

CROSSTALK

April 2003 *The Journal of Defense Software Engineering* Vol. 16 No. 4



THE
PEOPLE
VARIABLE

4 **Destroying Communication and Control in Software Development**
 This article explains how large systems-development projects fail when people destroy communication and control within an organization, either intentionally or not. The author also reveals where and when project managers can intervene with countermeasures.
by Dr. Gerald M. Weinberg

9 **Experiences Applying the People Capability Maturity Model**
 In the seven years since its release, the People Capability Maturity Model has become a road map for implementing practices that continually improve the capabilities of an organization's work force.
by Dr. Bill Curtis, Dr. William E. Hefley, and Sally A. Miller

14 **Obedience Training for Managers**
 If you are making capability maturity model-based process improvements in your organization, these authors draw some helpful parallels between your challenges in changing mindsets and in animal trainers looking to make behavior modifications.
by Virginia Slavin and Paul Kimmerly

18 **People Projects: Psychometric Profiling**
 This article introduces the Compatibility Identification Set as a tool used to help project managers build successful project teams by examining the specific job skills of each potential member.
by Kasey Thompson



Software Engineering Technology

24 **Prospecting for Knowledge**
 This article explains the concept of knowledge management and discusses the challenges in managing tacit knowledge as well as overcoming the critical obstacles to ensure portal success.
by Jüris Kelley

28 **Project Expectations: The Boundaries for Agile Development**
 If you are challenged with managing project scope in an agile development project, this author recommends managing project expectations versus waiting for requirements and plans to stabilize.
by Diana Meikelburg

Departments

- 3** From the Publisher
- 17** Coming Events
- 23** Visit CrossTalk at the STC
- 27** Letter to the Editor
Web Sites
- 30** JOVIAL Services
- 31** BackTalk

CrossTalk

SPONSOR	<i>Lt. Col. Glenn A. Palmer</i>
PUBLISHER	<i>Tracy Stauder</i>
ASSOCIATE PUBLISHER	<i>Elizabeth Starrett</i>
MANAGING EDITOR	<i>Pamela Bowers</i>
ASSOCIATE EDITOR	<i>Chelene Fortier</i>
ARTICLE COORDINATOR	<i>Nicole Kentta</i>
CREATIVE SERVICES COORDINATOR	<i>Janna Kay Jensen</i>
PHONE	(801) 586-0095
FAX	(801) 777-8069
E-MAIL	crosstalk.staff@hill.af.mil
CROSSTALK ONLINE	www.stsc.hill.af.mil/ crosstalk
CRSIP ONLINE	www.crsip.hill.af.mil

Subscriptions: Send correspondence concerning subscriptions and changes of address to the following address. You may e-mail or use the form on p. 22.

Ogden ALC/MASE
 6022 Fir Ave.
 Bldg. 1238
 Hill AFB, UT 84056-5802

Article Submissions: We welcome articles of interest to the defense software community. Articles must be approved by the CROSS TALK editorial board prior to publication. Please follow the Author Guidelines, available at <www.stsc.hill.af.mil/crosstalk/xtlkguid.pdf>. CROSS TALK does not pay for submissions. Articles published in CROSS TALK remain the property of the authors and may be submitted to other publications.

Reprints and Permissions: Requests for reprints must be requested from the author or the copyright holder. Please coordinate your request with CROSS TALK.

Trademarks and Endorsements: This DoD journal is an authorized publication for members of the Department of Defense. Contents of CROSS TALK are not necessarily the official views of, or endorsed by, the government, the Department of Defense, or the Software Technology Support Center. All product names referenced in this issue are trademarks of their companies.

Coming Events: We often list conferences, seminars, symposiums, etc. that are of interest to our readers. There is no fee for this service, but we must receive the information at least 90 days before registration. Send an announcement to the CROSS TALK Editorial Department.

STSC Online Services: www.stsc.hill.af.mil
 Call (801) 777-7026, e-mail: randyschreffels@hill.af.mil

Back Issues Available: The STSC sometimes has extra copies of back issues of CROSS TALK available free of charge.

The Software Technology Support Center was established at Ogden Air Logistics Center (AFMC) by Headquarters U.S. Air Force to help Air Force software organizations identify, evaluate, and adopt technologies to improve the quality of their software products, efficiency in producing them, and their ability to accurately predict the cost and schedule of their delivery.



Engineers at Their Best



During my career, I have worked with many professionals but none more bright, hard working, and fascinating than engineers. Even though I am an engineer, many of my co-workers tell me that I'm not a typical engineer. Sometimes that bothers me. What is a typical engineer, and why are engineers stereotyped anyway?

Maybe it is because on the surface, many engineers are quiet and very studious. They are often confined to a cubicle as they perform their work solving difficult problems. They also spend many hours reading to continually learn the latest technology and trends.

I have worked with enough engineers to know that although they need their time alone, they have much to contribute. They often thrive in projects with team settings, and, due to their mathematical mindset, they absolutely love solving problems. I believe that engineers are at their best when they are looked upon to answer questions or to offer their viewpoints. More times than not, they just want to be asked.

As a manager, I have come to realize that effective communication with engineers and the many other professionals on my staff is the key to successful projects. I also realize that more times than not, almost everyone has his or her own opinion on project matters. Resolving differing opinions might be the toughest part of being a manager, but I believe that knowing the opinions is critical. It lets you listen to the pulse of your project firsthand.

The theme of this month's issue is "The People Variable" in which we focus on the most critical aspect of software development: the people. We all know that developing and sustaining a software-intensive system is difficult and that often the biggest project struggles are centered on the people and not the technology. This month's issue begins with *Destroying Communication and Control in Software Development* by Dr. Gerald M. Weinberg. This article is a clever look at how managing software projects can be like fighting battles. Weinberg stresses the importance of information integrity and protecting systems from disruption due to human error.

Next, Dr. Bill Curtis, Dr. William E. Hefley, and Sally A. Miller bring us *Experiences Applying the People Capability Maturity Model*. Assessment data from 49 organizations during the last seven years indicate that many are realizing the benefits of implementing practices aimed at attracting, developing, organizing, motivating, and retaining employees. We continue looking at the critical people variable with *Obedience Training for Managers* by Virginia Slavin and Paul Kimmerly. This article brings an enlightening parallel between change agents and animal trainers, and more specifically shows how managers can bring more discipline into an organization with a low maturity level.

If you are a program manager, don't miss reading *People Projects: Psychometric Profiling* by the Software Technology Support Center's Kasey Thompson. He introduces a Compatibility Identification Set as a new approach to forming effective project teams. As we continue to focus on the people element of software development, Jüris Kelley reminds us in *Prospecting for Knowledge* that we also need to consider the concept of knowledge management to succeed in today's information-based world. And finally, our issue wraps up with Diana Mekelburg's *Project Expectations: The Boundaries for Agile Development*. This article acknowledges that project scope management can be a formidable challenge with agile software development for traditionally trained managers. The author explains how to overcome this by managing project expectations instead of waiting for requirements and plans to stabilize.

As you strive to be the best you can, I hope this month's issue stimulates you to work and communicate more effectively with the people in your organization and project environments.

Tracy L. Stauder
Publisher



Destroying Communication and Control in Software Development[©]

Dr. Gerald M. Weinberg
Weinberg & Weinberg

More than half of large systems-development projects fail, either intentionally or not, because people destroy communication and control within the organization. This article explains how this occurs when using basic tools, including requirements, configuration management, testing, and more. Fortunately, as the author reveals the methods used to sabotage these tools, he also advises project managers on where and when to intervene with countermeasures.

More than half of large systems-development projects fail, and of those that succeed, very few are delivered on schedule [1]. The situation is so bad that nontechnical executives often ask, "How does a project get to be one year late?" Years ago, Fred Brooks gave the astute answer, "One day at a time" [2].

Astute as it was, Brook's answer was not too helpful for those trying to teach nontechnical executives how to avoid these hyper-extended projects – or at least to see them coming. What exactly *happens* on those days where the project falls behind? Watching the progress of the war in Afghanistan, I finally realized how to explain this dynamic so everyone could understand.

In modern warfare, the first step in hostilities is to destroy the enemy's communication and control system, after which they can be easily defeated because of their confusion and inability to coordinate their forces. This is exactly the strategy that seems to be taking place when the development organization destroys its management's ability to control the organization. It may or may not be intentional. It may not happen all at once. But one day at a time, one small step at a time, the net effect is the same as a carefully planned and executed war – destroying the manager's ability to manage successful projects.

The *war* in this case is a war against *nature*, including human nature. A software product is, in essence, composed of millions, or tens or hundreds of millions of tiny *parts*. Each of those parts must be built correctly, but that is not enough. Building the product consists of putting those correctly built parts together in the proper sequence. If we consider each part as analogous to a military target, the desired product is a large set of targets. When all the targets have been hit, in the proper sequence, the war is won and the product is built successfully.

But by nature, human beings are not

well equipped for such precision work. Errors (missed targets) occur in every product development. Unless a project is well managed, the war is never won, and the product is never finished or is finished poorly.

Why would a project be managed badly? It is an application of the *first law of bad management*: "If what you're doing isn't working, do more of it" [3]. I used to

*"As their
communication
and control system is
destroyed, managers
simply do not know that
what they are doing is
not working."*

wonder how managers could be so stupid as to continue doing things that were not working. After watching many software development wars, I realized that it is not a matter of stupidity. As their communication and control system is destroyed, managers simply *do not know* that what they are doing is not working. They do not know that essential targets are being missed.

Suppose you were a military general directing air strikes by using video cameras to show the target area. Suppose those cameras were not actually showing the target area, but recorded scenes from some other area. You would continue bombing, thinking you were hitting targets until the enemy mustered a surprise attack with forces you thought had been eliminated.

Managers whose communication and control systems have been corrupted usually do not know anything is wrong until the delivery date arrives. The delivered code is not complete, and what is complete is full of errors that have not been

eliminated. If the communication and control system has not been destroyed but instead *rigged* to fool them, they may not even know that they do not really have the product they expected to have. In the following sections, I will cover five general categories of communication and control disruption:

- Destroying information.
- Destroying information infrastructure.
- Hiding information.
- Degrading the believability of information.
- Inserting misleading information.

Let me emphasize that these disruptions do not have to be intentional, although they might be. People with the very best intentions, even the managers themselves, may do each of them. That is because people are not perfect data recorders. They are influenced by stress, by pressure, by what they think will happen to themselves or other people, and by just plain mistakes. Only in the smallest of projects can a manager rely solely on personal reports by individuals, no matter how well intentioned they may be.

Instead, as projects grow larger, managers must create and guard communication and control systems that protect against disruption by individual error; these systems must in turn be protected from disruption. I will illustrate such disruptions by examining typical mistakes when using the following principal tools that managers require for communication and control:

- Requirements will be used to illustrate destroying information.
- Configuration management will be used to illustrate destroying of information infrastructure.
- Technical reviews, project management reviews, and quality assurance will be used to illustrate hiding information.
- Testing will be used to illustrate degrading the believability of information.
- Demonstrations and risk management

processes will be used to illustrate inserting misleading information.

Of course, the mistakes are not limited to the following examples, and each type of mistake can be found in each communication and control tool.

Destroying Information

In software systems, the product is invisible unless special efforts are made to render it visible. To take a simple example, an individual programmer might make 50 test-runs but never record that fact, or tell anybody. He might find a dozen faults with his code that he does not fix and never mentions. Pair programming (as in eXtreme programming) is one way to prevent this kind of information destruction, but not if both members of the pair collaborate in the destruction.

In project reviews, we frequently see another common form of information destruction: the shading of figures. To take a common example, a piece of work on the critical path is a week behind; however, the project manager makes a *slight adjustment* so that this delay will not raise an issue with upper management. After all, he or she reasons, everyone will just work a little harder and catch up. Maybe they will, and maybe they will not, but now the manager is steering by a distorted *video* rather than by a view of the actual battlefield.

Requirements

Because software is an invisible product, The Zero Law of software engineering states, "If you don't have to meet quality requirements, you can meet any other objective" [4]. Thus, whenever there are not clear requirements on quality, anybody can say, "We're doing just fine." It is like dropping bombs without a map of target locations – all you can say is that you have dropped a lot of bombs, not what effect it is having on the enemy. So, without defined requirements, so-called progress reports are not reports of progress at all but merely reports of effort expended – the number of bombs dropped in the dark, and the number of days working with unknown results.

Without requirements, when the software finally becomes executable and something bad happens the developer can say, "That is not a bug, it is a feature." ("That hospital we destroyed was a secret enemy base.") By preventing requirements from being explicit, developers can thus ruin any real information about quality.

But how is this done when everybody knows the importance of requirements? Early in a project when a manager tries to obtain explicit requirements, he or she will

hear dozens of excuses, some of the most common of which are these:

- "We know what is needed, so writing it down will be a waste of time."
- "It is too hard to get everyone to agree. It just creates conflict."
- "You cannot really know the requirements until you let the customer see it, so we should not bother them."
- "It will take a long time, and we have to start the real work of coding, otherwise we will not meet the schedule."

Each such argument has some appeal for the manager who does not want to waste time, create conflict, bother customers, or miss the schedule. However, in accepting such arguments, the manager self-destructs his or her ability to communicate and control. In doing so, he or she ensures that time will be wasted, conflict will be rife, customers will be bothered, and the project will not meet its schedule – exactly the opposite of the intended effects.

"By preventing requirements from being explicit, developers can thus ruin any real information about quality."

So why do managers keep falling into this self-destructive trap? I believe it is because there is some truth to each argument: Requirements work, when done badly, can waste time, engender conflict, irritate customers, and delay a project. The solution, though, is not to eliminate requirements work and rush into coding, but to create and support an effective requirements process. There are a number of ways to do this, and the manager must not fall into the trap of indecision about which one to use.

Choose a process for managing both initial requirements and changes to requirements. Train people or hire experienced people to execute your process. Above all, see that the process is actually carried out. Only then can you have fact-based communication and control. Otherwise, you will be like an artillery commander whose gunners report they are never missing the intended target, which is technically true because nobody specified a target.

Destroying Information Infrastructure

High-level managers can generally evaluate the success of a requirements system because they should be able to understand the targets – those features and attributes that their customers desire, at least the nontechnical ones. But high-level managers are generally not qualified to examine technical details and determine their correctness. Even if qualified, certainly they have no time for the job. Instead, they must rely on indirect information about quality beneath the requirements level.

The principal management tools for obtaining such information in an understandable form are personal reports from the technical staff plus reports from quality assurance, technical reviews, and testing. These are the radar detectors, reconnaissance satellites, cryptographers, and field reports in the war against error. Anything that deranges or distorts personal reports, quality assurance, technical reviews, or testing will destroy essential communication and control information.

Underlying all these reporting systems is the configuration management system (CMS) that has the job of retaining all essential project information: requirements, design, code, test plans, test data, test results, review reports, project management information, architectural data, and user documentation.

The CMS is designed to prevent physical destruction of information such as altering of reports; unrecorded changes to requirements, code, or tracking data; unauthorized entries in data fields; or physical failure of media. Without a functioning CMS, the manager cannot rely on the accuracy of any information. However, the CMS is easily undermined in numerous ways that illustrate the destruction of the information infrastructure.

The CMS

Perhaps the most common way to destroy a CMS is by passive behavior. People do not object directly to the system, but they fail to use it or fail to use it correctly in ways that might be attributed to innocent misunderstandings.

For instance, items that are supposed to be placed in the CMS are not found there. When someone asks the responsible people, they might say, "Oh, we did not know you needed that in draft form. We still have some unresolved issues, so we thought we would wait until it was perfect before we put it in." It is difficult for a manager to fault people who merely say

they are trying to do a good job.

To take another example, consider the bug-tracking database, which is part of the CMS that is supposed to report each error found in testing. Each error report remains *open* until the error is tracked down and repaired. Successful managers rely on statistics from the bug-tracking database to monitor project progress and decide about product release. Statistics are varied, but include the types of errors detected, the rate of finding and removing errors, and detection of error-prone parts of the product.

Such statistical information, however, loses its usefulness if errors and their handling are reported accurately, but passive corruption of this database takes place such as these many forms:

- Developers remove error reports claiming that they are not *really* errors, giving specious reasons like “that particular build wasn’t done correctly,” which is another interesting fact in and of itself.
- Managers remove error reports claiming, “We are not supposed to be testing that yet,” which gives their managers an overly optimistic sense of progress.
- Testers file incomplete error reports, omitting such valuable information as the original cause of an error, which would help management detect those areas that need additional support or training.

