SEPG
FEDERAL AVIATION ADMIN
INTERFACE
NEWSLETTER OF THE SOFTWARE ENGINEERING PROCESS GROUP
VOLUME 6, NUMBER 3 • AUGUST 1997

WHAT THEY DON'T TEACH YOU AT SCHOOL

© by Geoff Mullery, UK

A criticism of university training is that it does not require the student to deal with realistic problems. Those responsible for training search diligently for realistic problems on which students can practice the techniques they are taught, but there is a fundamental flaw in the belief that a small practical example can be found that will adequately illustrate a real life problem.

To give you some insight into the problems a method user faces, I will take the "lift" problem ***[editor's note:*** *a "lift" is the English for elevator****]*** and illustrate what happens in the real world (although, fortunately, rarely all on one project). I emphasize that I have seen multiple examples of each complication I add. At each stage the instructor should ask himself two questions:

- how would my method have dealt with the situation?
- could I have built that into a small training example or otherwise addressed it?

An elaborate definition of the lift problem is unnecessary. All that is required is a system to allow users to call a lift and get it to deliver them to a floor of their choice. For a typical 'real' complication, assume that there are ***m*** lifts in ***n*** different locations in the building (***m*** > ***n***) and several companies are involved in developing the system. The contracts are separate—there is no prime contractor for the whole.

This may not be sensible, but your first lesson in reality teaches you that there will always be a strong demand—which frequently sounds eminently plausible at the time—that you do at least one thing which is not optimally sensible, if not wholly irrational. The next thing reality teaches you is to discard the notion that people are prepared to cooperate, that the only problem is that they don't properly understand what is needed. Mistake!!!

On a big project some people have their own agenda. They either aim **not** to cooperate or aim to gain power by using the work activity to inflate their importance. They will deliberately tell you things which are not entirely true. Maybe not lies as such, they are just economical with the truth—not unlike second hand car tradesmen!

In many projects you will meet the "she's too busy to talk to you" syndrome. You may not be allowed to talk to lowly lift attendants or maintenance engineers. Your customer fears can't possibly articulate the requirements "properly," and so provides you with surrogate users—user "authorities." Truth is not necessarily what the customer wishes to convey—a good image is often more important.

User authorities are either people who used to work on that kind of thing (when it was done via a large hand crank rheostat—the kind you see in old movies such as ***Grand Hotel***) or who aren't busy at the moment and once used a lift. These people are either ignorant of how ignorant they are, and accidentally mislead you, or are only too clear about how ignorant they are and try to avoid telling you anything, in case they might be blamed later.

Then there are your technical colleagues. On a big project some come from another department or another company. You are a rival. Some don't believe in this method—some don't believe in **any** structured or formal method. Each such person will make the most of the weaknesses of your method and attempt to diminish positive results by any subtle (and frequently not so subtle) means—obfuscation and obstruction are their chief tools.

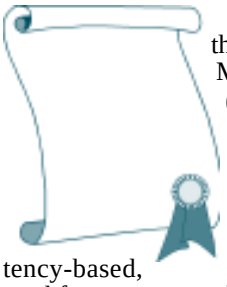
Be sure that if your company bid a realistic price and schedule for the project, they didn't get the contract. If you are doing the contract you can be fairly confident that little short of a miracle will allow you to meet your schedules and that there is insufficient funding. Even at this stage the **lift** is not the problem your method most needs to address.

Your customer has a computer services department who are very keen on touch sensitive displays. They have developed a library of display management software and you are required to use a display system (for lift selection panels)

continued on page 4

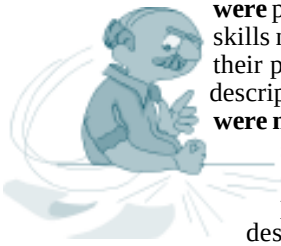
COMPETENCY DEVELOPMENT AND ASSESSMENT INITIATIVE AT THE FAA

by Jim Kimball, ASU-10



FAA submitted the new Acquisition Management System (AMS) to congress for approval on April 1, 1996. Part of the AMS promise was to provide a competency-based, skilled acquisition workforce, supported by a comprehensive learning system for ensuring that important skills could be provided to the work force as needed.

