The War is Never Easy

Anntoinette C. Gurvin
General Dynamics Information Systems

ABSTRACT

The paper describes how an organization plans and executes training to ensure the best possible use of resources to meet organizational objectives. There is a discussion of common problems, and measures to avoid or correct those problems. While the focus of the paper is achieving compliance with the CMM Training KPA, its content is relevant to any organization engaged in training technical personnel.

1. INTRODUCTION

In his book *The Greatest Generation* Tom Brokaw tells the stories of the men and women who came of age during the Great Depression and the Second World War. [1] A common purpose and common values — duty, honor, economy, courage, service, love of family and country, and, above all, responsibility for oneself, united this generation, according to Brokaw. I was raised by members of this generation and throughout my youth their battle cry was, “If the training is hard, the war is easy.” It was a useful aphorism for deflecting a child’s complaints, but my own life experience has taught me that the war is never easy, and the real goal of training hard is not to lose the war.

Any software organization that relies upon the abilities and skills of their technical personnel to remain profitable and competitive must undertake a systematic effort to ensure that the necessary skills are acquired, augmented and honed. Defense contractors have the additional challenge of meeting the requirements of the Capability Maturity Model (CMM). Almost every Key Process Area (KPA) contains requirements for training, and training itself is a Level 3 KPA.

2. COMMON TRAINING CHALLENGES

Many software-engineering organizations share some or all of the challenges listed below to maintaining their technical competence. In the next section, discussion will focus on the key elements of professional development planning that mitigate these challenges.

2.1. Budget Limitations

Underlying all other challenges is the sheer expense of maintaining technical competence for software staff. No budget is ever enough to execute every activity that an organization identifies as a training requirement. Technical training tends to be very expensive, usually in the range of $200–$300 per employee per day, not including labor. Resources are never sufficient to execute every training task that an organization identifies in the initial planning phase.

2.2. Breadth of Requirements

The breadth of training required is overwhelming. The tools and the technologies to which they apply are nearly numberless and extremely volatile. Environment elements are added, deleted and changed with increasing frequency. The organization must train their engineers in the processes used to execute development efforts. When the annual budget is prepared there are many unknowns about what additional requirements may surface in the coming year.

2.3. Business Relevance

Technical training must be relevant to the business. Potential new business often requires training when it materializes, but the expense cannot be justified until the business is assured. After the new business is won, the training becomes reactive – the challenge then is to be sure that it occurs in time to benefit the engineers but not so early that it can not be applied while it is still fresh.

2.4. Organizational Effectiveness

Engineering organizations often overlook or under-prioritize the need for training in 'soft skills' such as interpersonal communications, effective meetings, leadership or team building, because of intense focus on technical competitiveness. The complexity of modern software systems dictates development by teams with layers of technical and project leadership and complicated interfaces with customers and other elements of the organization.

2.5. Lack of Commitment

Many organizations struggle with employees failing to attend mandatory training because they do not view it as relevant to their position. This is especially common when the training is related to processes or 'soft skills.'

3. CREATING THE ORGANIZATION'S TRAINING PROGRAM

3.1. Developing a Professional Development Philosophy

The Webster's Dictionary definition of training is:

Function: *noun*
Date: 1548
1 a : the act, process, or method of one that trains **b :** the skill, knowledge, or experience acquired by one that trains
2 : the state of being trained [2]

There are a wide variety of acts, processes or methods available and an infinite number of skills, knowledge or experiences possible to achieve the state of being trained.

Employees who feel 'entitled' to receive any training they desire will be disappointed. Their expectations will always exceed the company resources available to meet those expectations. Underlying all the tasks necessary to execute an organization's training plan, is the need for a clear statement of the mission and objectives of the company's resources to be applied to training.

It may not be possible to overcome the entitlement mentality, but clarity in communicating the mission and objectives avoids creating unreasonable expectations. We may refer to the company's 'training plan,' but the result is the employees' 'professional development,' and the semantics are important.

Enabling employees to acquire new skills benefits both the company and the employee, but the company will benefit only as long as it retains the employee. The employees continue to benefit when they change employers. In fact, the professional skills acquired at the expense of their current employer often make them more attractive to other employers. Thus it only makes sense that the professional development responsibility is shared between employers and employees. Employees must be willing to contribute to their professional development.