The CMS obtains its value by making information available to all who might need it, while protecting information from corruption by those who have no authority to change it. Because of its protection function, the CMS must have the ability to restrict access. When that restriction is applied to *reading* the information, however, the whole purpose of the CMS is undermined.

In complex projects, you never know who needs to know what. For a development organization to be successful, information must flow freely. But managers may become territorial and say that certain data “is relevant to our group only.” They may request that the CMS restrict access to these data, and upper management may mistakenly support them.

Why would upper management make such a grave mistake? Perhaps they fear that morale would drop if people knew the true state of the group’s work. Perhaps they are trying to protect the group manager from blame. Maintaining morale and keeping a blameless atmosphere are laudable goals, but if these goals can be reached only by dismantling

the information infrastructure, the project is a lost cause.

How do you protect your CMS? First of all, understand that your CMS is not just some technician’s tool, but a management tool that underlies all communication and control. It belongs to you so manage it, which means the CMS group should report to upper management, not to project management. Second, set and enforce a policy of complete and open information at all times and resist plausible sounding arguments for hiding information that is in the CMS. Third, manage your people well without blaming, because blaming leads to the desire to hide information from management [5].

Hiding Information

Projects certainly generate swarms of data, and sometimes managers argue for hiding that much data to protect workers

“... set and enforce a policy of complete and open information at all times and resist plausible sounding arguments for hiding information that is in the CMS.”

from excess complexity. These managers restrict people’s access to the CMS *for their own good*, but the side effects are ruinous. More astute managers control complexity by creating special functions to manage the data, extracting useful information in condensed form.

Typical of such functions are technical reviews, project management reviews, and quality assurance. Because their job is to extract relevant information from mountains of data, anything that hides data from these functions also hides information from management.

Technical Reviews

Technical reviews extract relevant data by transforming technical detail into a non-technical answer to the question: “Does this work product do what it is supposed to do?” Moreover, if the answer is *no*, the technical review provides information about what else needs to be done, and what issues need to be resolved.

In other words, technical reviews are a form of testing, with three major advantages:

- They can be applied to any work product, not just code.
- They can be used much earlier in a project to save dead ends.
- They can find types of errors that may not be found by testing.

Without technical reviews, managers are at the mercy of individual reports from their technical staff. Unfortunately, this is an ineffective source, for software developers are notoriously unable to give accurate assessments of their own work. By the time (during testing) a manager discovers that a developer was overly optimistic, most of the damage has been done and the costs and time are not recoverable.

Many developers (and testers and documenters) would rather not have management know the true status of their work. They figure that “if I just have a little more time, and nobody bothers me, I’ll get it all right.” And, sometimes, they are correct. Unfortunately, a large project that relies on such self-assessments will *always* fail because some of these predictions will invariably be wrong, and the managers will not know which ones.

Project managers can be victims of the same optimism, but technical reviews will soon reveal the true state of their project to their own managers, unless they can somehow conceal the results. They may try to convince their managers that “technical reviews are not really needed for this particular part.” They may argue that the product is too complex, or that it is too simple. They may argue that reviews would slow things down, or that their programmers are very good so nothing important could go wrong. Or perhaps they argue that the programmers would be upset to have their work reviewed (and of course we must not upset our developers, even if it costs us our project).

If project managers fail to convince their managers not to hold technical reviews of a product candidate, they may hold “reviews” that do not follow an effective discipline. Or, they may hold effective reviews, then fail to follow through on addressing the issues raised. When the review concerns a requirements document, the project managers may order the developers to make the revisions but continue designing and coding from the old documents. Then the code and the requirements become so misaligned that the requirements cannot be used for designing test cases against the code. When bogus *reviews* are done for appearances, not for impacting the quality of the product, they are, as claimed, a waste of time and destroy morale. They also prejudice the organiza-

tion against future technical reviews.

You can spot and prevent bogus reviews principally through the institution of a corps of professional review leaders trained and experienced as facilitators of the human processes involved [6]. As your organization stabilizes, you can monitor reviews with appropriate measurements that will make bogus reviews stand out from real ones [7].

Project Management Reviews

It is quite natural for complex projects to drift off course. In order to steer them back on course, managers conduct regular project management reviews. These reviews transform masses of data about project status into information that can be used by upper management to assess the true state of projects and the rate of progress. Without accurate information high-level managers cannot design actions that will bring projects back on target.

Project managers, however, often view these project reviews as unnecessary interference with their authority. Privately they will say, "I just want to do my project, without interference from above." To them, well-run project management reviews threaten to expose their imperfections, unless they can somehow manipulate the reviews to hide information, rather than reveal it. They may carefully script the reviews and rehearse them so that none of the really important information leaks through to their managers.

These misguided project managers will hide information behind a slick presentation laden with lots of irrelevant data expressed in *techno babble*. If forced to discuss risks, they will do whatever is necessary to pooh-pooh them – a position that is easier to support when upper management responds to honest risk reporting by emphasizing that risks are *not* acceptable.

To avoid this trap, watch out for reviews that are run too smoothly. Insist that your reviews use only documents and files used in actual day-to-day work, not those specially prepared for the meetings. Between meetings, spot check to see if what you are seeing are actual work products. Above all, monitor and compare predicted and actual accomplishments, where accomplishments are strictly tested/reviewed work products and not abstractions such as *45 percent complete*, and are not chunks too huge to see work products from one review to the next.

Quality Assurance

The quality assurance function transforms data into useful management information by helping to establish processes and stan-

dards that, if followed, will assure quality, and then by assuring that these processes and standards are actually being followed. By observing what people are actually doing, quality assurance can provide early warning of likely missed targets; even before product candidates are available for review or testing.

Obviously, quality assurance cannot do its job if it cannot observe what people are actually doing. One of the principal ways of hiding information is to exclude quality assurance people from various working meetings. And, if those meetings produce minutes, quality assurance people are excluded from the distribution list.

Prevent these abuses by having quality assurance report to the highest levels of management, and not to project management. Insist that minutes of all meetings go in the CMS so you can check that such minutes are available to quality assurance

"The tester's job is to reveal missed targets; management's job is to protect them from being abused for doing their job ... train them [testers] in accurate, nonjudgmental communication, so you can trust what they tell you."

for every meeting. And, if people start asking you to exclude quality assurance from meetings, that is the time to dig under the rock to see what is hiding.

Degrading the Believability of Information

Those managers and developers who want to hide information on the true state of their projects see quality assurance people as an impediment. Because the quality assurers' job definition requires that they be allowed to observe *anything*, simply excluding them from meetings and minutes may prove difficult. A more *effective* tactic may be to discredit the quality assurers by saying they are disruptive, do not know enough to understand what is being

done, are not team players, or are too negative. That way, reports from quality assurance can be ignored, and eventually the assurers can be excluded altogether from meetings and access to information. If the assurers then try to object to any practice, they can be further discredited by saying, "How would they know? They have not participated in anything, and they have not seen the real data."

Of course, this style of defamation can be used on anyone who speaks up: a tester, an architect, a consultant, or a manager from another area. "What could they possibly know that we, the builders, do not already know?" Once upper management believes this falsehood, any report or recommendation from such an *outside* source can be safely ignored until it simply disappears. Nobody will know the true state of the project until it is far too late to put it back on track.

Do not be put off by arguments that the quality assurance people are disruptive; if necessary, simply instruct them to observe and report, and not try to say anything in the meetings. Accusations such as this will tend to disappear if you provide skilled professional facilitators for troublesome meetings. At the very least, you will be able to believe their version of what is actually happening in meetings you cannot attend.

Testing

Testing is the best place to illustrate the degradation of believability, because testing gives the most solid information about how bad a product really is. If developers cannot discredit test results, then all their mistakes are exposed to management view.

The most fundamental tactic for discrediting testers and the information they provide is to blame them for carrying the following messages:

- "Testers are always negative; don't they have something positive to say?"
- "If they were team players, they wouldn't focus so much on what's wrong, but would help make things look good."
- "Why don't they try to understand why those are not really errors? They're not developers, so they should listen to us."

Managers who fall for these ridiculous arguments succeed in cutting off their most reliable (albeit late) source of real information about hitting targets. The tester's job is to reveal missed targets; management's job is to protect them from being abused for doing their job. So, protect your testers; definitely do not consider them some lower form of

employee, like *developer's little helpers*. If necessary, train them in accurate, non-judgmental communication, so you can trust what they tell you.

Inserting Misleading Information

In war, enemies often conduct *misinformation* campaigns, inserting incorrect information in order to give a false impression of the true battle situation. In war, this is done intentionally, but in software projects, the misinformation does not have to be intentional to destroy a project.

For example, when issues are raised in testing or technical reviewing, reports to management may show that these issues have been assigned to individuals or groups who are never actually given the assignments. Or, when issues are classified according to severity, classifiers may be pressured to downgrade each issue "so we do not alarm management."

Demonstrations

The classical case of misleading information is the demonstration. Demonstrations are not really part of the information infrastructure because managers have known for a long time that they are almost always rigged to make a product look much better than it actually is [8]. This may be great for sales, but woe to any manager who believes a demonstration instead of data from reviews and testing.

Demonstrations may be rigged in numerous ways, including the following:

- Developers may add to the test system to support a demonstration, and then remove items because they were not really ready for testing, let alone actual use.
- A developer may run the demonstration to carefully avoid any feature or combination of features and data that do not work properly.
- Developers may emphasize showy features for their *gee-whizz* effect, ignoring essential features that are mundane, but difficult to implement.
- Developers may avoid stressing the demonstration by bypassing attributes such as security, performance, and error-recovery. Instead, they show a few *normal* activities done under the easiest of conditions.

Never be fooled by a demonstration because they are *not* product demonstrations but only a sales technique. If you want a *real demonstration*, take the product out of the hands of development and put it in the hands of an acceptance test team.

Risk Management Process

Testing, though a solid source of data on missed targets, often comes too late in a project to permit effective management action. A risk-management system attempts to identify potential trouble spots early while managers still have a chance to thwart them. However, because it is assessed early in the project, risk information lacks the solid foundation of testing and is all too easily used to mislead rather than to lead.

Any software project is replete with risks; the handling of risks is the true test of management mettle. The best managers, like the best generals, want to enter their battles with risks fully laid out in front of them and their troops. The best armies have the courage to face risks head on, as do the best development staffs. But if generals lack confidence in their troops' courage, they will be tempted to mislead their troops about the existence and seriousness of risks.

This kind of distortion places the troops in a position of always being surprised when risks are realized and ill prepared to deal with them. It also causes troops to lose faith in the wisdom of their leaders, and to believe that their leaders think poorly of them.

With an open, explicit risk management process, a manager can minimize risk distortion, optimize the handling of risks that do occur, and bolster the confidence of their staff.

Conclusions

Managing software projects, like fighting battles, is a challenging business, but it becomes well nigh impossible without high-quality information. Consequently, the first job of a successful software manager is to ensure the quality of the information needed for communication and control – to protect this information from error, loss, and distortion regardless of the source. Most of all, the general has to protect troops from blame for communicating information, regardless of how unwanted that information might be. ♦

References

1. Jones, C. Software Systems Failure and Success. Boston, MA: International Thomson Computer Press, 1996: 4-5.
2. Brooks, Fred P. The Mythical Man-Month. Reading, MA: Addison-Wesley, 1982: 153.
3. Weinberg, G. M. Quality Software Management: Volume 1 Systems Thinking. New York: Dorset House,

1991: 62.

4. Weinberg, G. M. Quality Software Management: Volume 2 First-Order Measurement. New York: Dorset House, 1992: 295.
5. McLendon, J., and G. M. Weinberg. "The Blaming Organization." IEEE Software 1998.
6. Freedman, D. P., and G. M. Weinberg. Handbook of Walk-Throughs, Inspections, and Technical Reviews. 3rd ed. New York: Dorset House Publishing, 1990.
7. Humphrey, Watts. S. Managing the Software Process. Reading, MA: Addison-Wesley, 1989.
8. Weinberg, G. M. "How to Automate Demonstrations." Datamation Vol. 8 (1962): 40-42.

About the Author



Gerald M. Weinberg, Ph.D., is a principal in the consulting and training firm Weinberg & Weinberg. For more than 45 years, he has worked on transforming software organizations. Weinberg is author and co-author of more than 40 books, including "The Psychology of Computer Programming," and "An Introduction to General Systems Thinking." His books cover all phases of the software life cycle, including "Exploring Requirements," "Rethinking Systems Analysis and Design," "The Handbook of Walkthroughs," "Inspections and Technical Reviews," "General Principles of System Design," and "The Roundtable on Project Management." His books on leadership include "Becoming a Technical Leader," "The Secrets of Consulting," "More Secrets of Consulting: The Consultant's Tool Kit," "The Roundtable on Technical Leadership," and "Quality Software Management," a four-volume series. Weinberg is also known for his conferences for software leaders, including the "Amplifying Your Effectiveness Conference."

Weinberg & Weinberg
10131 Coors Road N.W.
Suite I-2
Albuquerque, NM 87114
Phone: (505) 897-9707
E-mail: hardpretzel@earthlink.net

Experiences Applying the People Capability Maturity Model

Dr. Bill Curtis
TeraQuest

Dr. William E. Hefley
Carnegie Mellon University

Sally A. Miller
Software Engineering Institute

This article introduces the People Capability Maturity Model® (People CMM®), describes key contributions of the People CMM®, and provides a summary of benefits and lessons learned from its use. The People CMM was first published in 1995 [1]. Anticipating the emergence of human capital, information technology work force, and work force aging issues [2, 3], senior leaders in the Army's Chief Information Office and Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence sponsored development of the People CMM. In the seven years since its first release, the People CMM has successfully guided work force improvement programs in many organizations such as The Boeing Company, Lockheed Martin Corporation, Computer Sciences Corporation, Intel Corporation, Novo Nordisk A/S, Tata Consultancy Services, Infosys Technologies Ltd., Wipro Technologies, the U.S. Army, and the Federal Emergency Management Agency [4, 5, 6].

The People Capability Maturity Model® (People CMM®) is a road map for implementing work-force practices that continually improve the capability of an organization's work force. The People CMM¹ is a process-based model that assumes work-force practices are organizational processes that can be continuously improved through the same methods used to improve other business processes.

In particular, the People CMM assumes that work-force practices can be improved through the staged process transformations that underpin Humphrey's Process Maturity Framework [7]. The People CMM applies the Process Maturity Framework to develop the work-force capability of an organization. Each successive level of the People CMM produces a unique transformation of the organization's culture by equipping it with more powerful practices for attracting, developing, organizing, motivating, and retaining its work force.

The People CMM establishes an integrated system of work-force practices that mature through increasing alignment with the organization's business objectives, performance, and changing needs. Although the People CMM was designed primarily for application in knowledge-intense organizations, it can be applied in almost any organizational setting with appropriate tailoring.

The practices at Level 3 of any well-formed capability maturity model produce an architecture for a critical aspect of an organization's strategic infrastructure. For instance, Level 3 practices in the Capability Maturity Model® for Software (SW-CMM®) and CMM IntegrationSM (CMMI®) produce the architectures of standardized processes that support an organization's

software and systems business.

Likewise, the People CMM produces the architecture of work-force competencies an organization requires for executing its business. Achieving Level 3 of the People CMM and either SW-CMM or CMMI will enable an organization to have a standardized architecture for its development processes and a strategically designed work force strong in the domain specialties required to perform them.

The People CMM was designed to achieve four objectives in developing an organization's work force: develop individual capability, build work groups and culture, motivate and manage performance, and shape the work force. Figure 1 depicts how the process areas at each maturity level are organized to support the four primary objectives (represented in the columns) of the People CMM.

Although the People CMM can be represented in the appearance of a continu-

ous model, failure to implement a cohesive system (or bundle) of integrated practices at each level can have harmful consequences. One example of these consequences is often seen in organizations that encourage people to work as teams, while still rewarding them as individuals. Thus, practices in the People CMM should be implemented using a staged, rather than continuous strategy.

Guidance for Improving Work-Force Capability

The Process Maturity Framework was designed to apply to practices that contribute directly to the business performance of an organization, that is, to the organization's capability for providing high-quality products and services. Since the capability of an organization's work force is critical to its performance, the practices for managing and developing them are excellent candidates for improvement using the Process

Figure 1: Objectives Pursued Across Levels in the People CMM

People CMM Objectives and Their Supporting Process Areas				
Levels	Developing Competency	Building Workgroups and Culture	Navigating and Managing Performance	Shaping the Workforce
5 Optimizing	Continuous Capability Improvement		Organizational Performance Alignment	Continuous Workforce Innovation
4 Predictable	Competency Based Assets Mentoring	Competency Integration Empowered Workgroups	Quantitative Performance Management	Organizational Capability Management
3 Defined	Competency Development Competency Analysis	Workgroup Development Participatory Culture	Competency Based Practices Career Development	Workforce Planning
2 Managed	Training and Development	Communication and Coordination	Compensation Performance Management Work Environment	Staffing

[®] Capability Maturity Model, CMM, and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

SM CMM Integration, SEI, and SEPG are service marks of Carnegie Mellon University.