ASU-10 and ASU-250 formed a software skill assessment team that developed and implemented a competency - based assessment process that focused on the individual job series. The initial survey of the ARA work force showed that in many cases individuals



were performing skills not defined in their position descriptions and **were not** performing skills that **were** defined in the position descriptions.

In April of 1995, the program office requested the Quality Division, ASU-200, of the Office of Acquisition in the ARA Line Of Business to develop a plan for providing highly trained and experienced software quality assurance personnel for technical support on software intensive projects. At this point it became clear that it was pointless to try to map job activities one for one with the job series. The demands on the work force were too dynamic.

Changes in methods of oversight made many skills called out by existing job series unneeded. New skill requirements were being identified at a rate faster then the personnel system could accommodated. We needed a system that could respond rapidly, effectively, and efficiently to grow the required competencies. Instead of the current approach, it was

determined to survey the specific skill needs of ARA, group them into "roles" based more or less on job tasks that personnel were actually performing, and then make a rough map of roles to job series—at least until we could replace the relatively inflexible job series approach.

Having identified a number of roles, it was now necessary to provide a plan for (1) determining who in the work force possessed the requisite competencies and to what degree, and (2) developing a program to ensure that those personnel identified as candidates for the needed roles received necessary training or other skills enhancement.

Having defined a software quality assurance role, the computer specialist branch of the quality division ASU-250 developed a career development transition plan that was consistent with the software quality activities called out in the Software Engineering Institute's Software Capability Maturity Model. The Competency Assessment and Career Development Training program developed by the Army Materiel Command Software Task Force, chaired by the Defense Acquisition University was used as a model.

The ASU-250 computer specialist branch then conducted individual panel interviews of approximately forty field quality assurance representatives in order to assess their competency levels for the defined skills. A baseline was established which identified the skill gaps across the division. A gap analysis was conducted to determine the needed learning opportunities to raise the level of the division software quality capability to a suitable minimal level, as well as to respond to the program office request.

Where are we today? The ARA Office of Business Management (ABZ) was formed in March of 1997 to extend and generalize these efforts beyond ASU to all of ARA.

ABZ began by identifying the core services provided by ARA. It then identified the roles associated with each core service and the competencies needed to successfully perform in that role. Currently, they have defined some 25 roles and expect to have about 40 roles defined when done. The members of the Intellectual Capital Investment Plan Council, composed of the deputies and chief scientists, have provided the names of above average performers for each of the roles defined and ABZ is currently overseeing the generation of competencies for each of the roles based on the input from the above average performers.

During August and September, a competency assessment, or survey, will be completed by all ARA employees and their managers and team leads. The survey will be distributed, completed, and collected electronically and will be accessible via the Intranet. (Details to follow.) Finally, at the end of this year, the assessment data will be analyzed and the results of the analysis distributed.

Shortly after both the employee and his/her manager completes the assessment by the end of September, the employee will receive a report showing the employee's responses as well as that of his/her manager or team lead. It is important to note that this report will not be made available to the manager and that the employee's responses will be treated as confidential. The intent is that the employees use this assessment data to help prepare their individual development plans. ■

*inter*FACE

is published quarterly by SEPG

DOT/FAA/AIT-5
800 Independence Avenue, SW
Washington, DC 20591

Chief Scientist for Software Engineering
Arthur Pyster (202) 267-8020

Editor
Norman Simenson (202) 267-7431

www.faa.gov/ait/sepg/news1.htm

GOVERNMENT LESSONS VIA THE INTERNET

The General Services Administration is moving its 1000 by the Year 2000 training program online. The program is designed to prepare the next generation of federal systems managers, practitioners, and program officials who use information technology and manage information. Information Resources management (IRM) covers many disciplines, including computing, communications, information management, software engineering, and records management.