This philosophy may be implemented in many ways. What is important is that it is understood by all involved. Examples of shared responsibilities in our organization include:

a. We do not authorize employees to charge to the training overhead budget to read a book or other technical publication. We do, however, buy the books.

b. We will pay the fees associated with technical classes, such as achieving Microsoft certification. The employees do the class work on their own time.

c. The software-engineering department normally does not approve conference attendance unless the employee publishes and presents a paper at the conference. Then we sweeten the deal with a nice check after publication.

Through careful, imaginative planning and commitment to professional development by both the employer and the employee, resources can deliver a great deal of benefit to both parties.

3.2. Creating an Organizational Infrastructure

There are two engineering levels that comprise our training infrastructure.

In 1998 the Vice President of Engineering made it known that he wanted engineering departments to coordinate training objectives through an ‘Engineering Learning Team.’ The members were the managers responsible for training within their departments. Our first task was to document the engineering training process. Then the team evaluated the various Lotus Notes databases that the departments used to support their training activities. These separate departmental databases were retired in favor of a single engineering training database. The Learning Team was responsible for evaluating the training goals in the 1999 Engineering Functional Plan and measuring our progress towards those goals. The Learning Team provided not only support to engineering department managers, but visible proof of engineering commitment to learning.

The second level of infrastructure is the documented software engineering training process, also owned by the training manager. The elements of this process are described in the next section. Software engineering managers all have focus areas (software process improvement, staffing and training) in additional to their common management responsibilities. I developed the 1999 software engineering training plan in the role of training manager. My colleagues in Software Functional Management supported all my activities. This support included reviewing the draft training plans and budgets, training requests and proposed responses, training calendars, and various training proposals during weekly staff meetings.

3.3. Creating the Training Plan

3.3.1. Gathering Requirements

Like every engineering activity, planning for staff professional development begins with documenting the requirements. Each year offers new challenges to every organization. Each organization has a unique way of documenting their goals and plans, but most organizations produce the same types of output during their annual planning process. The following documents and activities are specific to one organization, but should be easily mapped to any organization.

Strategic business plans are formulated at the executive level and usually project four or five years into the future. In our organization, they identify the business pursuits upon which we will focus in the coming years. This information is input to both our staffing and training functions, as the business pursuits are usually associated with specific technologies. For example, fibre channel technology may be prevalent in a business unit’s pursuit list.

Department plans such as the Engineering Functional Plan (EFP), the Software Process Improvement Plan, and the Technology Insertion Plan are prepared at lower levels of the organization, and are usually applicable only to the coming year. As such, they tend to contain much more specific requirements. The EFP addresses engineering-wide initiatives. For example, engineering leadership might decide that all members of the department should receive training in conducting effective meetings.

The software process improvement plan identifies process initiatives with reference to the organization’s Capability Maturity Model (CMM) goals. Almost all Key Process Areas (KPA), regardless of level, require process training to satisfy an ‘Ability’ requirement of that KPA. Readers interested in CMM compliance should refer to the compliance matrix in Appendix A.

There may be additional requirements relative to methodologies adopted to support processes. An example would be identifying Object Oriented Design (OOD) as the organization’s design methodology in support of the design process. A related plan, the Technology Insertion Plan, identifies tools that the organization intends to purchase in support of processes and methodologies. To continue the previous example, the plan might indicate the intention to implement OOD via Rational Rose™. In this example these plans signal the necessity of having staff capable in OOD and able to execute the process using Rational Rose™.

Individual programs create development training plans that must be reviewed for program specific requirements. As an example, a contract may require that the program be executed using Integrated Product Development (IPD). The engineers assigned to this program may require IPD training in order to execute the contract successfully.

A final source of training requirements is the employee's personal development files. Most organizations review employee performance at least annually. In the course of these evaluations they usually identify development needs for the employees. Advanced education is the most common type of individual professional development activity and is usually funded through some type of tuition assistance program, managed by the organization, but budgeted by department. Reviews should also be analyzed for trends, such as a number of employees who need to develop interpersonal skills. If the group is of sufficient size, the organization might decide to schedule this training. Individual managers are responsible for following up with employees to ensure that they are pursuing development objectives. They have an important role in encouraging the employees to take responsibility for their own professional development using the organization's training assets as an enabler.

As a follow up to the entire requirements analysis process, we request input from all software engineers. We ask them to forecast their tuition reimbursement requirements and make suggestions for what should be considered for inclusion in the training plan.