	Level 1	Level 2	Level 3	Level 4	Level 5
1996					
1997					
1998				← Lead Assessor or Program Initiated	
1999					
2000					
2001					
2002					

= 1 Organizational Assessment Reported by an SEI-Authorized People CMM Lead Assessor
 Figure 2: People CMM Maturity Profile of 49 Organizations Assessed Through November 2002

Maturity Framework. Thus, the People CMM has been designed to increase the capability of the work force just as the SW-CMM is designed to increase the capability of the organization's software development processes.

The People CMM's primary goal is to guide organizations in improving the capability of the work force. Work-force capability can be defined as the level of knowledge, skills, and process abilities available for performing an organization's business activities. Work-force capability indicates an organization's readiness for performing its critical business activities, its likely results from performing these business activities, and its potential for benefiting from investments in process improvement or advanced technology.

The following paragraphs describe how the People CMM supports growth in work-force capability as the organization matures. At the Initial Maturity Level (Level 1), work-force practices are performed inconsistently or ritualistically and frequently fail to achieve their intended purpose. Managers usually rely on their intuition for managing their people and may not receive guidance on practices unless they are legally mandated.

To achieve the Managed Maturity Level (Level 2), managers begin performing basic people management practices such as staffing, managing performance, and making adjustments to compensation as a repeatable management discipline. The organization establishes a culture focused at the unit level for ensuring that people have

the skills and resources needed to meet their work commitments. The fundamental objective of all capability maturity models at Level 2 is to stabilize the local work environment, whether it is a project or some other form of work unit.

By applying the concept of committed work at Level 2, both staffing and performance management activities are integrated into a framework that balances workload and objectives with the resources available for performing the work. These practices control commitments in the same way achieved in other capability maturity models through project planning. Managers ensure that people have the skills needed to perform their work, that they have the information and coordination skills needed to work effectively with others, and that the work environment provides the needed resources and minimizes distractions. At Level 2, units are able to manage the skills and performance needed to accomplish their committed work.

To achieve the Defined Maturity Level (Level 3), the organization identifies and develops the knowledge, skills, and process abilities that constitute the work-force competencies required to perform its business activities. The organization develops a culture of professionalism based on well-understood work-force competencies. A work-force competency is a cluster of knowledge (what must be known to perform skills), skills (what must be done to accomplish work tasks), and process abilities (how skills are to be performed using the organization's standardized processes).

An organization's strategic work-force competencies might include software engineering, systems engineering, manufacturing, and field service among others. It is the process abilities within a work-force competency that enable the organization to integrate its architecture of competencies with its standardized process architectures. These process abilities also provide a formal structure for developing work groups through roles and standard processes that can be tailored. In achieving Level 3, the organization develops the capability to manage its work force as a strategic asset.

To achieve the Predictable Maturity Level (Level 4), the organization quantifies and manages the capability of its work force and their competency-based processes, in addition to exploiting the opportunities afforded by defined work-force competencies. Level 4 of the Process Maturity Framework has traditionally been limited to quantitative management of the organization's standard processes. Results and observations of high maturity organizations during the past decade indicated that they were implementing more than just quantitative management. Level 4 software organizations were implementing a range of practices such as software reuse and structured mentoring that were enabled by having a defined Level 3 process, and that had the effect of reducing variation through means other than quantitative management.

The People CMM incorporates process areas at Level 4 that extend beyond the traditional quantitative management focus, but remain within the philosophy of reducing variation and performing predictably. The organization creates a culture of measurement and exploits shared experience. At Level 4, the organization has the capability to predict its performance and capacity for work.

To achieve the Optimizing Maturity Level (Level 5), everyone in the organization is focused on continuously improving their capability and the organization's work-force practices. The organization creates a culture of product and service excellence. At Level 5, the organization continuously improves its capability and deploys rapid changes for managing its work force.

Where Has the People CMM Been Adopted?

Early adoption of the People CMM has occurred primarily in organizations that have already adopted the SW-CMM. Not surprisingly, among the earliest adopters were aerospace companies such as The Boeing Company, Lockheed Martin Corporation, and GDE Systems (now BAE

Systems). Government agencies such as the Federal Emergency Management Agency are adopting the People CMM to address the government's objective of raising the performance and capability of the federal work force. The strongest adoption has occurred in many Indian software companies. The maturity profile of reported People CMM assessments during the last seven years is displayed in Figure 2.

Although many companies were using the People CMM to reduce the high employee turnover rates endemic during the late 1990s, the three main reasons for adoption in the Indian software industry were more complex. First, India's interest is a natural outgrowth of their belief that their highly skilled work force is their greatest natural asset. As Narayana Murthy, chairman of Infosys Technologies Ltd., said, "Every night all my assets walk out the gate."

Thus, the People CMM provides Indian software companies with a road map for investing in their most valuable asset. Based on software companies' success using the People CMM, the Confederation of Indian Industries is now engaged in a vigorous campaign to extend the adoption of the People CMM to all industries in India.

Second, the People CMM allows Indian software companies, especially those in the outsourcing business, to address one of their customers' most important concerns. The outsourcing business has been plagued by deals that transferred all of one company's software people to another company, only to see decades of application knowledge disappear as many of these people leave the outsourcer within a few years. Even if no developers are transferred to an outsourcer, which is often the case with outsourcing arrangements between U.S. firms and India-based service providers, the clients consider their business with the outsourcer to be an investment in the outsourcer's employees who are learning the client's applications.

Thus many Indian companies are using the People CMM to demonstrate that they have implemented work-force practices that maximize their ability to retain the staff serving their clients. Since the client sees the outsourcer's staff as a critical resource in which they have invested heavily, the People CMM provides an assurance that their investment in application knowledge will be retained. Otherwise, the client may pay for the development of the outsourcer's application knowledge many times over.

Third, the People CMM has been used as a means for sustaining the capability achieved in a high-maturity environment.

Company	Initial Turnover	Level 2 Turnover
Boeing BRS	1998	1999
	7%	5%
Novo Nordisk	1996	2000
	12%	8%
GDE Systems	1996	1998
	7.8%	7.1%

Table 1: *Annualized Voluntary Turnover*

By the late 1990s, excessive turnover among many Indian software companies was threatening their ability to sustain the performance and capability of their high-maturity practices and their achieved capabilities. The People CMM not only addressed turnover, but also implemented a system of practices that builds a work force capable of achieving the performance levels that most benefit from quantitative management. These practices supplement and are complementary with those of other CMMs [8].

Not surprisingly, the recent People CMM assessments reporting attainment of Level 4 and Level 5 capabilities all emerged from India. The implementation of structured mentoring, reusable assets and experiences, empowered work groups, and quantitative analysis of the effect of work-force practices on process performance reinforced and supported the practices implemented through SW-CMM and CMMI. Comments from students in the "Introduction to the People CMM" course indicate that they better understand and appreciate the intent of SW-CMM and CMMI at higher maturity levels when they understand how high maturity work-force practices contribute to the organization's capability.

What Benefits Have Been Achieved?

The benefits of implementing the People CMM differ by the maturity level attained. Organizations achieving the People CMM Level 2 uniformly report increases in work-force morale and reductions in voluntary turnover. Table 1 presents a sample of the voluntary turnover reductions for companies that reported achieving Level 2. These results are not surprising since years of research have shown that one of the best predictors of voluntary turnover is employ-

ees' relationship with their supervisors. The primary change at Level 2 is to get unit managers to develop repeatable practices for managing the people who report to them and to ensure the skill needs of their units are met.

Organizations that achieve Level 3 experience productivity gains associated with developing the work-force competencies required to conduct their business activities. For instance, Figure 3 (see page 12) compares the level of competency among the members of a software development project at Infosys (shown as the overall competency index) with the project's cost of quality (rework). Infosys reports a significant correlation of 0.45 ($p < 0.05$) between these variables, indicating that 21 percent of the variation in the cost of quality can be accounted for by the collective competency of the team. That is, the more competent that the members of a development team are in the knowledge and skills related to the technology and application on a project, the less rework the project will experience.

These results are consistent with results obtained by Boehm and his colleagues in calibrating the productivity factors in COCOMO [9, 10]. These data are an example of the quantitative analyses of work-force capability implemented at Level 4 from an Infosys site that has recently reported attaining People CMM Level 5. Infosys was recently assessed at the People CMM Level 5 and uses data such as these for evaluating the effectiveness of its work-force management practices.

At Level 4, an organization begins to achieve what Deming [11] referred to as *profound knowledge* about the impact of its work-force practices on its work-force capability and on the performance of its business processes. This knowledge enables management to make trade-off decisions

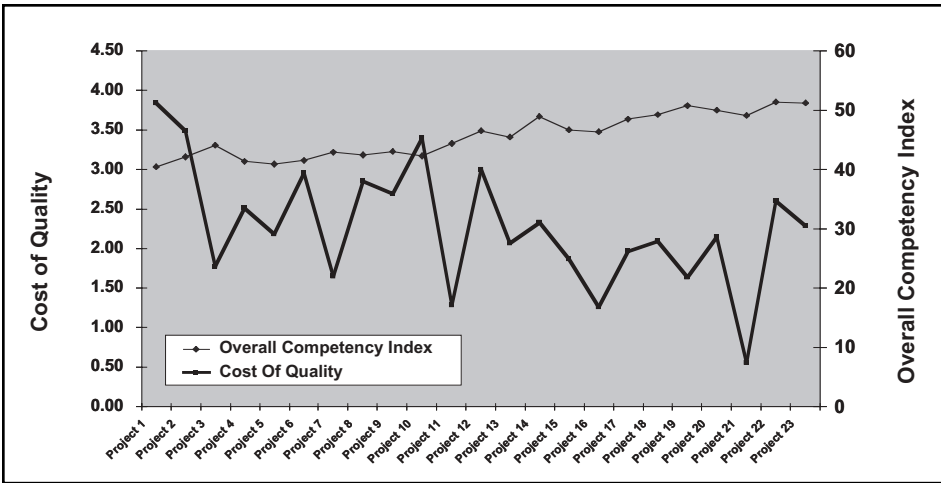


Figure 3: Correlation of Competencies With Cost of Quality at Infosys

regarding investments in work-force practices. For instance, Figure 4 presents a comparison developed by Tata Consultancy Services regarding the percent of time spent in training and its correlation with criteria such as defects per person-hour, review efficiency, effort, and rework.

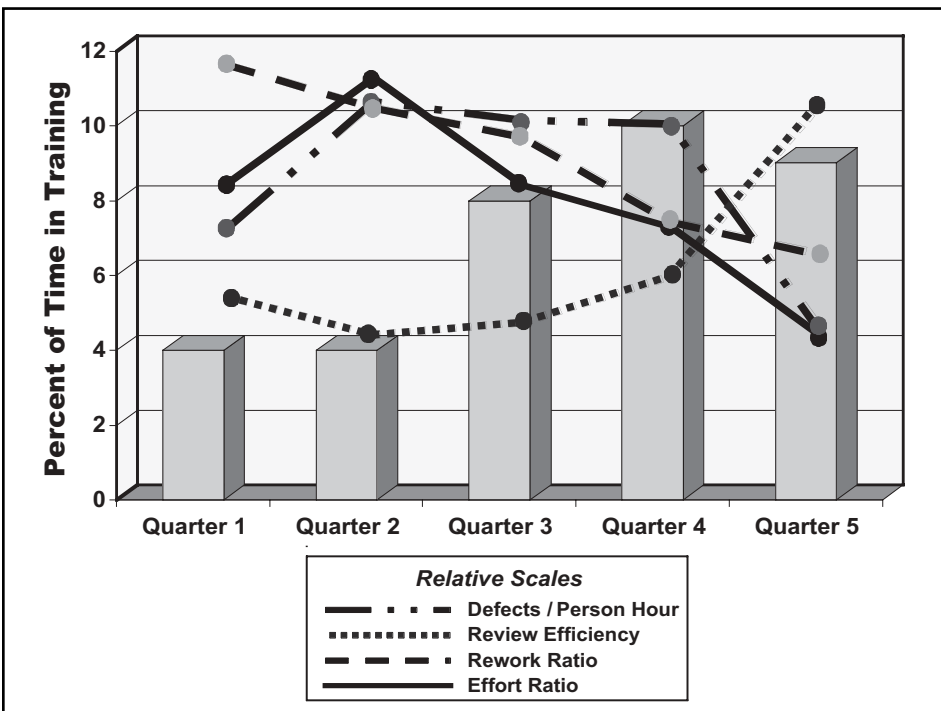
The trends in Figure 4 are all in a favorable direction with various measures of effort and quality decreasing, and review efficiency increasing as training time increases; however, data are needed through more quarters to determine the absolute strength of these relationships. Once the strength of these relationships is understood, and asymptotes or other important trends have been determined, then management is armed with a powerful quantitative tool to make decisions regarding the optimal investment in training. Similar mentoring data identified trade-offs

regarding sending senior people on overseas assignments versus using them as mentors at sites in India. High-maturity organizations are able to adjust their work-force practices to achieve targeted performance objectives using their work force.

Lessons Learned in Applying the People CMM

People CMM-based improvement programs should be conducted as part of an overall organizational improvement strategy. Human resources professionals have stressed that a program based on the People CMM model should not be treated as just a human resources initiative. Rather, it should be presented as a program for operational management to improve the capability of its work force. Professionals in human resources, training, organizational

Figure 4: Relationship of Percent Time in Training to Various Performance Baselines at TCS



development, and related disciplines have unique expertise that can assist operational managers in improving their work-force practices. Nevertheless, the responsibility for ensuring that an organization has a work force capable of performing current and future work lies primarily with operational management.

When introducing multiple improvement programs, the organization needs to assess the amount of change it can reasonably absorb and adjust expectations and schedules accordingly. This is especially acute at Level 2, where the individuals absorbing the majority of the changes are project- and unit-level managers. In order not to overload these managers with change, the organization should stage the introduction of improvement programs. Under many circumstances, project managers should first master project management skills (SW-CMM or CMMI). After acquiring these skills, managers can then undertake improvements guided by the People CMM to supplement their project management activities.

Many People CMM improvement programs start with performance management. While some managers may not have open positions requiring staffing activities, and others may not be involved in compensation decisions, all are involved in managing performance. Implementing improvements guided by the performance management process area have the added advantage of focusing on the relationship between managers and those who report to them, which is critical for retaining employees.

Performance management is also the process area at Level 2 most likely to have near-term effects on productivity, quality, and efficiency, at least at the unit level. Performance management, and especially handling unsatisfactory performance, is typically one of the weakest areas in low maturity organizations. Therefore, improvements in conducting performance management activities often yield benefits for the organization, while getting the entire management team engaged in the launch of a People CMM-based improvement effort.

When an organization achieves Level 3 or higher on SW-CMM or CMMI, it is easier to integrate the People CMM activities simultaneously with process improvements, since many of the higher level process issues have been incorporated into People CMM practices. As organizations progress with multiple capability maturity models, they find that they are able to develop inter-linked architectures for both their business processes and the work-force competencies

required to perform these processes. When implemented effectively, these architectures enable effective execution of the organization's business strategy.◆

References

1. Curtis, B., W. E. Hefley, and S. Miller. People Capability Maturity Model[®]. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1995.
2. Walker, David M. "Human Capital: Building the Information Technology Work Force to Achieve Results." Testimony before the Subcommittee on Technology and Procurement Policy, Committee on Government Reform, U.S. House of Representatives. Washington: GAO, 2001. GAO-01-1007T.
3. McClure, David L. "Human Capital: Attracting and Retaining a High-Quality Information Technology Work Force." Testimony Before the Subcommittee on Technology and Procurement Policy, Committee on Government Reform, U.S. House of Representatives. Washington: GAO, 2002. GAO-02-113T.
4. Curtis, B., W. E. Hefley, and S. A. Miller. People Capability Maturity Model: Guidelines for Improving the Work Force. Reading, MA: Addison Wesley Longman, 2002.
5. Gray, R. People CMM Panel Session: Practical Approaches to Initiating and Sustaining a Successful People CMM Effort. Proc. of Software Engineering Process Group Conf., Phoenix, AZ, 2002.
6. Snyder, C. Initiating and Deploying the People CMM Across Intel's Information Technology Department. Proc. of Software Engineering Process Group Conf., Boston, MA, 2003.
7. Humphrey, Watts S. Managing the Software Process. Reading, MA: Addison Wesley Longman, 1984.
8. Hefley, W. E., and S. A. Miller. Software CMM[®] or CMMSM? The People CMM[®] Supports Them Both. Proc. of Software Engineering Process Group Conf., Boston, MA, 2003.
9. Boehm, B., et al. Software Cost Estimation with COCOMO II. Upper Saddle River, NJ: Prentice Hall, 2000.
10. Clark, B. "Quantifying the Effects of Process Improvement on Effort." IEEE Software Nov./Dec. 2000: 65-70.
11. Deming, W. Edwards. Out of Crisis. Cambridge, MA: MIT Press, 1986.