GSA has agreements with colleges and universities around the nation to offer federal employees courses in information technology planning, policy, and management. The Carnegie-Mellon University, Columbia University, and University of Maryland University College will offer graduate-level courses online for GSA's certificate program. Universities award an IRM certificate after successful completion of six graduate-level courses. This can now be accomplished through the On-Line IRM Certificate program.

For the new online curriculum, students will rely heavily on videotapes and the World Wide Web as research and study tools. ■

For more information about online study options, contact Annie L. Barr at (202) 208-2780 or Yvette C. Gibson at (202) 501 3970. **Internet:** *annie.barr@gsa.gov* **or** *yvette.gibson@gsa.gov. See also* **http://www.itpolicy.gsa.gov/mkp/1kby2k/1x2intro.htm.**

SEPG LEARNING RESOURCE GROUP

by Susan Hermanson, TRW

What is a SLRG? Another name for pond scum? A new law out of Congress? Another automated system? None of the above. SLRG stands for SEPG Learning Resource Group. Previously the group was called the Training Working Group (TWG). The name change reflects the philosophy of the group's recently approved charter—to provide learning opportunities to support the Software Engineering Process Group's efforts to increase the FAA's maturity in acquiring software intensive systems. SEPG in the name refers to the group's sponsor and the intention to narrow the focus specifically to supporting SEPG initiatives. Learning resources in the name emphasizes that learning opportunities include more than just formal training classes. It includes workshops, seminars, mentoring, on-the-job training—in fact, anything that advances the knowledge of the recipient.

Many people have attended training provided by the Training Working Group. The TWG provided courses in cost estimation, the SEI CMMs, project planning and tracking, consulting, Mil-Std-498, SLIM, etc. These courses were open to any FAA member and to eligible support contractors. The SLRG will continue to provide open training. Approval to offer an open course comes from the SEPG. Individuals wishing to nominate a course should bring it to the attention of their SEPG representative.

As The Learning System progresses in its identification of FAA learning gaps, this will also affect the selection of courses.

To better improve the maturity of FAA's processes for acquiring software intensive systems, the SLRG is focusing its efforts on the primary participants—the SEPGs, the process actions teams (PATs), the process improvement facilitators, and the acquisition executives and managers. These groups and individuals will receive training in process improvement, process definition, process measurement, capability maturity models, and the process areas they have targeted for immediate improvement. Although general categories of training have been identified, specialized training is defined by meeting with the various groups to determine their current level of knowledge in an area and to identify specific shortfalls. The knowledge shortfalls or "gaps" are then used as criteria for course selection, tailoring, or development.

SLRG members come from the same organizations as the SEPG: AUA, AND, AIT, ASD, ASU, ACT, AMI, ARS, and AOS. The group is co-chaired by Bill Norton, AIT, and Sean Jenkins, ASU. ■

Training Opportunities...

- **Intermediate Software Acquisition Management 201**
  September 9-26; Ft Belvior, VA
- **SLIM/SLIM Control**
  September 16-17; FAA HQ, Washington, DC
- **Software Development Cost and Schedule Estimation**
  October 20-22; Washington, DC
- **Cost Estimation and Economic Evaluation of Projects**
  January 5-8, 1998; Washington, DC

Contact your SEPG representative for further details.

FAA SOFTWARE ENGINEERING PROCESS GROUP

Art Pyster Chief Scientist for Software Engineering	AIT-5
Linda Ibrahim SEPG Chairperson	AIT-5
Tanae Gilmore SEPG Secretary	SETA
Rebecca Deloney	AOS-1
Tom Marker	ASU-250
Natalie Reed	ACT-24
Ross Ridgeway	AMI-100
Raghu Singh	AIR-200
Cindy King Skiles	AUA-7
Tom Skiles	ATR-300
Rebecca Taylor	ASD-420
George Zerdian	AND-500
Alternates	
Adrian Caster	AOS-5
Rob Hanes	AUA-310
Bob Laws	ASU-250
Louis Pelish	AIT-500
Art Salomon	ASD-130
Herman Tharrington	AND-3

continued from page 1

which uses this library. They will supply all of the "needed" library documentation.