3.3.2. Setting Training Plan Objectives

After requirements have been identified, the next step in planning is to determine the priority assigned to each class of requirements. Our 1999 plan had the following objectives:

The following elements are inputs to the software engineering training plan and form the objectives as described. They are listed in the order in which they are prioritized for 1999.

a. The Software Process Initiative (SPI) Plan focuses on attaining, at a minimum, a CMM rating of Level 3 in 1999. The highest priority is overcoming deficiencies identified in the 1998 evaluation, which requires training, especially in the first half of 1999. Teamwork training is an over-riding requirement.

b. GDIS functional plans contain requirements derived from programs, including internal research and development (IRAD). This year, the EFP also has the following goal: "Increase the current level of engineering employees involved in a continual learning activity by 20%."

c. Programs require training in technologies to expand our base of expertise, specifically, in fibre channel and VxWorks.

d. Individual development needs, as documented in our Professional Development and Feedback process, indicate that a significant number of software engineers require training in interpersonal communications.

e. Support for the Object Management Group (OMG) CORBA Real-time SIG Co-chair is funded by the SWE training budget. This includes membership, conference fees, labor and all travel expenses. Our level of participation is being reviewed during 1999 and we are seeking co-sponsors.

3.3.3. Establishing and Balancing the Budget

Usually the dollar amount allocated for training is a known quantity well before any other training plan details are determined. The planners must try to put that number out of their minds and execute the following steps in order.

a. List all the potential activities that were identified in the requirements analysis phase by category such as ‘Technology Related’ or ‘Process Related’.

b. Assign a cost to each activity. In our organization budget is allocated in five categories: Training Fees, Conferences, Training Travel, Training Labor, and Tuition Reimbursement. The list of activities is reproduced for each of the first four categories, and costs for a single activity may be assigned to more than one sheet. For example, we may forecast a vendor fee to conduct a Team Building seminar on the training fees worksheet. The labor for the employees will be forecast on the same line on the training labor worksheet. This is a good method because it gives reviewers more visibility into exactly how the budget has been allocated.

c. Determine, based upon requirements analysis, the relative priority of all entries. Rearrange the lists from highest to lowest priority.

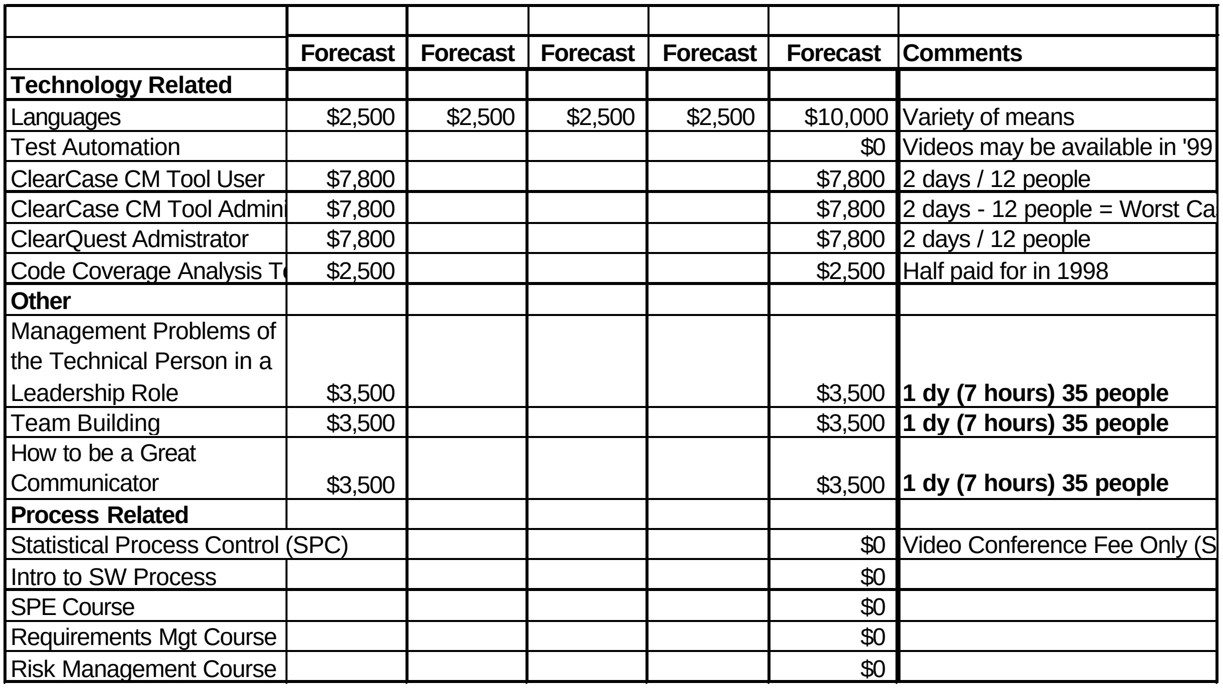
e. Now we are allowed to consider the budget number. Compare the forecast for all activities to the budget figure. Begin removing activities from the bottom of the list until the forecast and available budget balance. Do not delete these activities. Create a new table below the budget table so visibility into the decision process is retained. As time passes, budgeted activities may be overcome by events or budget may be increased. Then these activities may be promoted into the forecast.