Notes

1. The People CMM is available as both a technical report from the Software Engineering Institute and as an Addison-Wesley book. For more information, see <www.sei.cmu.edu/publications/documents/01.reports/01mm001.html> and <www.awprofessional.com/catalog/product.asp?product_id={2699E666-10C7-4865-B5DA-01C678D54988}>.

About the Authors



Bill Curtis, Ph.D., is co-founder and chief scientist of TeraQuest. He is a former director of the Software Process Program at the Software Engineering Institute. He is co-author of the Capability Maturity Model[®] for Software (SW-CMM[®]), principal architect of the People CMM[®], and a member of the CMM[®] IntegrationSM product team. Previously, Curtis directed research on user interface technologies and the software design process at MCC, developed software measurement systems at ITT's Programming Technology Center, evaluated software methods for the GE Space Division, and taught statistics at the University of Washington.

TeraQuest
9108 Benview Court
Fort Worth, TX 76126
Phone: (817) 249-2259
Fax: (512) 219-0587
E-mail: curtis@acm.org



William E. Hefley, Ph.D., is a senior lecturer at Carnegie Mellon University. He is a lead assessor for the People Capability Maturity Model[®] (CMM[®]), CMM-Based Appraisal for Internal Process Improvement, and Standard Capability Maturity Model IntegrationSM Assessment Method for Process Improvement. He is co-author of "The People CMM" and its assessment method, and a member of the CMM[®] IntegrationSM product team. He was instrumental in launching the Software Engineering Institute's software process improvement efforts. In prior industry roles, he participated in and managed systems development and user interface projects in areas such as space system; Command, Control, Communications, and Intelligence; and manufacturing systems.

IT Services Qualification Center
Institute for Software
Research International
School for Computer Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
Phone: (412) 268-4576
E-mail: bill.hefley@cs.cmu.edu



Sally A. Miller is co-author of the People Capability Maturity Model[®] (CMM[®]) Versions 1 and 2, and is a senior member of the Technical Staff, Software Engineering Institute (SEI). She is the SEI's lead instructor for the Introduction to the People Capability Maturity Model[®] course and coordinator of the People CMM Lead Assessor Track of the SEI's Lead Appraiser Program as well as an SEI-authorized People CMM lead assessor. Miller manages the SEI's People CMM efforts. She is a guest lecturer at Carnegie Mellon University and a graduate of Grove City College.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
Phone: (412) 268-5678
E-mail: sal@sei.cmu.edu

Obedience Training for Managers

Virginia Slavin
Abbott Labs

Paul Kimmerly
Defense Finance and Accounting Service

Capability maturity model-based process improvements bring about significant organizational change. Managers and change agents are always looking for methods to bring about this type of change. The authors draw parallels between organizational change agents (or process whisperers) and animal trainers. Some of the same techniques used to train animals can be used to lead an organization to a capability maturity model and make them change.

Every software engineering process group (SEPGSM) has faced that moment when it provides management with an important report or agenda only to have the manager say something like, "I want to thank the SEPG for putting together this report, but I'm not going to use it."

The SEPG's immediate reaction is to want to rub his nose on the document and scream, "Bad manager! Use the paper! Use the paper! Bad!" If only it were that simple.

Particularly in organizations with lower maturity levels, the people responsible for causing these behavioral changes can be at a great disadvantage when pitted against the *organizational culture*. However, there are some basic training concepts used in training horses and dogs that can also be useful when introducing organizations and their management to the new behavior required when implementing capability maturity model-based process improvement.

The authors will relate their experience training horses and deaf dogs to illustrate how these same concepts can be used to change behavior in an organization. Managing change requires learning new behaviors and transitioning from an undisciplined environment to one that is controlled. This is not that different from the transition required in teaching a horse to carry a rider or teaching a deaf dog sign language.

The following are some basic steps in making the transition successful whether training animals or changing people's behavior:

1. Get the subject's attention.
2. Take little steps.
3. Always teach in an environment of safety and respect.
4. Never underestimate the power of peer pressure.
5. Understand the motivational influences, and how they are used in each situation.
6. Do not reward undesirable behavior.
7. Make the desired behavior the obvious choice.

8. Failure is just an opportunity for more training.

Get the Subject's Attention

To get the subject to trust you and to be willing to learn, you must first get his or her attention. Find the best motivator for that particular subject. Food treats are often good motivators for both dogs and horses (and people, by the way). The important point is to find a good motivator for the target trainee, whatever it may be.

"There are some basic training concepts used in training horses and dogs that can also be useful when introducing organizations and their management to the new behavior required when implementing capability maturity model-based process improvement."

The Preferred Way: Get a deaf dog's attention by offering food and then teach the sign for *treat*. Repeat this until the animal begins to respond to the sign. This will cause the subject to look forward to interacting with you, in the hopes that it will receive a treat.

The Sub-Optimal Way: Whack the subject across the nose with a newspaper. While this will get its attention, it has a greater chance to startle and cause a fearful reaction rather than a trusting one. The subject will spend time trying to avoid you and newspapers instead of looking forward to seeing you.

Organizational Translation: A good first step is to find out the subject's biggest information need and look for a way to provide it quickly. Managers are always looking for information/data about their projects. One of their biggest frustrations can come from the inability to get good status information. They quickly tire of asking, "Are you done yet?" Find a way to give them a fast response to their information need, and they will come back asking for more. The key concept here is to keep it simple.

Take Little Steps

Each of us is familiar with this concept. In school, we learned the alphabet, and then we learned to read. We learned basic addition before we learned fractions and calculus. We all learned in kindergarten (or from Sesame Street) the basics before we started applying them to more difficult problems. Our training subjects are the same way. They need a foundation upon which to grow, and time to develop from this foundation.

The Preferred Way: Training a horse to carry a rider requires breaking the training into the following steps:

1. Teach the commands for walk and whoa, but especially whoa. Repeat until it is obvious that the horse understands what you want.
2. Introduce the bridle. Put it on the horse, and then go to the next step.
3. Review the commands for walk and whoa. This shows the horse that he can do familiar things with this new tool on his head. Continue until the horse shows no signs of discomfort or confusion.
4. Introduce the saddle. Again, just show the horse that it is a new tool.
5. Review the commands for walk and whoa, reinforcing the lesson learned with the bridle.
6. Introduce the rider, again, just another tool.
7. Mount. Okay, this time the tool does something different.
8. Dismount.
9. Repeat steps seven and eight twice

more. Get the horse familiar with the new roles of rider and horse.

10. Review the commands for walk and whoa. Go back to something familiar to show that this can still be done even with the new tools and roles.

With this method, you are constantly reinforcing ideas the horse is familiar with before introducing a new command. Your horse will learn to do one thing and be comfortable with it, so when you ask for one more thing, it is not such a stretch for the horse. This makes it easier for the horse to understand that it can try one new thing.

The Sub-Optimal Way: Teach a horse to carry a rider by applying all those lessons at once.

1. Catch the horse and throw on the bridle and saddle. This will be hard because the horse will be looking at you like you have lost your mind and at the same time is trying to run away.
2. Try to climb on. This will be hard because the horse will still be trying to run away, and is pretty convinced that you have lost your mind.
3. Try to hold on – provided you are able to climb on. This will be hard because the horse will be bucking in an attempt to throw you and the gear off because you are obviously confused and do not know what you are doing, and therefore must be dangerous.
4. Try to go somewhere. This will be impossible because the horse still does not know walk or whoa or any other command, and certainly has lost all traces of faith in your ability as a leader figure.

In this situation the best you can hope for is to not break anything important. You might win the battle, but in the long run the horse will respond as if to a “crazy idea being attempted by that person again.”

Organizational Translation: If you are tasked with implementing a metrics program, for example, do not expect to implement a full-blown Software Engineering Institute’s Capability Maturity Model® (CMM®) Level 4 metrics program all at once. Start with one or two easy-to-collect and use metrics (in a Level 1 company this could take some creative research to develop), and get those incorporated into everyone’s way of life.

Work on getting everyone to rely on these metrics. The more useful they are, the more motivated people will be to use them. As this change becomes *institutionalized*, add a few more metrics, maybe more complicated measurement concepts, maybe different charts with the

same data. Then get these incorporated into the culture.

People are less resistant to small changes than large ones. If you follow these steps, you should be able to grow a culture where the individuals all rely on and support the metrics that are being used. If individuals start creating and using their own metrics, that is great! The key here is to institutionalize the concept that metrics are good.

Always Teach in an Environment of Safety and Respect

All subjects learn better in an environment of safety and respect. Studies have been conducted that show children under stress or in fearful situations have a harder time concentrating on new lessons. At work, this translates to making sure that your subjects are not defensive or afraid of looking bad in front of their peers and bosses, which can be tricky at times.

“The staff should know that they can share their concerns with the SEPG, and that the SEPG will bring those concerns up to management without naming names.”

The Preferred Way: Deaf dogs rely on their other senses heavily, especially sight, smell, and touch. Never wake a deaf dog suddenly. Allowing the dog to sniff your hand and then applying a light touch will let the dog wake without startling. Deaf dogs also react to visual distractions very easily. When training a deaf dog, find a place where you will not be disturbed, and where there is no chance for sudden intrusions that would startle and distract the dog. Find an environment where the dog can focus primarily on you and will not be visually distracted or frightened by other new things occurring around it.

The Sub-Optimal Way: Take a deaf dog to a public park for training. The park brings a new set of visual and aromatic distractions. Children rushing the dog to pet it, or other dogs coming over to play

can easily startle a deaf dog and shift their focus away from you.

Organizational Translation: Find a manager’s comfort zone. Especially in meetings with his or her peers or bosses, be sensitive to surrounding distractions. Do not use one manager as a bad example of a situation and another as a good example. No one should ever leave a training meeting or change-introduction session feeling like they have been set up or outed to the rest of the group.

Non-attribution should be a strict rule for the SEPG. It is very important for the SEPG to create an atmosphere of trust with managers and their staff. The staff should know that they can share their concerns with the SEPG, and that the SEPG will bring those concerns up to management without naming names. Managers must know that they can share information with the SEPG and that it will not be broadcast to the world. Open and honest communication between the SEPG and the organization is critical in forming that comfort zone for implementing change.

The SEPG should keep in mind that some people often need time to think about new ideas before they are ready to support or embrace them in front of others. The best place to teach new concepts in these cases may be in one-on-one sessions or small groups of peers. Remember that there are no bad managers, just sub-optimal situations.

Never Underestimate the Power of Peer Pressure

Some of us are innovators, and some of us are not. However, those who are not tend to be more inclined to adopt change if they see someone else do it first. Peer pressure can be a powerful *change-enhancer* if used correctly.

The Preferred Way: Horse trainers use this method when teaching horses to jump new obstacles. If a new jump obstacle frightens a horse, a typical way of getting past the obstacle is to have the frightened horse follow closely behind another horse. If the other horse jumps the new obstacle, the frightened horse will usually jump as well to avoid being left behind. As a result it learns that there is nothing to be afraid of and willingly jumps the obstacle by itself the next time.

The Sub-Optimal Way: Singling out a horse for corrective action does not encourage learning. Horses are very social creatures, just like most engineers we know, and it is sometimes very difficult to get one or two to break away from the

herd and *blaze a new trail*. They are strongly motivated to stay with their peers; no amount of begging, pleading, or beating will cause them to willingly break from the herd for long.

However, one must be careful to remember that peer pressure is a motivator, not a tool of force. We want to entice the horse to jump by harnessing his urge to follow. We do not want to scare the horse into jumping by repeatedly pulling him away from the herd for discipline. He will just become more frightened by the obstacle and dig his heels in even more. Similar reactions have been observed at meetings where a manager is dressed-down by his boss in front of his peers.

Organizational Translation: Success by another manager will often create peer pressure. If a manager sees that another manager is able to respond to issues and answer a senior manager's questions quickly and effectively, he or she will want to be able to do the same thing.

Process improvement in one area can often breed process improvement in another area. A manager will want to know how things are done in another area if he or she thinks the same things can benefit his or her projects. By bringing the lessons learned and the examples from a successful project, a SEPG can use peer pressure to build process improvement in a new area of the organization.

Keep in mind that peer pressure tends to be an *instantaneous motivator*, and not a long-term motivator. However, it can be used to get the SEPG's foot in the door with a project. That provides an opportunity for the SEPG to get their attention and build some successes.

Understand the Motivational Influences and How They Are Used in Each Situation

Not all motivators work with all subjects in all situations. The trainer must determine the appropriate motivators to use for a given subject and situation.

The Preferred Way: Use praise, peer pressure, food, and encouragement. As mentioned earlier, food can be a great motivator for a young, deaf dog. When the dog begins to focus on you for treats, you can begin to teach new signs and use food as a rewarding motivator. The correct motivator will be the one that causes the dog to want to give you what you are asking for in return. This causes the dog to constantly look for ways to give you what you are asking for, because it is an enjoyable experience.

The Sub-Optimal Way: Use fear,

pain, peer pressure, and force. Fear and force can easily spook a deaf dog. A frightened dog will not respond to training. The best approach is to find ways to entice the desired behavior. The less effective motivators will be those that force the desired action because once you remove the motivation, the action ceases as well. If a dog or horse is trained to do any action because the trainer adopts an *I-can-make-you* philosophy, then the trainer better be prepared to *make* the dog or horse every time he wants that behavior. Most trainers are limited in how much time and energy they have available to make animals do things.

As an example of different types of motivators, most dogs love to come into a nice air-conditioned house in August when it is 110 degrees outside. While these dogs would normally also love to go outside and play ball or learn new tricks, it is going to take a pretty strong motivator to get that

"By bringing the lessons learned and the examples from a successful project, a SEPG can use peer pressure to build process improvement in a new area of the organization."

dog to go outside at 3 p.m. on one of those days. And kicking them out of the house is not going to necessarily make it easier to get them out the next time, either. What would motivate the dog in this situation? A good trainer needs to know this.

Organizational Translation: Understand your audience and the motivational factors that are already at work. If a manager wants to look good to his boss, and his boss is asking for detailed data in his status reports, help him find a way to do this. Show him the steps to take and the process to implement in order to get what he thinks he needs.

This is the main reason that improvement activities are tied to business goals in the attempt to motivate the use of the improvements. However, there may be times that a manager does not exhibit the expected behavior. It is important in those cases for the SEPG to look for

other *hidden* goals or different types of motivating influences.

Do Not Reward Undesirable Behavior

Everything you do with animals teaches them how to respond to you. People are the same way. Sometimes it is easy to unintentionally teach undesirable behavior.

The Preferred Way: When trying to house break your dog, the easiest way is to put him outside after every nap, every meal, and then every 30 minutes of play. After a while, he will learn to go to the door at these times, and will prefer to *do his business* outside.

The Sub-Optimal Way: Wait until the dog does his business then rub his nose in it, yell at him, and put him on papers. Keep in mind that dogs love attention. By making a big fuss over an *accident*, the dog is getting attention for the wrong behavior. It is an easy trap to fall into. Soon, you will have successfully trained the dog to pee on the carpet then madly run circles around the living room before diving for the papers in the kitchen and waiting for attention. While it will be quite entertaining to the outside observer, it does not exactly achieve the intended goal of the training session. The dog will be confused by all of the shenanigans, but be glad to get some attention.

Organizational Translation: If you are trying to establish process improvement efforts with a manager and he or she continues to cancel meetings or resist suggestions, do not give up. Often managers do not want to start improvement efforts because they see it as one more thing to get in the way of the *real* work. A SEPG cannot force a manager to get involved, but he or she also cannot stop trying to work with the manager. If the SEPG backs off, he or she has rewarded undesirable behavior.

Make the Desired Behavior the Obvious Choice

The right answer should be the obvious answer. Most learners will only offer up so many responses to a new situation before they give up. Conversely, if they offer up the first response and nothing bad happens, then that becomes their answer to the situation, whether it is the desired response or not. See the previous section on Undesirable Behavior.