Premature design? Of course—did you think you were just working on requirements? Feel like resigning? You may do so, but that means your method failed at a fairly early hurdle. Let's assume you carry on.

It becomes clear that not all of the requirements can be deduced from the available documents, so you must consult with the "user authorities." They repeatedly change their mind. Each time you go to confirm what you last discussed, they mention something else. Frequently, they completely change what they said the last time and blame you for not understanding. They refuse to sign up to anything and insist on writing the minutes of meetings. The minutes never say quite what you think happened. In fact, such minutes rarely say anything intelligible.

You are told there is a new need to synchronize the positioning of some lifts with the arrival of ground shuttles from metro stations. Sensors at some locations will warn of an impending arrival—you must interface with these. At other locations, another contractor has been hired to supply similar (but not identical) sensors and you must interface to those also.

You are granted an extension and more money, but the contract changes must be approved. Your management refuses to carry on without contractual cover, so the project is suspended for three months until the new contract is agreed. By then two of the team have left (to take up pig farming in Alaska?) and have to be replaced. Naturally, given that you are already over budget, you will make do with the least skilled (and cheapest) people you can persuade yourself can do the job.

They don't have the right skills, so they must be trained—but without extending the project end date. The customer is becoming worried by the escalating cost and slipping schedule and is looking to cancel the project—preferably holding your company responsible for breach of contract.

Next the customer says that the defense department on the fourth floor must restrict access, so some lifts must have a secure ID card reader to permit door opening at that floor. Lifts without card readers must not stop there. Meanwhile, after specifying the interface to the ground shuttle sensors, it turns out that existing sensors don't work the way their documentation said, so you must change the specification and redo a substantial amount of work that has been already completed. One of the sensor suppliers has been contracted to the wrong specification and can't be made to change, so you must cope with both interfaces, neither of which is as originally documented.

The customer informs you that one pair of lifts has broken down, their equipment is obsolete and repair is uneconomical, so they will no longer be used. You must remove them from the system, and since they interface to shuttle sensors you must remove their interfaces. (What!?! No cost savings? But the system is now smaller!)

Now, the finance department on the tenth floor decide **they** need access control. They are very important and mustn't bother with cards, so they have hired another contractor to fit a TV system which their receptionist will use to control access. You must provide the receptionist with a remote control door opening button, and a warning buzzer when the relevant lifts arrive at that floor.

One of your colleagues has a health problem (nervous breakdown?) and becomes unavailable. If the work is not completed on time your company will lose a lot of money, so you and another colleague are asked to share the orphaned work.

Your customer says that the opening ceremony for the new system, to be performed by the Queen as part of the celebrations for her (90th?) birthday has been moved forward several months because of a mix up over whether the ceremony was on her actual or her official birth date. Your management reluctantly agrees.

Next, the customer's display library is found to have bugs and doesn't behave as the documentation says. You must re-do the specification to cater for this. The shortened schedule has already led to problems in performing the design, code, and test. Everyone is looking for someone else to blame.

Implementation is being done by a rival company. They are threatening the customer with legal action over the need to keep their team doing nothing. Your customer, though admitting that the initial fault was not yours, seizes on the point that your method was supposed to lead to easily maintainable requirements. (And you can't even accommodate a few small changes?)

Your management want you to produce the changes very quickly—even if they won a court case their reputation would be damaged—so they place intense pressure on you to cut corners and to look for ways to blame the customer and the implementor. Recall that the customer and implementor organizations are doing the same.

Finally the project is complete. The Queen cuts the tape and a symbolic lift journey is undertaken. The system breaks down on the fourth floor and the Queen can't get out because she hasn't an ID card. Two hours later emergency services breaks through the lift shaft wall at enormous expense—only to find they are in the wrong shaft. However, in the meantime, the Queen has managed to open the lift doors by using her bank credit card—revealing a major security breach in the ID card reader system—and is immediately arrested. The next day the project is vilified in the media. Your method is quoted as the one used in specifying the requirement and most certainly the reason for the fiasco. On television, representatives of the customer and several implementor organizations are seen nodding savagely in agreement in the background.

