist for the American Arbitration Association, arbitrating disputes involving software and software services.

**Book Club:** *Peopleware (Productive Projects and Teams)*, by Tom DeMarco and Tim Lister.

#### Roundtables:

**Boston SPIN Software Testing Forum** Leader/Host: Paul Piper

**Process Improvement Forum** – Facilitator: Judi Brodman  
Problems, solutions and lessons learned

**Job Seekers Roundtable** Facilitator: Michael Brother  
**Managing with Mini Lifecycles** Facilitator: John Brtis

## June 18, 2002 –

### Achieving CMM Level 2 and Beyond

#### Judi Brodman and Steve Hannigan

Steve Hannigan, the top manager of the Battelle Cambridge Office (BCA), a small organization with diverse projects, discusses how his organization achieved Level 2 and many goals in Level 3. He discusses what was needed (resources), how long it took, and what lessons were learned. Judi Brodman, the consultant who guided BCA, presents her perspective on how the organization progressed to Level 2 and the approaches she applied to help them achieve their goal.

#### Book Club

*Surviving the Top Ten Challenges of Software Testing (A People-Oriented Approach)* by William Perry & Randall Rice

#### Roundtables:

**Boston SPIN Software Testing SIG** Leader/Host: Paul Piper

Combined with Book Club

**Process Improvement Forum** Facilitators: Judi Brodman and guest

**A Minimalist Approach to Initial CMM Implementation** Facilitator: H. S. Lahman

**The Personal Software Process** Facilitator: J. Erik Hemdal  
A Few Unexpected Lessons  
(Follow-up to RT of Dec-01)

**SPIN Elections!** 

**NEW**

## Order Your Book Club Books through SPIN



Boston SPIN is bringing you more convenience for your book club selections. You can now order Book Club selections and your purchase benefits SPIN. We receive a percentage of every purchase made by linking to Amazon.com from our site.

Book Titles are linked to their reviews on Amazon. Go to the This Month's Meeting or Book Club Schedule pages on our website where you can read an abstract of this month's Book Club. From there it is just a single click to the Amazon.com website to purchase it!

## SPIN Information

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. Additional information about the Boston SPIN can be found at our WEB HOME PAGE: <http://www.bostonspin.org/>.

For more information about our programs and events contact Barry Mirrer, Program Chair, [bmirrer@alum.mit.edu](mailto:bmirrer@alum.mit.edu).

### Cancellations (including weather)

Starting at 3pm, we'll notify you via email to the SPIN distribution list, we'll post the notice on the SPIN web page, and we'll send the cancellation announcement to Channel 7 TV and WRKO AM 680.

### SPIN Meeting Location

Boston SPIN meetings are held at The MITRE Corporation in Bedford.

Please be aware that MITRE has advised us that, due to increased security concerns, you will need a Picture ID for admission to the SPIN meetings. We encourage you to leave all carrying bags, backpacks, and briefcases behind (i.e., in your car). Otherwise, you should be prepared to have these opened and inspected upon arrival.

MITRE's campus is located at 202 Burlington Road (Route 62), Bedford. SPIN meetings are held in the 'S' building. Directions can be found on our Web site: <http://www.bostonspin.org/>.

## Sponsors

The following organizations/individuals support the Boston SPIN:

- ② The MITRE Corporation <http://www.mitre.org/>
- ② Raytheon Company <http://www.raytheon.com/>
- ② Edelman & Associates <http://www.edeltech.com/>
- ② Quality Search <http://qualsearch.com/>
- ② UMASS – Lowell (provides support)

## Email Lists

To receive Boston SPIN specific notices, send an email to: [jwithall@onebeacon.com](mailto:jwithall@onebeacon.com)

## Future Programs

We welcome your suggestions for future Boston SPIN programs. Program suggestion forms can be found on the Boston SPIN web site. We are always looking for interesting speakers. If you'd like to speak at Boston SPIN, please review the criteria specified on the Boston SPIN web site before sending an abstract to Barry Mirrer at [bmirrer@alum.mit.edu](mailto:bmirrer@alum.mit.edu).

## Newsletter Call for Articles

The *In-the-SPIN Newsletter* is always in need of new and interesting articles dealing with Process Improvement, software development methodologies, Project Management and other related subjects that may be of interest to our readership. Please send any articles that you would like to have considered for publication in the Newsletter to either of the editors listed below.

Send letters-to-the-editor, and general correspondence to:

- ② Judi Brodman, Co-editor of In-the-SPIN, [brodman@logos-intl.com](mailto:brodman@logos-intl.com)
- ② Sheila Lynch, Co-editor of In-the-SPIN, [salynch@mitre.org](mailto:salynch@mitre.org)

Back issues of the *In-the-SPIN Newsletter* can be found on the Boston SPIN web site: <http://www.bostonspin.org/>.