The Preferred Way: If you want to teach a horse to load into a trailer, set up the situation so that the horse chooses to give the desired response. If you move the horse to the trailer, it has four options: move right, move left, stop, or go into the trailer. Position the trailer so that if the

horse moves right, there is a wall. The horse will not go that way more than once. If the horse stops, apply pressure, which the horse will not like. If it turns right, keep moving it in a circle back towards the opening in the trailer. The horse will tolerate this for a while, but will become bored. If the horse steps into the trailer, the pressure to move forward is removed. The horse gets to stop and rest, thereby receiving a reward.

The Sub-Optimal Way: If you just try to lead the horse into the trailer on the first try, the horse will not want to go. The trailer is dark, cramped, and smells funny, none of which will motivate the horse. If you just stand there and pull the horse forward, it is not a big problem for the horse. It can stand there all day and even take a nap. He can rest his head against the lead rope so you are basically supporting his weight. The horse has learned that if he does not want to go in the trailer, he does not have to. He can take a nap.

Organizational Translation: If you

want a manager to take a desired step, make it the best option. If the manager moves right, have an unfilled senior manager's request waiting there. If the manager stops, apply pressure from a senior manager sponsor. If the manager moves left, keep coming full circle to the same idea. If the manager accepts the idea and takes the necessary step, the senior manager is satisfied and stops applying pressure, plus the SEPG quits being a nuisance about the idea.

Failure Is Just an Opportunity For More Training

If at first the ideas above do not succeed, do not give up. Review your knowledge of the manager, the motivations, and the rewards. Pay close attention to the situational aspects. Review what the manager did learn and formulate a new plan. Start the steps outlined above again. And remember, the longest journeys still begin with single steps. ♦

About the Authors



Virginia Slavin has 16 years of experience in software development and management in multiple industries. Starting at General Dynamics, Fort Worth, Texas, which is now Lockheed Martin, she has worked in various development environments, including embedded, simulation, and tool development. During this time, she was also an assessment team member for several Software Engineering Institute Capability Maturity Model® (CMM®) assessments, and as a result is now an authorized lead assessor for the CMM for Software. Recently, she made the switch from the defense industry to the medical industry by joining Abbott Labs, a medical diagnostic device company, and has since been applying her broad range of experience helping Abbott climb the CMM ladder.

Abbott Labs
P.O. Box 152020 MS 2-30
Irving, TX 75015
Phone: (972) 518-6804
Fax: (972) 518-6844
E-mail: virginia.slavin@abbott.com



Paul Kimmerly has 15 years of experience in software development for the different incarnations of the Defense Finance and Accounting Service (DFAS) Technology Services Organization in Kansas City, Mo. A member of the Software Engineering Process Group since 1993, he has served as the group's chair for the past seven years. In addition to his local duties, he chaired a group that addressed DFAS-wide process improvement issues affecting six software development sites. Kimmerly is currently the chair of the Kansas City Software Process Improvement Network. He presented at the 1997 and 2000 Software Engineering Symposiums and has contributed several articles on process improvement to CrossTalk.

DFAS-KC/TKZ
1500 E. 95th St.
Kansas City, MO 64197
Phone: (816) 926-5364
DSN: 465-5364
Fax: (816) 926-6969
DSN: 465-6969
E-mail: paul.j.kimmerly@dfas.mil

COMING EVENTS

April 28-May 1

Software Technology Conference 2003



Salt Lake City, UT
www.stc-online.org

May 3-10

International Conference on Software Engineering

Portland, OR

www.icse-conferences.org/2003

May 12-16

STAREAST '03

Orlando, FL

www.sqe.com/stareast/

May 12-16

QAI International Quality and Testing Conference

Orlando, FL

www.qaiusa.com

May 19-20

9th Annual

Golf Tournament/Partnership Day



Montgomery, AL

<https://web1.ssg.gunter.af.mil/partnership/>

June 2-6

Applications of Software Measurement

San Jose, CA

www.sqe.com/asm

June 9-13

International Conference on Practical Software Testing Techniques

Washington, DC

www.psqtconference.com

June 25-28

Agile Development Conference

Salt Lake City, UT

www.agiledevelopmentconference.com

People Projects: Psychometric Profiling

Kasey Thompson
Software Technology Support Center

This article introduces a tool known as the Compatibility Identification Set (CIS). The CIS is an approach used to help project managers (PM) build successful project teams by examining the specific job skills of each potential member. The CIS provides the PM with new insight as to how this proposed team will work with one another on a daily basis, and identifies potential problems and personality conflicts before they become exposed in the workplace.

You might infer from the title “People Projects” that this article will cover the softer aspects of software project management; you may be correct depending on what you consider *softer* to be. Nevertheless, the following information attempts to persuade you that people are the chief component in a system also known as a project team – a *hard* element indeed.

Knowing that people are responsible for everything on the planet but the weather and other such divinities is the first and most basic concept required in managing any person, program, project, or idea. Managing teams is an art, not a science, because the essence of what is being managed invariably will always be a person. Please refer to “The Second Law of Consulting: No matter how it looks at first, it’s always a people problem” [1].

This law draws from a greater belief that every problem, issue, philosophy, concept, and process stems from the same source: humans. People are the spice of life, the straw that stirs the drink, and the cause of, as well as the answer to, most of life’s problems. So why are the majority of project efforts spent analyzing and developing software?

Think of both people and software from a systems perspective. Software components are not inserted into an existing system without first knowing what affect they might have on the other components and the system as a whole. Project teams however, are created based on availability or convenience without giving conscientious consideration to how each person and the team will perform when the time comes to roll up your sleeves and actually perform the work. Assuming it is agreed that the person is the most important component in any project, should not equal time be spent defining the human mechanisms of our frontline system, the project team?

Humans are tempered by different belief systems, educations, hopes, dreams, goals, and even different metabolic rates that physically limit how fast people think

and work. So why would project managers (PM) ever think they could pigeonhole employees into the same styles of management, project or otherwise?

A New Approach to Forming Project Teams

How do PMs form and manage great teams? This article introduces a three-pronged approach known as the Compatibility Identification Set (CIS). The CIS is an approach used to bring together defined skills, intangible abilities,

“Project teams however, are created based on availability or convenience without giving conscientious consideration to how each person and the team will perform when the time comes to roll up your sleeves and actually perform the work.”

and the requirements of the project team position to provide a PM with the information needed to form an effective unit.

Before exploring the CIS, please understand that it is assumed the PM knows the formal training and defined job skills of each person in the organization, or that this information is available through the candidate’s current position description or resume.

The three prongs of the CIS are composed of two categorical groupings, candidate traits (CT) and position characteristics (PC); and the existing information

(EI) referred to above as formal training and job skills. Each grouping will be described in detail later in this article, but the following is a high-level description.

The CT assist in defining the intangible traits and abilities of potential project team members. The PC simply define the intangible requirements of the position to be filled. A cross-comparison of the CT, the PC, and the EI presents a focused perspective of what is needed for project success. The CIS brings the three areas together to present a clear picture of who should be selected as project team members, thus giving the project team the best chance for success (see Figure 1).

Identify Your Staffing Needs

The first step is to examine what the project team is to accomplish. Nothing is new here; simply find multiple experts that are available to help you accomplish the project goal, and identify any expertise gaps in which your organization may be lacking. Now you know what you have to choose from and where you may need to fill in some expertise from a different source.

Do not just take who or what is offered to you by your superior. Receiving permission to hand-select a team may take some lobbying efforts on your part, but a project worth funding is a project worth staffing and doing right. Make the point to your supervisor that this minimal amount of time spent building a good team will significantly reduce project duration and cost.

A study in the *Journal of Applied Psychology* found that the more complex the task, the greater the disparity in productivity between highly productive people and average performers (127 percent difference in complex tasks) [2]. Tom DeMarco has stated, “An individual can only succeed to the extent that the whole prospers. And the whole can only prosper to the extent that everyone does well” [3].

The issue of teaming has the greatest single effect both on schedule and budget in the communication-centric software profession. Teaming is also the reason

staffing profiles are a key element in successful cost estimation techniques. Experience, knowledge, environment, communication skills, and even task complexity all play significant roles in the productivity and functionality of project teams. If noncohesive project teams begin with one strike against them, then it seems only reasonable that managers take the time to implement some form of analysis to build cohesiveness and give the team the best chance of success.

More than 50,000 organizations in the United States alone perform psychometric testing as a means of hiring, placement, and promotion [2]. A survey of 500 top executives across the United States reveals that 76 percent of these companies test more than 150 people per year, 35 percent test more than 500 people per year, and 30 percent declared they would increase the practice in future years [4].

Groups such as Chevron Corp., Hewlett-Packard Company, T. Rowe Price Investment Services Inc., Federal Reserve Bank, Wells Fargo & Company, Stanford University, and SBC Communications, Inc. (formerly Pacific Bell) use psychometric profiling to decipher employee skills and abilities. Such profiling methods become essential not only to find out what skills are lacking but also to identify utilizable strengths and potencies in areas previously undiscovered. These same methods also make information available that can be used for building truly great project teams.

An example of this is found in the consultation work performed by Dr. Nancy Haller of Applied Psychometrics. Haller stated, "Team building occurs when there is an understanding of work styles among team members" [5]. Haller uses psychometric testing to elicit data from individuals to learn what areas most interest and stimulate each person. Conversely, these same tests provide insight into what areas of work a person detests or feels apprehension toward. The results allow for an educated placement of the individual not only for their benefit and enjoyment, but also for the improvement of the organization as a whole.

Evaluating What You Have

Once you have identified a list of potential experts and specialists, you need to explore what makes each person tick. While exploring, keep an eye open for individuals with an aptitude for learning or willingness for training. Training existing personnel is usually a less costly option than bringing in outside help. Also, once a team member is trained they tend to share

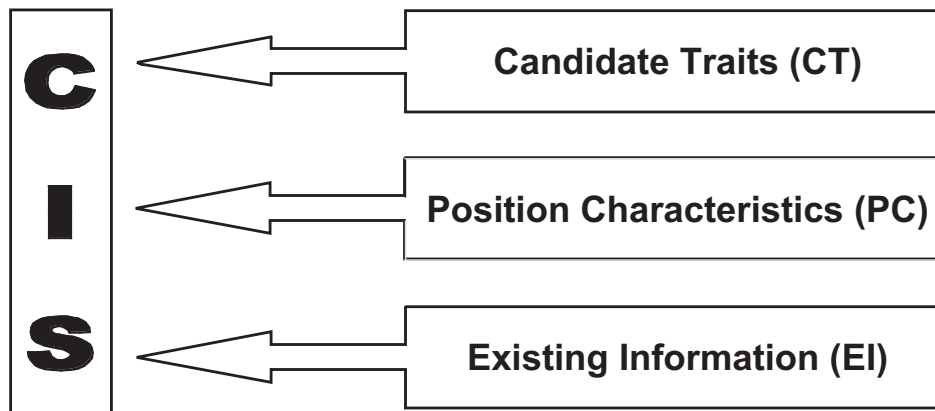


Figure 1: *Compatibility Identification Set*

their learning with others by exhibiting the new skill, allowing others to learn by observation.

Without getting into the discrediting practice of labeling people, the PM needs to identify traits; the traits that may or may not make this person a good fit for your project. This step assumes multiple resources are available. It is understood that this is a luxury, and if it is not one offered by your organization, see the section "Managing With What You Have." Government organizations and larger private industries usually do have multiple resources and if that is the case, each person available should be placed into the CT prong of the CIS.

The CT prong (see Figure 2), the first of three, actually consists of seven groups of traits or specific qualities. The purpose of the CT prong is to identify these traits in potential project team members. Traits should not be considered either bad or good. It becomes apparent that each grouping of traits has positive aspects if used in the correct combination and setting as well as possible negative aspects if no thought is used in the positioning of the person possessing those traits.

It is understood these are generalities and this exercise entails inserting people into predefined groups. However, it is also understood that a great portion of the PM's job is performing these duties by selecting the best person for the job and selecting the best team for the project. Admittedly there is a lack of science in

this process but remember this is an art, not a science. These are people who are associates, people who when treated like a number, an object, or a unit will reciprocate that treatment in the work they perform. Also remember, the CIS is a tool to assist PMs in selecting and managing their number one resource: people.

The seven categories below are used to define how a person interacts with another or a group. Use these categories to classify potential team members to use in conjunction with the data output from the PC prong that will be defined later. Try to find the category that best describes the potential project team member. Each person may be a combination of categories but select only the one that is most descriptive.

Candidate Traits

- **Bridges** [6]. Individuals who allow two or more people to communicate more effectively simply through their presence. Bridges bring people together due to an ability to communicate with a variety of individuals. Bridges put others at ease and have a sociability and responsiveness that invites others to participate. Bridges also increase communications throughout the project by increasing the communication of those around them.
- **Clusters** [6]. Individuals who associate with others of a similar skill set. An example might be quality assurance or configuration management personnel.

Figure 2: *Compatibility Identification Set Traits and Areas*

Candidate Traits (CT)		Position Characteristics (PC)
Bridges	Liaisons	Communications
Clusters	Social Networks	Work Intensity
Heroes	Well-Wishers	Level of Technical Knowledge
Isolates		Problem Solving and Politics

A cluster enjoys talking about work even outside of the office. A cluster is usually very confident and capable when working within his or her domain.

- **Heroes.** Individuals who are motivated by new challenges, lack fear of failure, and look to champion new ideas and projects. Usually lacks a specific expertise but possesses a well rounded, but limited background in many skills. Sometimes a hero is just someone who loves his or her job and enjoys what he or she does for a living.
- **Isolates** [6]. Individuals who prefer little to no personal interaction. Isolates are generally unassuming with meticulous and methodical work habits. Isolates are very structured in their approach and can work to a schedule. It is sometimes assumed that isolates are a hindrance when working in a team and that can be true, but isolates are also dependable, capable, and hard working. Such people can be invaluable on a project.
- **Liaisons** [6]. Individuals who transfer information to others effectively through their communicative abilities. Liaisons crave communication and involvement. Liaisons increase communications throughout the entire project but unlike the bridge that helps others to communicate, the liaison does the communicating through his or her own efforts.
- **Social Networks** [6]. Groups of people who associate together based on social or other life-style type similarities such as race, sex, faith, pay scale, etc. Social network members can be invaluable when working with customers, contractors, or any stakeholder of the same social network. A member of a social network should not be construed as difficult to work with outside of his or her established associations. On the contrary, he or she has already displayed an aptitude for building relationships while working with others. Politics and sociability are a factor on any project, and a person with social skills can open doors and clear paths for project managers when there is seemingly no other route. Do not, however, mistake sociability with good communication skills.
- **Well-Wishers.** Individuals with no vested interest in the project who show no concern with the success of the group or organization as a whole. This person has no malicious intent but rather simply lacks a desire to get involved. In other words, he or she

wishes others well but would rather not get involved. The value here is in identifying the person as a well-wisher and therefore knowing not to enlist their services.

Keep in mind that each person, no matter how neatly they fit into a category, is going to possess a differing level of skills and knowledge (existing information) that must be weighed either with or against their identified traits.

Some work on your part may be required depending on how well you know the person or how long this person has been with the organization. In some cases the person may be a new hire or a transfer. In that case, you may need to contact past supervisors. It is suggested that every candidate be interviewed. The interview questions will be designed by you to elicit information specific to your project. Make

“Receiving permission to hand-select a team may take some lobbying efforts on your part, but a project worth funding is a project worth staffing and doing right.”

this interview short, to the point, and painless for both you and the candidate by being straightforward. Questions should be drafted with the project needs in mind so as to find out if this person is a possible asset. Evidence of the candidate's communication skills will be demonstrated from the onset of the interview, but talk to co-workers and past supervisors as well, after all, everyone has a bad day now and then.

During the interview, take five minutes with each candidate to talk about their goals, likes, and dislikes of the position they are in currently. Find out if they are looking for career broadening opportunities or just to do something they love, or at least something different. Ask questions specific to the project and begin to look for a fit. Even the best of managers do not know all their people have to offer. Likewise, the best managers realize that the skills their people are trained in are sometimes not suited to their personal traits.

Without being rash, take another five

minutes to look over your notes from the interview. Form a profile of your potential project-team member based on his or her track record with other projects, past personal experiences, your personal opinion (managers can actually have these without being sued), and information from the interview that was recently held.

Does this person fit precisely into one of the above categories, or is he or she a combination of two or more groupings? Be bold and characterize this person for placement into a singular grouping. Remember that none of the categories should be viewed as negative. Each type, with the exception of the well-wisher, has value in the correct situation, and all may be indispensable depending on what is needed from the positions on your soon-to-be-formed team.