In conclusion, I emphasize that such complications are not untypical of projects which desperately need a methodical approach. Current techniques fail on these projects. But abbreviated training problems showing the highlights of sound new methods cannot hope to reproduce all the real world effects. We are asking too much from the training problem definer operating within the constraints of a one semester academic or a "quickie" one or two week career enhancement course.

What **does** seem missing from training environments is sufficient indication to students that these extra

continued on page 7

THE LEARNING SYSTEM

by Craig Berry, ABZ-200

The "learning system" is essentially a broad based method for assuring that the Research and Acquisitions Organization (ARA) workforce gets the training and development needed to get the job done. The learning system will encourage individual and organizational learning, and build an environment where work promotes learning, and learning is sought and provided just in advance of the need for the skill. Self-directed learners are essential to the success of the learning system. Individuals have the ultimate responsibility for monitoring and controlling their own learning outcome.

To make these insights and responsibilities a part of the culture takes a lot of reflection and self-analysis on the part of individuals. But as individuals gain and share knowledge with the larger team, an interactive, generative process occurs that facilitates organizational learning. What emerges is a learning organization—a learning system.

In a learning system, most of the responsibility for learning is transferred to the individual. This contrasts with the traditional training model, where the focus is on solutions designed for groups of more or less identical members, where training is designed to meet general needs, with little provision for individual needs or capabilities. In the learning system model, **individuals**, often in cooperation with their fellow **group/team** members, identify their own knowledge gaps within the context of the organization's goals and mission. They work with supervisors and other group leaders to design and obtain the best training to meet their needs at the lowest cost to the organization.

This focus on the individual within the context of ARA's mission needs requires that the widest possible variety of learning options be available. These can include individually designed learning projects, internship opportunities, and traditional courses, seminars, and vendor training. By focusing on the individual, the system attempts to provide sharply tailored learning events and activities which will accommodate individual needs in the most effective and least costly way possible.

The first step in creating a learning system for ARA is the development of competency profiles for the many tasks within ARA. The learning system team has been working with the ARA Intellectual Capital Investment Plan (ICIP) council, composed of the deputy directors, the Directors of the Office of Business Management (ABZ-1), and the chief scientists, to define the various roles associated with the work of ARA. The members of the ICIP council have also identified individuals in each of the roles who will be interviewed to determine the critical competencies needed to perform the role.

Once all the roles have been defined, and their necessary competencies identified and validated, the ARA workforce will participate in a competency assessment process. They will receive individual feedback on weaknesses and strengths relative to the role(s) they are occupying or seeking. This feedback will be used to focus employee development and help ARA determine how best to allocate training investment in the workforce. ■

Letter from the EDITOR

WHERE DO WE GO FROM HERE?

I had hoped to dedicate this entire issue to describing the new FAA environment of Competency Assessment and Learning Systems, but since the Office of Business Management is in process of doing that, I do not want to upstage them.

In the meantime, I have led off with what I consider to be a cautionary (and very funny) tale about why most idealized methods fail in the real world. It is not meant to scare off the serious reformers—just to warn them that they had better pilot any major innovations in the real world before declaring success and going on to other things.

In 1969, in his book, **The Age of Discontinuity**, Peter Drucker made the clear distinction between the "supervisory" style of workplace leadership and the "managerial" style. In the old, production line style of workplace, the supervisor or foreman was also the chief artisan, and directed, instructed, and rated the workers in exactly the job they were expected to perform. As Peter Drucker pointed out, this is a completely inappropriate model for the "information" workplace where each of the workers can be expected to have one or more skills far in excess of the manager, who is simply another information worker with a different set of skills—managerial skills.

The information workplace, therefore, has to be "team" centered with often a technical leader, or "coach," in addition to the manager. The function of the manager is now to act as chief negotiator for the team—securing needed resources (including needed specialists) for it and determining what the proper product should be. The determination of the group product is a result of negotiations with other like groups, upper management, and what that particular group believes is doable and in keeping with the customer's wants and needs. The old model where all direction flowed down and all information flowed up has been replaced by the two-way information flow model. Since everyone is now in on the "picture"—tasks are a result of mutual negotiation and everyone pitches in to accomplish what needs to be done. This is the primary reason that unnecessary tasks are seen as such and eliminated from the workplace—sparking productivity.

Thanks to the most educated work force in history, the "information" workplace model has been embraced by industry, even on the old production lines. In general, it has worked remarkably well, and in the past decade or so, American productivity has skyrocketed. It appears that the FAA is now preparing, gingerly, to adopt this model. But it will be a great shock to many first line managers who must now learn to defer to the people working for them on any but strictly managerial issues. (It will also be a shock to many of the old troops, who must now assume full responsibility for their work product.)

Norm

DUAL TRACKING IN ENGINEERING ORGANIZATIONS

by Norman Simenson, AIT-5

Over something like 40 years, I have seen some very good engineering and some very bad engineering; some very good engineering organizations and management, and some very bad engineering organizations and management. I have some very decided opinions about what separates the Boeings from the companies whose names I cannot reveal for fear of being sued. Indeed, since coming to the FAA in 1991, I even worked with one company that wound up with half its management team going to jail—but that's another story.

A good engineering organization recognizes clearly that it ***is*** an engineering organization—and not something else. This recognition must come from the very top and affect the way the organization is structured and run. Certainly every organization produces something for an **external** customer—although some organizations tend to forget that from time to time. But I think the FAA is quite good in that regard. The need to serve our customer provides a lot of motivation for some very long days and hard work, frequently in the face of strong forces trying to get us to do something other than our job. The Office of Research and Acquisitions and other development and maintenance organizations within the FAA also recognize that they **are** primarily engineering organizations.

Every superior engineering organization I have worked for has used a dual track for management and technical staff. At almost every level of management, there is a corresponding level of purely technical staff which provides the engineering expertise for that level. For example, in a recent visit to the FAA, speaker John Vu reported that Boeing has been a dual track organization for many years. John Vu, as Boeing associate technical fellow, reports directly to a group president of Boeing.

In such organizations, very clear distinctions are made between management and technical areas, and the distinctions are observed. Any manager that acts as his own chief engineer is not considered to be competent in either area. In my own case, when I was a program manager, I made sure that I had very competent staff—certainly a chief engineer that was far more "on top of things" technically than I. If I felt that I could arrive at a better technical solution over a weekend than my chief engineer could during the normal work week, I would have replaced him—ASAP!

For what a successful program manager **is** responsible—beyond budgets and schedules and meetings and resource allocation—see my article in the August, 1996 issue on "Engineering Disasters..." In that article, I went on to say that "Engineering disasters are ***always*** due to bad management and ***never*** to bad engineering." And I listed some of the things program managers must do to assure a successful program. I should have added another **key** principle—a good program manager is ***never*** her own engineer! A good engineering manager need only know enough engineering to understand what her chief engineer is trying to tell her. If she knows more, and can make helpful suggestions, well and good. But they must ***never*** be more than suggestions.

The chief engineer must be free to make the final technical decisions within the cost, schedule, and resource constraints imposed by or, preferably, jointly arrived at with the program manager. On one stint as chief engineer (of a highly successful program, I might add), I did not even report to the program manager! At CALSPAN, Inc., I was designated a "technical area manager," and the TAM and PM were jointly responsible for a program. We both reported—equally—to the next highest level, a junior vice-president who did not like to be bothered by petty squabbles, and let us know it! So, we invariably worked out our own differences.

Higher levels of management should be even more careful to abstain from trying to do engineering. Their engineering role should be strictly limited to choosing among the technical alternatives presented to them by their technical staff. There are several things wrong with upper management trying to do technical "stuff" beyond that. If they have an engineering background and they are pretty good managers, they must now have lost all of their critical engineering skills (except for the very critical skill of being able to talk to and understand engineers). What skills they do retain will invariably be based on obsolete technology and obsolete engineering technique.