In just a matter of minutes, you have *profiled* your potential team member and know where to best use this person's abilities for the good of the project. The profiled has recently acquired an unpleasant connotation, but in this case it is favorable for both parties simply because the PM is trying to build a mutually beneficial relationship and place this candidate in a position where they will be successful.

Anyone who has selected another in any professional manner has followed the above steps either formally or informally. The CT prong simply provides a structured method to conscientiously form a profile using defined criteria. For example, you may identify a person as an isolate, and they may also possess the most knowledge on the needed subject matter. The PM must now decide if the isolate and the position are a match.

The position may be an analyst who needs to thoroughly review production reports to remove bottlenecks from a current process. This person may be perfect for the job because he or she likes the solitude of his or her own work area and enjoys problem solving; or, he or she may be wrong for the job due to the fact that this person needs to report findings multiple times per week and dreads the very thought. The PM needs to thoughtfully examine the person's abilities, traits, and the position requirements. The CIS will provide some guidelines to make this a rational decision based on these three factors.

Position Characteristics

The CT prong provided insight into the potential project-team member's traits. Next, the PM will examine the position itself using the PC prong. The PC prong is a group of five loosely defined areas to

help define the characteristics of the position to be filled and more specifically what skills are needed to function effectively in that position. These are nontechnical skills such as personal and group interactions, communications, and mannerisms.

The categories are based on previously observed project teams, project results, and project management methodologies needed to function in a specific arrangement within a larger group [7]. Each area includes questions regarding the responsibilities of the person performing the work. The purpose is to provide a starting point for evaluating possible project team members and the positions they may eventually fill. Here the PM must begin to form specific suppositions as to who should fill this position. The word *suppositions* will undoubtedly raise some eyebrows but there is a need to suppose or assume to provide a basis from which to compare one potential team member with another.

Use the following questions and statements from the PC prong to construct a profile of what the position will require of the person who eventually fills the slot. Ask yourself the questions in each characteristic area pertaining to each specific position on the project team. Multiple *yes* answers designate a strong indication that the position will require that characteristic. A mixture of *yes* answers pertaining to various characteristics indicates a uniqueness of the position and points out the possibility of filling the role with any number of characteristic types. The PC prong, unlike the CT prong, will allow combinations of needed skill due to varied requirements.

- **Communications** [8]. Is this position central to information dissemination? Does this position interact with three or more people on a daily basis? Does this position entail the use of e-mail, phone skills, writing skills, or a strong verbal ability? Is this position responsible for reporting either up or down the chain? If so, an Isolate will not do. Are the people this position communicates with outside this person's particular group? If so, a Cluster may struggle but a member of a Social Network has already proven the ability to communicate with others and may be a good fit in such a position. Communication is essential for most project team members. Reports, conflicts, scope changes, and meeting times all need to be communicated, and it is crucial to keep everyone informed of project status. The best candidate is a Liaison followed closely by a Bridge.
- **Motivation**. Will this position need to

lead others or will this position be required to keep others on task? Is schedule the most important factor to the success of this position? If so, find a person looking to get ahead. Find out what motivates the person who will fill this position. This person may have motivating factors inline with what you are trying to accomplish. Is it money, family, further knowledge or experience, job satisfaction, or is this person just not motivated by anything? The best candidate for a motivation type position is a Hero. A well-placed Hero may cut your work in half but beware, a misplaced Hero may cause scope creep or worse, budgetary problems due to lack of direction from above. Heroes are not happy when they are not busy, and they will find things to do, project related or not. The next best person is a Liaison. A

“The purpose of this exercise is not to take the human element away from the selection process but rather to increase it and stimulate some thinking about the work being performed.”

Liaison may not motivate or direct others but a worst-case scenario is that the Liaison will communicate your energy and direction to others.

- **Problem Solving and Politics** [8]. Does this position require autonomy? Will this person be secluded where support and assistance are concerned? Is there an element of creativity needed? Will the position need to be resourceful or resilient when it comes to keeping on schedule and reducing costs? Look closely with whom this position interfaces. Will this position interact with the media, inspectors, supervisors, contractors, the public, or even report to your boss in your absence? Will the work being completed by this position be reviewed by a large number of people? If so, either a Hero or a Liaison is your best bet unless the people this position interacts with are within a specific Cluster or Social Network.

If this is the case, work to your strengths and use the resources you have in those existing areas to magnify previous relationships and ease working tensions as much as possible. People perceive that they work well with others with whom they are already comfortable or have much in common. Perception is sometimes the best tool to break down communication barriers and solicit other's assistance in solving problems.

Recognize that all projects have political factors (both internal and external), from governmental restraints to personal agendas (admit it, everyone has them). A good manager will identify them prior to, during, and even after the project is completed and mitigate such factors by placing the right person in the right position.

- **Work Intensity**. Does this position involve a constant level of similar and repetitious work? Is the position scoped for a single person or even a small team? Is there a lack of variety in the work performed? Does this position entail intense or extensive analysis? If so, look to an Isolate or a Cluster to do the task. Keep in mind that nothing will shut down a Liaison or member of a clearly defined Social Network faster than seclusion.

Conversely, work intensity may imply heavily multi-tasked positions. Such positions may require multiple meeting attendance, interaction with teams, individuals, and even learning new tasks on a daily basis. Heroes thrive in such situations and become energized by the variety of people and responsibilities that such a job demands.

- **Level of Technical Knowledge**. You are on your own here. You should know best what is needed, technically speaking. If you do not know, find someone who does so you make sure you have the expertise available. It sounds obvious, but projects without the needed resources flounder and slip into schedule and budget oblivion. Sometimes knowledge is disregarded as a resource but knowledge, like people, drives projects regardless if it is digging ditches or launching satellites. The important thing is to define a specific level of needed expertise from which to cross-reference your candidate's skills.

All too often a project team is comprised of people who have listed skills on their resume, or have once dabbled in a subject, but have never really spent any time learning or work-



Get Your Free Subscription

Fill out and send us this form.

OO-ALC/MASE

6022 Fir Ave., Bldg. 1238

Hill AFB, UT 84056-5820

Fax: (801) 777-8069 DSN: 777-8069

Phone: (801) 775-5555 DSN: 775-5555

Or request online at www.stsc.hill.af.mil

NAME: _____

RANK/GRADE: _____

POSITION/TITLE: _____

ORGANIZATION: _____

ADDRESS: _____

BASE/CITY: _____

STATE: _____ ZIP: _____

PHONE: (____) _____

FAX: (____) _____

E-MAIL: _____

CHECK BOX(ES) TO REQUEST BACK ISSUES:

AUG2001 SW AROUND THE WORLD

SEP2001 AVIONICS MODERNIZATION

JAN2002 TOP 5 PROJECTS

MAR2002 SOFTWARE BY NUMBERS

MAY2002 FORGING THE FUTURE OF DEF.

AUG2002 SOFTWARE ACQUISITION

SEP2002 TEAM SOFTWARE PROCESS

OCT2002 AGILE SOFTWARE DEV.

NOV2002 PUBLISHER'S CHOICE

DEC2002 YEAR OF ENG. AND SCI.

JAN2003 BACK TO BASICS

FEB2003 PROGRAMMING LANGUAGES

MAR2003 QUALITY IN SOFTWARE

To Request Back Issues on Topics Not Listed Above, Please Contact Karen Rasmussen at karen.rasmussen@hill.af.mil.

ing within the area of expertise that they claim. This is not to say that people will purposely mislead you; just be sure of what you are getting (caveat emptor).

You can now select a well-designed team once you have gathered all the CT, PC, and EI data. Match candidate traits with the position characteristics that are best suited for each other. The next step is to take the multiple matches for the same position and ascertain who is most appropriate based on the position's technical needs.

There is always the possibility that you will have no matches. If this is the case, you will need to ask yourself if you are setting yourself up to fail. *The right tool for the right job* became an adage because it is true. PMs need to tool their projects with the right people or else take a good hard look to see if they should be taking the project on in the first place. If you were the customer, wouldn't you prefer your potential PM told you that they could not perform the job rather than try and fail while wasting your budget for the year?

The purpose of this exercise is not to take the human element away from the selection process but rather to increase it and stimulate some thinking about the work being performed. Thoughts about who will perform the work and with whom team members will interact on a daily basis to accomplish the work should be addressed prior to the project kick-off date rather than two months into the project.

The other option is training. Because you now have a better idea of what your people can do and where their interests lie, you can logically obtain training for the good of the project and the project team member, not to mention the organization as a whole.

Existing Information

The third prong, EI, includes all hard-copy artifacts and intellectual knowledge of a candidate's past training, experiences, or capabilities. Such descriptions are comprised of résumés, certificates of training, degrees, and observations from you, past teammates, or supervisors. The EI prong comes with one caveat: training certificates and listed résumé skills are not proof of ability or alacrity. Find some demonstration or first-hand account of project-required skills prior to placing a team member in a key position based solely on their claim of talent. Remember, in the past year a president of a Fortune 500 company and the head coach of the Notre Dame football team were fired for lying on their résumés.

Managing With What You Have

A lack of resources places further importance on the selection and placement of your project team members. Even if you are handed a pre-selected team, you as the PM have the opportunity to use them in the manner you see fit. A small, thoughtfully formulated and structured team can outperform the largest of its counterparts. By simply using the CIS to identify each individual's traits and evaluate your team's abilities, you can match them with the tasks that need to be accomplished.

Again, use training, if available, to fill in the gaps where expertise or knowledge is lacking. Another stopgap measure to use when you are understaffed or lacking resources is to merely rely on the existing capability of your team. Sounds novel doesn't it? People are resilient and resourceful. Use them. You have been given a team of people to accomplish a task and although you may not have what you think you need, you do have a fully functioning group of self-reliant, imaginative, and yes sometimes even ingenious people. In fact, humans are the same species of animal that created all this software stuff in the first place.

Gather your team together for an informal *brainstorming* session. You say it sounds outdated. When was the last time you were a part of such a session where no good ideas were formulated? It is a widely held belief in the PM community that the majority of all the world's problems could be solved in just such a meeting. Just as a side note, another name for these brainstorming sessions is resource management.

Bringing It All Together

As the PM, you are the common thread and the sole person to give direction to the project [9]. You also have the ability to elicit ideas from your project team to gain insight as to where the project is heading. You have taken the time to construct a complimentary team, one that will function to its member's strengths. Now listen to your professionals and use their knowledge to further your own. Use every occasion presented to you to benefit from the talent that you have incorporated. If this practice sounds opportunistic, it is. It is also what makes a good PM great.

The CIS will allow you a much better understanding of your team's capabilities. Once you have positioned your Heroes, Isolates, Liaisons, Bridges, and others, take the time to bring the team together and discuss in a group setting what is expected of each person. Make known the capabilities

of everyone involved. What good is expertise if a member of the team who needs assistance is unaware that the capability exists within the group? Explain that crossing over to assist each other is a welcome practice. Portray the idea that the group succeeds or fails together. Then all you have to do is the actual work. ♦

References

1. Weinberg, Gerald M. The Secrets of Consulting: A Guide to Giving and Getting Advice Successfully. New York, NY: Dorset House, 1985.
2. Hunter, John E., Frank L. Schmidt, and Michael K. Judiesch. "Individual Differences in Output Variability as a Function of Job Complexity." Journal of Applied Psychology 75.1 (Feb. 1990): 28.
3. DeMarco, Tom. The Choir and the Team. New York, NY: Dorset House Quarterly, 1995.
4. A Human Investment Consulting Firm. Facts and Statistics. 2000. Human Investment Consulting. 30 Jan. 2003 <www.humaninvestment.com>.
5. Haller, Dr. Nancy, Applied Psychometrics. Personal Interview 16 July 2002.
6. Robbins, Steven P. Organizational Behavior: Concepts, Controversies,

Applications. 6th ed. Englewood Cliffs, NJ: Prentice Hall, 1996.

7. Pearce and Robinson. Strategic Management, Formulation, Implementation, and Control. 6th ed. Boston, MA: Irwin McGraw-Hill, 1997.
8. Stoner, James A. F., and R. Edward

Freeman. Management. 4th ed. Englewood Cliffs, NJ: Prentice Hall, 1989.

9. Smith, Larry W., and Kasey S. Thompson. "Why This Project Is Like Herding Cats." Software Technology Conference, Salt Lake City, UT, 2002.

About the Author

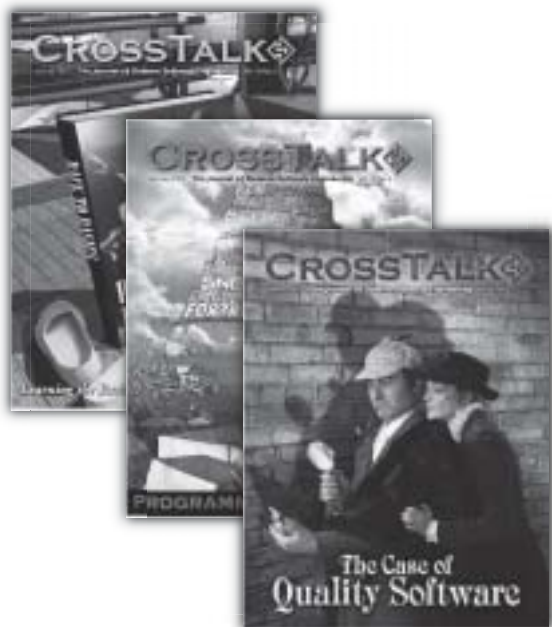


Kasey Thompson is manager of the Legacy Program for the U.S. Air Force and a project manager for the Software Technology Support Center at Hill Air Force Base, Utah. He manages the Air Force's legacy language maintenance and conversion program and provides oversight on a mixture of projects that span from process improvement to cost estimation with customers as diverse as Arlington National Cemetery and NASA. Thompson supplies business structure and team building consultation to private industry and Department of Defense organiza-

tions. He recently developed the Compatibility Identification Set as a tool to build better, smarter, and more cohesive project teams. Thompson has a bachelor's of science degree in lifestyle management from Weber State University and a master's degree in business administration from the University of Phoenix.

Software Technology Support Center
 OO-ALC/MASEA
 6022 Fir Ave., Bldg. 1238
 Hill AFB, UT 84056
 Phone: (801) 775-5732
 DSN: 775-5732
 Fax: (801) 777-8069
 E-mail: kasey.thompson@hill.af.mil

CROSSTALK 
 The Journal of Defense Software Engineering



VISIT US AT THE STC

Be sure to visit CROSSTALK at our information kiosk and at booth 403 at the Software Technology Conference in Salt Lake City April 28-May 1. We will be giving away *Evolution of Programming Language* posters, back issues, and Author Packets, plus a new CD-ROM containing the entire 2002 volume of CROSSTALK.

CROSSTALK, The Journal of Defense Software Engineering
 801.586.0095 • DSN 586.0095 • www.stsc.hill.af.mil/crosstalk



Prospecting for Knowledge[®]

Jüris Kelley

Knowledge in Motion LLC

This article describes the four pillars of knowledge management – content, people, process, and technology. It discusses the challenges of managing tacit knowledge, as well as overcoming the critical obstacles to ensure portal success. Knowledge harvesting approaches are presented, along with how to perform an effective knowledge audit.

It is often said that people are an organization's most valuable asset. While that remains true for some, others find the pendulum has swung toward the organization's other key asset: information, or more accurately stated, *content* (data and information).

Some may disagree. They are quick to point to the Internet's vast holding of content and how useless it can seem as you drown in data yet thirst for information. Those people, however, do a great injustice to modern knowledge-management tools and related solutions that can be used to harvest valuable information from such vast content repositories. They are like the Luddites¹ of the British industrial revolution who opposed technological change; they will even destroy any laborsaving system that may diminish their sense of employment.

Knowledge management (KM) is not so much about managing tacit (existing) institutional knowledge as it is about managing the disparate content sources and providing information-harvesting capabilities to our teams. KM is about optimizing the communities of practice or other work groups and enabling them, through information technology (IT), to efficiently manage the enterprise's content and their own work group practices and procedures. KM attempts to eliminate the continuous reinvention of the wheel by providing repositories of best practices and knowledge nuggets. A KM tool kit provides us with the ability to efficiently gather, contribute to, organize, distribute, collaborate, and refine information.

We focus on information for a reason; knowledge itself happens only when human experience and insight are applied to data and information. As Charles West Churchman concluded more than 30 years ago, "Knowledge resides in the user and not in the collection of information. It is how the user reacts to a collection of information that matters" [1].

Our goals, therefore, are to leverage

the knowledge held by our employees (often referred to as *corporate memory*) and to make effective use of enterprise content to enable others to efficiently create knowledge.