Engineering and management are both wholly consuming disciplines with little overlap. No one can master and remain master of both. Each can consume many hours a week beyond just doing the job to remain current. Therefore, to the extent they still try to do technical "stuff," managers must be considered to be rank amateurs. (I remember a manager in 1988 fighting vigorously against going to that "unproved"—read, newfangled—fiber optic technology. This at a time when Pacific Bell was tearing out copper paths as fast as they could, at enormous cost, and replacing them with "unproved" fiber optic paths.)

A manager can insist on proved technology, but cannot legitimately insist on what that should be. The criteria for "proved" should be jointly determined by the manager and engineer; the actual choice of technology should be that of the engineer. If you (as manager) feel that your engineer has gone overboard for some current fad—replace him. The manager should never overrule her engineer on a purely technical decision. Actually, engineers tend to be far less susceptible to "silver bullets" which can apparently solve major problems with little or no expenditure of resources. The manager who thinks he still retains his full engineering credentials is often just exactly the target envisioned by the presentations aimed right at him promising easy engineering solutions. Engineers tend to be far more wary of vaporware, vapor tools, and vapor "techniques," and rarely go after a new technology until it has the approval of the appropriate engineering gurus—or at least an engineering friend.

continued on next page

"Dual Tracking in Engineering Organizations"
continued from previous page

Why do we need a "dual track" organization anyway? Upper management is most comfortable dealing with other management types anyway. There is always a communication problem between engineers and management and it tends to get worse at the higher management levels. But there is much at stake. Middle and upper management in an engineering organization cannot function well if all of the engineering input is filtered through lower level managers. Aside from the fact that the technical skills of lower level management will probably be such that the engineering input is garbled on the way up and upper level management objectives garbled on the way down, engineering and management have very different outlooks on things.

In his report on the Challenger accident investigation, physicist Richard Feynman recalls how he asked a group of engineers and a first level manager what the probability of an engine failure was. Everyone was trying hard to cooperate, but the manager (who insisted he was still a fully competent engineer) found this straightforward technical risk question very difficult to answer. Eventually, he estimated the probability at no greater than 1 in 100,000. The engineers, with little hesitation, had estimated the probability at about 1 in 200. Unfortunately, major decisions are based on such estimates and the decision makers are not served well if the information they get is of poor quality for technical or political or cultural reasons.

At each level of management, top notch technical skills are required to assemble the mosaic of lesser technical endeavors into the fully articulated technical picture at that level. This requires a stable, permanent team of engineers to maintain continuity and to establish mutual trust with the manager(s) at that level. If that is not done, we always risk developing something that is not a system so much as a loose collection of poorly interoperating individual endeavors. There is a tendency for the overall structure to reflect the political environment rather than the technical domain. (This is a major reason for rapidly changing high level requirements—political structure is notoriously unstable.)

Dual track organizations should strongly discourage movement of people between tracks. Such movement defeats the purpose of dual tracking—which is to encourage people to concentrate on one set of skills and build alliances within one community, but not both. ■

"What They Don't Teach You in School"
continued from page 4

problems exist, and any but the most rudimentary instruction on how one may methodically trap, respond to, and contain their damage.

Current methods training is not unlike training people to jump hurdles in the hope it will help them to jump over the moon. It may be a sound idea to know how to jump a hurdle without hurting yourself—but you'll have to think of something much more dramatic if you want to reach the moon.

We will never eliminate the real world complications, at least not in the real world, but in defining methods we must not entirely ignore them—and in method training we should not assume that we can do all that is needed via a small, "simplified" example. When the "simplified" example turns out never to work the same way in the real world, the result is another very cynical engineer who just learns to avoid **all** formal methods. ■

Geoff Mullery operates as an independent consultant on methods, tools and project support. His company is Systemic Methods Ltd., 12 Firs Close Farnborough, Hants GU14 6SR, UK. He has written extensively for the IEEE, the British Computing Society Requirements Engineering Specialist Group Newsletter, and other publications.