KM Defined

Even though I am more of a practitioner than a theoretician, I am often asked for a definition of KM. I must first admit that I am not a big fan of that term, for it is impossible to manage *knowledge*. KM is more of a *new economy* buzzword than anything else is. If you want to call the concepts of effectively managing content and collaboration *knowledge management*, then so be it.

"A KM [knowledge management] tool kit provides us with the ability to efficiently gather, contribute to, organize, distribute, collaborate, and refine information."

With that in mind, I have assembled this definition: "Knowledge management is a concept that combines content (data and information) with organizational processes and people, as well as the technologies that enable their effective use." It is a concept only, for we cannot manage knowledge itself. Knowledge exists between our ears. What is both attainable and desirable in today's information-based economy is to provide the right content to the right people at the right time, thus allowing people to leverage their tacit knowledge with timely content to effect organizational decision making for a competitive advantage. KM is the

fusion of content, people, processes, and technology.

From an IT perspective, KM combines records management, databases, workflow, and middleware tools, along with collaborative concepts and process improvement philosophies.

It is important to recognize that KM is a concept, or as others have suggested, a management practice, notion, or process. It clearly is not a software product, or a technology, or any single methodology.

Knowledge is something that has been widely discussed in religion as well as in scholarly and political pursuits throughout history. In the Bible in Job 34:35, "Job speaks without knowledge; his words lack insight." Chairman Mao Tse-tung in "On Practice" in 1937 wrote, "All genuine knowledge originates in direct experience." Similarly, Islam discusses knowledge in detail. The Koran teaches that knowledge depends on the use of our sight, hearing, intelligence, and other senses. It further correlates the stages of human existence with the three sources of knowledge.

Whether the Bible, the teachings of Chairman Mao, or the Koran, they would all agree that knowledge is not manageable, and it certainly does not reside in e-mail or a database.

Four Pillars of KM

Content represents one of the four pillars that must be effectively managed and optimized throughout the enterprise. The other three pillars are people, processes, and technology. These four represent the four pillars of Knowledge Nirvana[®] (see Figure 1).

The need to optimize our data and information content should be intuitive. However, most organizations today only effectively manage their operational data, so long as it resides in a database. The document is the default format in which we create, store, and share information, and most of these documents reside on unmanaged PCs.

Improving employee efficiency is yet

© 2002 Knowledge in Motion LLC. All Rights Reserved.
[®] Knowledge Nirvana is a registered mark of Jüris Kelley.

another area in which most organizations need drastic help. Employees can walk out the door or be transferred at any time, taking with them your organization's corporate memory. This issue is only subjectively addressed by most organizations.

Processes, either structured or ad-hoc, are at the heart of any functioning enterprise. Content is the subject of most processes and may reside within the process. Processes, especially the ad-hoc informal processes, build a trust relationship between those in the organization that may not be apparent to management or even to themselves. This *social network* builds trust between people more quickly than formal, mandated processes ever could.

Finally, technology, which is the great enabler that allows us to achieve our goals, compresses the time and effort of processes and allows us to effectively manage enterprise content. There are many such technologies at our disposal. Document and records management repositories are useful for the majority of unstructured content. Workflow products allow us to manage processes. Portals, although a generic term that is comprised of many technologies, represent a key software solution that all enterprises should leverage today.

Decision support systems, which have been used for more than a decade, attempt to aggregate large quantities of structured data to drive automated analysis and provide management with decision options. Like portals, decision support systems, which are also referred to as *business intelligence systems*, is really a generic term for a number of technologies, techniques, and specific software tools that attempt to support cognitive reasoning. Decision support systems are usually comprised of tools and applications that perform data extraction, transformation, data loading tools, data warehousing, data modeling, and query and reporting capabilities.

When faced with an issue, people can efficiently think cognitively and intuitively using their decision-making skills to assess the situation and establish a course of action. People easily use cognitive reasoning (learning from the past to make decisions for the future).

Decision support systems, however, have failed to support cognitive decision-making. This is largely due to their limitations to act in real time. Today's online environment in which we act at the speed of thought has all but shattered the dream of automating the decision process. Nevertheless, the technologies

developed and refined during the years under the banner of *decision support systems* are invaluable today as tools that perform functions such as data extraction and transformation.

Tacit Knowledge

Prospecting for knowledge is like Charles Dickens' musings: "... we had everything before us, we had nothing before us ..." [2]. We have vast knowledge before us, yet we are challenged to find it.

Employees accumulate knowledge – tacit knowledge – as they perform their jobs. They develop skills, certain expertise, and understandings during the course of their duties, both with their current employer and with prior employers. Consciously or not, most employees establish their own *best practices*. While this may be useful, it has limited value to the organization as a whole; others must go through their own experiences and eventually build on their own lessons learned to create a set of best practices.

Making this tacit knowledge explicit is a key objective of any KM initiative; however, this is nearly impossible to achieve. While there are some occasions during which we attempt to capture and manage this tacit knowledge, which is described later, we are left for the most part with a more humble objective of capturing the work products of those employees. In this respect, prospecting for knowledge is a little misleading; you do not really seek knowledge, you seek content. Within that content resides critical business information that either forms the foundation for, or is used by, the knowledge that exists between our ears.

Relevant Content

Most people focus on building an enterprise portal or other such application to manage their content. They have either performed a cost-benefit analysis or have agreed to its intrinsic value, and so they proceed. Performing the requirements analysis, designing the architecture, building the application, and deploying the system are the relatively easy tasks. The real challenges are in acquiring the content and ensuring that the content remains relevant.

I worked with one large IT company on the East Coast that decided to deploy a repository for one of its departments. It was clearly a good idea and could have offered great value to its employees and enhanced departmental productivity. When the portal-based repository became operational, there was enthusiasm to populate it with content. That

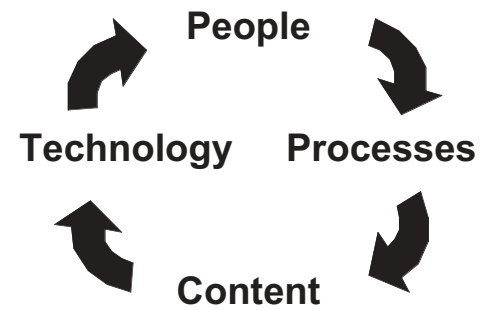


Figure 1: *The Four Pillars of Knowledge Nirvana*

enthusiasm lasted about one week. After that, people had their *real jobs* to do and started neglecting the repository. This is a key challenge for most repositories: ensuring that content is continuously contributed.

There are two primary ways to ensure that employees contribute content: the carrot-and-stick approach and the process approach. A third way – intimidation – has also been known to work.

The carrot-and-stick approach is intuitive and simple. Unfortunately, it also rarely works for very long. That East Coast IT department with its portal-based repository used the carrot-and-stick approach. The portal was operational for about one month when they realized that it was getting harder and harder to get employees to contribute content. Some employees who had been enthusiastic when the repository first became operational began to significantly reduce the amount of content they were contributing. Consequently, they had to be prodded with verbal requests for content. The time and therefore the cost of such prodding grew and finally resulted in the need for a full-time system administrator. Management then tried another tactic. They started offering rewards for employee contributions. Posters were created and hung in the hallways and elevators announcing the rewards – a drawing for free travel. The grand prize was a free three-day trip to Phoenix, Ariz., for two. Each contribution to the repository entitled that employee to one chance to win.

That was not a bad idea; however, such an approach will have a minimal and short-term impact. An organization cannot afford to offer such prizes forever, thus limiting the carrot. Furthermore, the quality of contributions tends to decrease since a few employees will contribute like crazy while others will not. Those who do not contribute as readily tend to guard their content more closely and do not contribute anything of any great value.

While the carrot-and-stick approach

may be useful in getting employees excited about a new corporate initiative, for any long-term success you will need to incorporate the repository into the business process; hence, the process approach.

Each business process should be defined and carried out in accordance with the approved process. Hopefully those processes have been optimized through some form of business process analysis. It would then be fairly easy to append the process to include copying and registering key documents into a repository.

That East Coast IT firm eventually did update some of its defined processes to further facilitate capturing content. They chose selected key processes such as the client engagement proposals, which followed a defined process of proposal development. At the end of that proposal development process, new steps were incorporated to ensure that the proposal and related supporting documents were properly captured in a proposal repository. Capturing that content consistently and repeatedly later proved to be of great value for content reuse.

With either approach, it is important to quantify knowledge sharing. This will allow management to assess the success of the organization's sharing or the lack thereof.

Process Approach

There are other ways to prospect for knowledge both within the enterprise and externally. Besides relying on employees to contribute their content, you can actively seek out that content and automatically populate a repository.

If your organization utilizes a document management system or even a simple file system on a shared server, you have the opportunity to access those files. Remember that every organization serious about productivity should have a document management system. There are no excuses for not utilizing this basic tool.

There are many products available that allow you to define numerous file directories for automatic scanning to identify newly added or changed files. Once tagged as new or changed, they can be copied into an actively managed repository. The challenge with this method is to properly index those documents. While there are automatic indexing tools, they all have limited use due to their relatively high error rate. A better approach would be to utilize the metadata entered by the author within the application; e.g., entering metadata within Microsoft Word by using the Properties Summary function to enter author name, an abstract, and key words.

As long as you have access to the files,

it is possible to copy them into a repository. For the most part, the only time you will not have this option is for locally stored files such as those on each employee's hard disk drive. Unfortunately, this is where most individuals store their content and as stated earlier, this practice should not be tolerated by any modern organization.

External content can also be captured with Internet-based tools often called *spiders*. A spider crawls into a Web site and copies all or selected content into your repository. Many organizations point their crawler to their competitor's Web site thus getting near real-time information on posted changes, such as new press releases. Obviously this has limited use given the highly controlled nature of Web content on corporate sites. Nevertheless, it is yet another tool that you can utilize to capture content.

Institutional Knowledge

There are other occasions when organizations actively prospect for knowledge. One of the most popular times is just prior to losing a valuable employee either through retirement, transfer, or termination.

Given the immense value in that employee's mind – their institutional knowledge – many firms offer employees an incentive to share that knowledge before they go. This is often performed by a recorded interview of the employee. This interview, whether or not it is videotaped, is referred to as *knowledge harvesting*.

Digitizing and storing the videotape is of limited value unless it can be keyed to find specific information at a later date. Otherwise, some tapes could take 20 hours to view, making them virtually useless in today's hectic business environment. One solution is to transcribe the interview word for word, providing a means to adequately search the content. When a hit is found, the application could provide a short synopsis of the surrounding text, or jump directly to the video segment that relates to the hit.

Another approach is to inventory the employee's video content in parallel with recording. Additional metadata and context could be added afterwards with the employee's guidance. This documented information then becomes considerably more searchable, shareable, and useful.

An organization also may want to capture content for high-value, repeatable, decision-making processes. It may be worth the investment to track and document this decision-making process. For example, technicians fix equipment in the field, which is a repeatable process.

Having support engineers or other technicians working in the field is relatively expensive. Small savings in their time would add up to significant savings in labor and travel costs. While every equipment manufacturer has repair manuals, any good field engineer has his or her set of notes that are relied upon much more than published manuals. As you can imagine, the value of these notes is high. Collecting them from various field engineers and assembling them into a collective document that all technicians could use would result in a very high-value, knowledge-sharing initiative.

In your organization, think about what institutional knowledge exists in employees' minds, and likewise what tangible, critical information exists in employees' possessions. Their documents, e-mail, and even scraps of paper in notepads may turn out to be much more valuable than every document submitted by the new programmer in order to be eligible for that free trip to Phoenix.

Knowledge Audit

Employees' collective knowledge is an organization's most valuable resource, yet organizations spend a great deal of time prospecting for that knowledge and its byproduct: information. A small and nimble organization thus tends to be better adapted at harnessing this knowledge, while large organizations can only benefit from that collective knowledge if they overtly harness it. To do so, the large organization must proactively encourage knowledge sharing. Sharing such knowledge and related quantifiable content such as white papers, reports, briefings, etc., all require an organization with a high *organizational IQ*. The more mature an organization, the greater its sharing and reuse of content and employee knowledge.

One way to start a knowledge-prospecting effort is to perform a *knowledge audit*. A knowledge audit will reveal what knowledge the organization has, how it flows, what the sharing and collaboration obstacles are, and what technology and infrastructure exist to enable such knowledge sharing. The knowledge audits will likely focus on processes and information flows just as much as documenting what knowledge currently exists in the organization.

The knowledge audit may take a macro view focusing on high-level repositories, flows, and general cultural attitudes within the organization. A more detailed audit may also be performed that would analyze the specific knowledge and content assets, information flows, and bottlenecks.

Once the audit has been performed, with its findings reported to management, the knowledge prospecting team will have a much clearer understanding of their project's scope and the challenges that lay ahead.

Not sharing the collective knowledge that exists is lost knowledge, and lost knowledge is squandered capital. ♦

References

1. Churchman, Charles West. *The Design of Inquiring Systems*. Basic Books, 1971.
2. Dickens, Charles. *A Tale of Two Cities*. Book One. Signet Classic, 1859.

Notes

1. These were people of any of a group of British workers who, between 1811 and 1816, rioted and destroyed laborsaving textile machinery in the belief that such machinery would diminish employment.

About the Author



Jüris Kelley is president of Knowledge in Motion LLC and author of the new book "Knowledge Nirvana[®]," Xulon Press, from which this article is adapted. Kelley focuses on content management and optimizing team collaboration to achieve the organizational competitive advantage. Kelley has consulted leading public-sector organizations and Fortune 100 companies, and has held project- through executive-level positions in virtually every aspect of the information technology life cycle. During the past 15 years, Kelley has designed, developed, and installed numerous systems, ranging from small departmental applications to several international solutions, each serving more than 20,000 users. Kelley is a technology mentor with the University of Maryland's Smith School of Business and serves on the American National Standards Institute C.22 Standards Committee for Electronic Content Legality.

Knowledge in Motion LLC
11234 Leatherwood Drive
Reston, VA 20191
Phone: (703) 216-5125
Fax: (703) 391-0758
E-mail: kelley@kinm.com

LETTER TO THE EDITOR

Dear CrossTalk Editor,

I thoroughly enjoyed the article "Evolutionary Trends of Programming Languages" by Lt. Col. Thomas M. Schorsch and David A. Cook, Ph.D. (CrossTalk Feb. 2003). I thought it created a clear and concise description of where computer languages have been, and where they are likely to go.

Even though I have been an information technology professional for more than 25 years, it is very difficult to keep up with all the trends in the technology. This article has clarified many

questions I have had over the years about languages and their uses. I was particularly interested in the authors' discussion of the differences and roles of system programming versus scripting languages. It has changed my whole view on how enterprise architecture needs to be addressed.

Although I have not seen your publication before, you can be assured I will check your Web site on a regular basis.

Keep up the good work.

James Blackburn
Independent Consultant

WEB SITES

Software-Engineer.Org

www.software-engineer.org

The Software-Engineer.Org promotes its site as a community for software engineers. Its objective is to improve communication about software engineering. The Web site is dedicated to free information sharing between software engineers, professionals, faculty members, and students. The site features links, articles, tools, downloads, a message board, active discussions, and more.

People Capability Maturity Model

www.sei.cmu.edu/cmm-p

Developed by the Software Engineering Institute, the People Capability Maturity Model[®] (People CMM[®]) is a framework that helps organizations successfully address their critical people issues. Based on the best current practices in fields such as human resources, knowledge management, and organizational development, the People CMM guides organizations in improving their processes for managing and developing their work forces.

Software Technology Support Center

www.stsc.hill.af.mil

The Software Technology Support Center is an Air Force organization established to help other U.S. government organizations identify, evaluate, and adopt technologies to improve the quality of their software products, efficiency in producing them, and their ability to

accurately predict the cost and schedule of their delivery.

bizjournals.com

www.bizjournals.com

Bizjournals.com compiles business news from 41 local markets and 46 different industries. News can be searched by industry, including software. The site includes sections for the latest news, business resources, marketplace, networking, and more.

Amplifying Your Effectiveness

www.ayeconference.com

The Amplifying Your Effectiveness (AYE) Web site contains information to participate in the next AYE conference. The conference is designed to increase effectiveness in leadership, coaching, managing, influencing, and working in teams. The AYE Conference is for people who work in arenas where problem solving is a key skill – such as systems development, product development, quality assurance, information technology infrastructure, customer service, and consulting.

Project Management Institute

www.pmi.org

The Project Management Institute (PMI) has more than 86,000 members worldwide. PMI establishes Project Management standards, provides seminars, educational programs, and professional certification for project leaders.

Project Expectations: The Boundaries for Agile Development

Diana Mekelburg
Extreme Project Management

How can you manage agile development when its practitioners value “working software over comprehensive documentation” and “responding to change over following a plan”? Control the boundaries. Manage the project expectations instead of waiting for requirements and plans to miraculously stabilize.

Isabella, a software development executive, listened quietly while Ravi, the project lead, reported what terrific success the project team was having with the eXtreme Programming (XP) methodology. Sitting next to him, the project sponsor smiled and nodded. Months before, when the software group had proposed using agile software development on the project, Isabella had approved their request without realizing that they intended to use something with *extreme* in its name. She also had not realized that the only requirements and planning documents that she would see would be very high-level – too sketchy in her opinion to control the project.

When Ravi finished regaling the meeting participants with stories of the project’s wonderful atmosphere of camaraderie, Isabella asked whether the system would be ready to install by the deadline. Ravi said that he could see no reason why it would not and flipped forward to the slide of his recently revised high-level schedule. Isabella felt her stomach churn as she searched the schedule for deliverables and found only activities. Terms like *refactor*, *user stories*, and *iteration* made her wonder whether they were building software at all or, like their earlier experiments in object-oriented design, were endlessly refining object components that no one would ever use.

She asked Mary, the project sponsor, what she thought of the project. Mary replied that she was very pleased with the demos that she had seen so far. When Isabella asked her if she thought the project was on schedule, Mary turned to Ravi for the answer. Isabella had heard enough. After the meeting had ended, she told Ravi privately that from now until the end of the project she would meet with him weekly, instead of monthly, to review the project’s status. She ordered him to bring complete requirements specifications and a detailed project plan to the next meeting for her review. After some heated dis-

cussion, Ravi stormed out of the room.

Later, Isabella told one of the other executives, “You would think that after all the money we have spent on training this organization in project management, at least a few of them would understand the concept of project scope.”

Agile Scope Management

Project scope management can take on a whole new philosophy and appearance with agile software development. This can be a formidable challenge for tradi-

“Traditional scope definition has always been a thin security blanket that cannot protect software projects from the crashing waves of scope change in volatile projects.”

tionally trained managers like Isabella. Although the intent of agile software development is to produce the best product possible in the least amount of time and for the least amount of cost, the result is often scope management that appears to be more improvisation than controlled execution. In these cases, the traditional decomposition approach to project planning and prediction is not possible.

In traditional software development methodologies, product scope is typically defined in a top-down manner, starting with high-level requirements that are decomposed to more specific requirements. The project manager can use a

parallel approach for defining project scope by building a work breakdown structure. This approach gives management a progressively more accurate estimate of the time and cost to complete the project, i.e., as the product and work are specified in greater detail, the project estimate becomes more accurate. Once the project’s scope baseline has been set, software managers’ main concern for managing scope is to guard constantly against scope creep, especially in the form of product feature changes.

These methods work well when the product definition is not too complex, controversial, or volatile. However, in many cases, the product is excessively difficult to define, and these methods are unreliable, misleading, and conflict-ridden. It is no wonder that software developers are willing to adopt a lighter, potentially more effective approach, such as agile software development. But where does this leave scope management?

The Agile Software Development Manifesto values working software over comprehensive documentation and responding to change over following a plan [1]. As Alistair Cockburn explains, requirements can be imperfect, and design documents and project plans can be out of date, yet the project can still succeed by applying such principles as communication and community. This can leave the traditional software project manager adrift in a sea of change, clinging to a frail life raft lashed together from in-person visits, whiteboard sketches, invention, and light-and-sloppy methods [2]. As scary as this image is, it is not new. Gause and Weinberg explored the notion that requirements documents are less important than the process of defining them back in 1989 [3].

Traditional scope definition has always been a thin security blanket that cannot protect software projects from the crashing waves of scope change in volatile projects. Even under contract, software scope is subject to disputes and threats of litigation. Traditional scope documents such as requirements defini-

tions and project plans do give software management a starting point for negotiating scope changes. Without these, the manager of an agile software development project seems to have little to bring to the negotiation table.

Outcome Expectations and Scope Management

The flaw in both traditional and agile approaches is the assumption that project success is determined by delivering specific product features, whether they are defined hierarchically through decomposition and change management or through collaborative iterations. Software development projects are successful only when they have met the stakeholders' expectations, the most important of which are not limited to specific product features.

There are three classes of stakeholder expectations: business outcome, project conduct, and product. Software development invariably focuses on product expectations as early in the project as possible. Meeting specific product expectations is more predictable, manageable, and, for software builders, more enjoyable than trying to meet either business outcome or project conduct expectations. However, this focus on the product exacerbates scope management problems, especially in iterative development such as XP. Repeated efforts to get the product functionality just right can lead to extraneous functionality or more elaborate functionality that works against stakeholders' other expectations. The key to controlling scope in iterative development is committing to business outcome expectations as the outer boundaries of scope.

Business outcome expectations are the effects that stakeholders expect the software development project to have on internal operations and/or on marketplace or other external environments. An example of an operational outcome expectation is reducing inventory errors in manufacturing. An example of an external outcome is capturing a segment of the personal digital assistant (PDA) market. Outcome expectations may align with corporate strategic goals, depending on the clarity, viability, or influence of the strategic goals.

Delivering the right software product features depends ultimately on whether those features support the business outcome expectations. This is especially important in agile projects. Without stable requirements to bolster or burden them, agile software development proj-

ects need clearly defined and committed business outcome expectations to contain them.

A key control device in XP, for example, is the story. Each increment of the product is planned to implement a story that represents a set of user functionality. The project sponsor, other users, and the development team decide jointly which story is to be implemented next, and how it will be implemented in product features. The sequence of stories can wander far from the original intentions for the project. Similarly, the features chosen to implement each story are defined iteratively and can also wander. In cases where either the development team or the customer is committed to some limited budget or timeframe for the overall project, this wandering can lead to problems in funding and deadlines.

XP and many other types of agile projects are expected to shift direction.

"The project team, which includes the project sponsor, commits to selected business outcome expectations early in the project and repeats this process throughout the project as major changes in the business environment occur."

Requirements volatility is a primary reason for selecting extreme methodology. However, changes can be contained. If the project team has collected, validated, and committed to meet a set of compatible and feasible business outcome expectations, they can use them to open negotiations about which shifts in direction to apply. They compare each iteration plan and/or user story to the committed expectations. Product functionality that is in line with the overall committed expectations are changed routinely, while changes that contradict or modify the committed expectations are handled as major changes in project scope.

The project team, which includes the project sponsor, commits to selected business outcome expectations early in the project and repeats this process throughout the project as major changes in the business environment occur. The wording for a business outcome expectation is, "As a result of this project, [some group] will be able to [do something]." This simple statement is supported by a collection of justifications, criteria, and evaluations. Outcome expectations do not refer to a specific product feature, nor do they define specific business functions. This allows the project team leeway in selecting the best detail solutions to meet the expectations.

Not all expectations can be met. Many compete for resources, a few directly contradict each other, and some are not justifiable. The most challenging step in committing to the project's outcome expectations is selecting the expectations to be met. Expectations that are rejected or deferred introduce risks to the project in terms of distractions and competition. The primary objective of commitment is a consistent vision of the project's success in terms of the expectations that the project is committed to meet. A secondary but crucial objective is to develop a plan for mitigating or responding to the risks of deciding not to meet other expectations.

Product Expectations and Outcome Expectations

The relationship between software products and business outcomes is often complex or tenuous, depending on a number of organizational factors. Few sponsors or users can envision the bridge between technology and big-picture business outcome, no matter how well they learn to match technology to immediate functionality. As the manager of an agile project, you can either evaluate these product-outcome relationships overtly early in the project, or you can rely on users and the project sponsor to keep the project on track by guessing.

To control project scope, every product expectation must support a committed business outcome expectation. In any product-centric methodology such as XP, every story and every product feature must be checked against the committed business outcome expectations. When they do not match, either the product feature (or business function) is out of scope and irrelevant, or the committed expectation is out of sync with reality.

Conduct Expectations

Conduct expectations are what the stakeholders expect to experience as part of the project. For agile development, the participation of the project sponsor, for example, can be extremely important. However, if the project sponsor expects to take a hands-off or infrequent visitor approach to overseeing the project, methodologies such as XP cannot be used effectively.

Conduct expectations include executives' perception of how predictable the completion of the project will be, and how much control and/or documentation the project will produce. In Isabella's story at the beginning of this article, she expects more documentation and supporting data for the project manager's predictions. If the project team had identified her expectations about the project's conduct as well as the project sponsor's and the project team's expectations, Ravi would not have been taken by surprise. Isabella would have had a chance to negotiate some compromise reporting.

As with product expectations, project conduct expectations must support the committed business outcome expectations. For example, if Isabella's concern for meeting the deadline is tied to a crucial business outcome, then expanding the

usual project management activities for an XP project could have been justifiable.

Conclusion

Agile software development challenges software managers and sponsors to give up their reliance on comprehensive documentation and intermediate work products. However, without boundaries, the iterative definition and development of product functionality can range out of control. By committing to compatible and feasible business outcome expectations, development teams can manage the scope

of agile projects successfully – to the satisfaction of sponsors, executives, and users. ♦

References

1. Cockburn, Alistair. Agile Software Development. Addison-Wesley, 2002: 213.
2. Cockburn, Alistair. Agile Software Development. Addison-Wesley, 2002: 177.
3. Gause, Donald, and Gerald Weinberg. Exploring Requirements. Dorset House, 1989: xvi.

About the Author



Diana Mekelburg, Project Management Professional and Certified Software Quality Engineer, is an information systems management consultant, trainer, and coach. She has taught project/program management classes to more than 500 people. Mekelburg has managed software development in a variety of organizations, including information technology, contract,

and commercial during three decades of software development, from mainframe to e-commerce. She has participated as an assessor on two large-scale Software Engineering Institute Capability Maturity Model® (CMM®) assessments and has taught CMM classes to more than 200 people.

Extreme Project Management
Phone: (713) 385-1118
E-mail: dmek@extreme-pm.com



JOVIAL GOT YOU PUZZLED?

STSC JOVIAL Services Can Help You Put the Pieces Together With:

- SPARC Hosted-MIPS R4000 Targeted JOVIAL Compiler
- Windows 95/98/ME/NT (WinX) Compiler
- SPARC Hosted-PowerPC Targeted JOVIAL Compiler
- 1750A JOVIAL ITS Products
- Computer-Based Training
- Online Support
- Use of Licensed Software for Qualified Users

Our services are free to members of the Department of the Defense and all supporting contractors.

Just give us a call.

If you have any questions, or require more information, please contact the Software Technology Support Center.



JOVIAL Program Office

Kasey Thompson, Program Manager • 801 775 5732 • DSN 775 5732
 Dave Berg, Deputy Program Manager • 801 777 4396 • DSN 777 4396
 Fax 801 777 8069 • DSN 777 8069 • Web Site www.jovial.hill.af.mil





A More Perfect Union

Throughout time people have joined together to improve their social, economic, and political condition. In theory, two people possess more potential than one and three more than two. Yet, in practice turning group potential into action has flummoxed the best of humankind.

It is not that we haven't tried. From anarchy to tyranny a plethora of "isms" and "ocracies" are available, including monarchy, fascism, communism, Marxism, socialism, theocracy, republicanism, federalism, confederacy, democracy, capitalism, syndicalism and mobocracy. While typically associated with political or economic environments these theories, or a commixture, are at the root of most human organizations, including yours.

History has found the two ends of this continuum, anarchy and tyranny, undesirable and damaging. Common sense would dictate that the answer to organizing, motivating, and governing groups of people lies somewhere between these two extremes, but where? What is the right balance between centralization and distribution, management and empowerment, or control and freedom?

In 1787 our founding fathers struggled with this issue at the Constitutional Convention. The New Jersey plan proposed sovereign power be divided between a central authority and constituent states that retain residual powers. The Virginia plan proposed creating a strong national government. Hamilton's plan proposed to return to a British-like government ... "something not very remote from that which they have lately quitted." The founding fathers had to find a compromise that was right for our nation.

Whether your organization is large or small, public or private, and whether you are an owner, manager, or employee you too must come to some compromise on how it will be governed. Those decisions will make the difference between success and failure. What can you do to favor success?

Let's look at my favorite pet peeve: cars at a traffic light. Assume that the average car is fifteen feet long, each spaced three feet apart, each traveling

an average speed of ten miles per hour, and an average green light of fifteen seconds plus five more seconds for the yellow light. A simple time-distance equation would tell you that sixteen cars can go through the intersection before the red light turns on. In real life you are lucky if ten make it. Why?

Drivers at a traffic light, like many organizations, lack leadership, communication and motivation.

LEADERSHIP: Many organizations identify project leaders but have no one to lead their projects. I'll leave all the intricate details of effective leadership to the specialist and just offer one simple but effective piece of advice: Show up! I'm amazed at how many managers insulate themselves from actual work and employees. Managers are hired to direct and motivate employees. That's hard to do when locked up in an office or a string of endless meetings.

When nominated for the presidency of the Constitutional Convention, George Washington was despondent over the loss of a brother, suffering from rheumatism, absorbed in the management of Mount Vernon, and had serious doubts about the convention. Yet he made the trip and accepted the nomination. For many of the delegates assembled, the general's mere presence gave an air of importance and legitimacy to the convention. The least a leader can do is become involved, walk the floor, or cubicles as it may be, and lend a sense of importance and legitimacy to employees and their projects.

COMMUNICATION: The next time you are waiting at a traffic light count how many seconds expire between the green light and your car's actual movement. The farther back you are the longer the time. The majority of this lag is due to the lack of communication. If the importance to get several cars through the intersection was well known and each driver was in communication with the other drivers, three or four more cars would easily get through the intersection. It works with Air Force Thunderbirds, why not with Ford Thunderbirds?

Of course your organizational goals are clear, and communication between

team players precise, right? What kind of communication dominates your project? In my experience, most managers are promoted because of their ingenuity and skill and think those same attributes will lead to successful management.

Naturally they try to solve all the problems and dole out the answers. Managers need to be less ingenious and more ingenuous. Drop the "i" and add the "u." Managers no longer develop solutions; they guide and support those who do.

When delegates decided to hold the Constitutional Convention in secret, Thomas Jefferson lamented in a letter to John Adams, "I'm sorry they began their deliberations by so abominable a precedent as that of tying up the tongues of their members." To ensure collaboration, be open, straightforward, and sincere with information, sanction, and direction.

MOTIVATION: A final reason for poor traffic light throughput is the fact that most drivers couldn't care less how many cars get through the intersection. Motivating groups to work well together to achieve a common goal, while critical, can be counterintuitive. Most people are motivated when you give them control over their work, yet giving up control leaves managers feeling undisciplined and vulnerable.

Benjamin Zander, conductor of the Boston Philharmonic Orchestra, offers his insight into the skill of motivation, "I try never to forget that the conductor is silent and the music is made by the players. The conductor's job is to awaken possibility in others. My only power is that which comes from making others powerful."

While I doubt I'll make a dent in traffic light throughput, I hope owners, managers, and employees will evaluate their responsibility in driving their organization towards a more perfect union.

— Gary Petersen
Shim Enterprise, Inc.

STC 2003

The Fifteenth Annual

Software Technology Conference

28 April - 1 May 2003 • Salt Lake City, UT

Strategies & Technologies: Enabling
Capability-Based Transformation



Participate in the premier
software technology
conference - endorsed
by the Department of
Defense

Opening General Session

Tom P. Szabig
Assistant Secretary of Defense
(Command, Control,
Communications, and Intelligence)

Speaker Luncheons

Stephen J. Mellor
Founder and Vice President,
Project Technology, Inc.

Dr. David A. Cook
Principal Engineering Consultant,
Shim Enterprise, Inc./STSC

Tom Burbage
Executive Vice President and
General Manager,
Joint Strike Fighter - Lockheed Martin
Aeronautics Co.

Industry Plenary Speaker

David W. Carey
Vice President, Information Assurance,
Oracle Government, Education &
Healthcare

Academia Plenary Speaker

Dr. William E. Halal
Professor of Management,
George Washington University

Closing General Session

Tim Border & Doug Nielsen
Motivational Speakers,
Self Management Systems

For full conference and trade show
information visit our Web site
or call today!

www.stc-online.org

800-538-2663

Source Code: CT7



Sponsored by the
Computer Resources
Support Improvement
Program (CRSIP)



Published by the
Software Technology
Support Center (STSC)

CrossTalk / MASE

6022 Fir Ave.
Bldg. 1238
Hill AFB, UT 84056-5820

PRSRST STD
U.S. POSTAGE PAID
Albuquerque, NM
Permit 737