THE SOFTWARE PROCESS IMPROVEMENT JOURNEY

... begins with a single step! John Vu of Boeing brought a very simple message: start small, collect lots of data, and let the data speak for itself. No rational manager or engineer will argue with a return on investment of over 7 times, reduction of defects in products delivered to customers of over 83%, and improvement in building to cost and schedule estimates of over 350%. Initially, of course, FAA won't have the data and will have to use data from industry or other sources to justify the investment—and everyone knows the FAA is unique! Every organization feels that way, and so did Boeing—until they got the data on their own results.

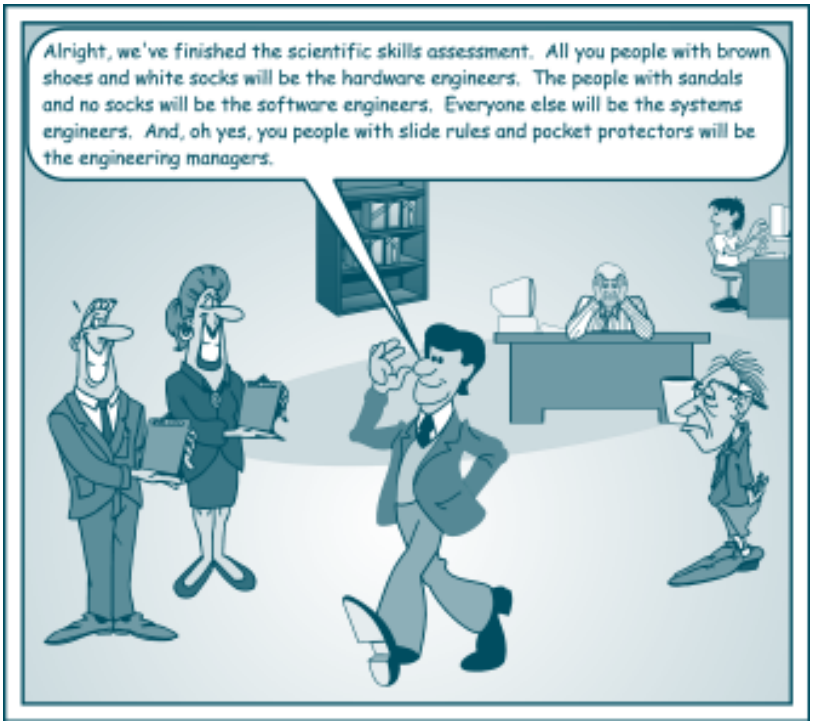
Software process improvement (SPI) doesn't impose new processes—it only insures that you have all of the processes needed to do the job—and then provides you with guidance on how to develop or improve them. Every engineer understands the value of feedback and how it can be used to eliminate noise from a system. That is the essence of SPI—the same approach which can provide you with a clean output from an amplifier can assure you of a clean product from a software development or acquisition. Feedback is provided by continually measuring what you are doing and comparing with what you have been doing and with where you want to be, and using the results to improve a process.

Capability maturity levels are meaningless if they cannot be explained in terms of business objectives which can be measured. The key measures used by Boeing fall into five categories of business objectives: cost, cycle time, quality or defect rate, customer satisfaction, and—soon to be officially added—morale or employee satisfaction. Since 1991, Boeing has seen an increase in the productivity of personnel using SPI of 240% coupled with an employee satisfaction increase of almost 50%—which is reflected in reduced turnover. There has been a 50% reduction in cycle times, an 83% reduction in defects before validation testing, and substantailly higher satisfaction ratings from customers.

Achieving this was not easy. It required total management commitment, total workforce involvement, a SPI budget that was never less than 5%, lots of just-in-time training, measurement and metrics, oversight, a culture of engineering excellence, and customer participation. But fully 80% of the effort went into culture change—not technical or process change. In total, Boeing has been working to implement SPI for at least 10 years.

John Vu is a Boeing Associate Technical Fellow for Software Engineering Research and Technology. ■

In This Issue



interFACE

Newsletter of the Software Engineering Process Group

Volume 6, Number 3
August 1997

SEPG
FEDERAL AVIATION ADMINISTRATION

DOT/FAA/AIT-5
800 Independence Avenue, SW
Washington, DC 20591