f. As with any budget, it is wise to assign a portion to reserve.

g. The budget is reviewed internally before undergoing review at the engineering level.

h. The approved training plan, in our case the Excel Workbook supplemented by a text description of the output of the entire planning process, should be archived in a common area, accessible to all employees.

Figure 1 is an example of a Training Workbook Worksheet.



	Forecast	**Forecast**	**Forecast**	**Forecast**	**Forecast**	**Comments**
Technology Related						
Languages	$2,500	$2,500	$2,500	$2,500	$10,000	Variety of means
Test Automation					$0	Videos may be available in '99
ClearCase CM Tool User	$7,800				$7,800	2 days / 12 people
ClearCase CM Tool Admini	$7,800				$7,800	2 days - 12 people = Worst Ca
ClearQuest Admistrator	$7,800				$7,800	2 days / 12 people
Code Coverage Analysis T	$2,500				$2,500	Half paid for in 1998
Other						
Management Problems of the Technical Person in a Leadership Role	$3,500				$3,500	**1 dy (7 hours) 35 people**
Team Building	$3,500				$3,500	**1 dy (7 hours) 35 people**
How to be a Great Communicator	$3,500				$3,500	**1 dy (7 hours) 35 people**
Process Related						
Statistical Process Control (SPC)					$0	Video Conference Fee Only (S
Intro to SW Process					$0	
SPE Course					$0	
Requirements Mgt Course					$0	
Risk Management Course					$0	

Figure 1. Training Plan Workbook – Fee Spreadsheet Example

3.4. Executing the Training Plan

A successful training program relies upon good communication throughout the year. [3]

3.4.1. Planning/Scheduling Events

The training manager is responsible for scheduling training events, dealing with vendors, determining who will attend, reserving facility space, and notifying attendees. We have a bulletin board in our main hallway dedicated to software training events. The focal point of this board is the three-month training calendar. All planned events are visible to all employees, giving them the opportunity to request that they be authorized to attend events.

Reminders are sent out periodically as the scheduled training draws closer. For required training, the invitation is frequently reiterated by the Vice President of Engineering, stating his commitment to ensuring employees receive mandatory training, and offering to negotiate schedule conflicts with programs. This has been an effective means of keeping attendance high and reinforcing the organization's commitment to training.

As the training manager, I subscribe to email services for our vendors, such as Microsoft MSDN. There are numerous cost-free seminars and technical briefings that we make available to employees. These events have been very popular with software engineers and cost little or nothing.

3.4.2. Employing a Feedback Mechanism

Feedback is an essential tool for managing and improving the training program. Participants evaluate each activity funded from the training budget. This data is analyzed to determine how effective that activity was. The data affects future planning. For example, in the past we have used outside vendors to deliver C++ training (a five-day event). This costs approximately $1,200 per student. The feedback stressed how much our engineers value the opportunity to expand their skill set, but they felt too much was covered too quickly with too few opportunities to practice. In investigating the outcomes we discovered that there was actually very little benefit unless the engineer put this training to work immediately. In practice, that happens only infrequently. The decision was made to use other, more traditional means of learning to add these skills, relying upon the employees to be motivated to share responsibility for their professional development.

During the planning phase, we request employee feedback two times. Early in the third quarter we solicit feedback on training activities for the current year. This feedback is used to improve the training process itself, measure satisfaction, identify employee expectations, and rate events and vendors. A bit later we solicit recommendations for additions to the plan under development for the coming year.

3.4.3. Communicating with Your Target Organization

There are several other routine activities designed to foster good communication about training between software engineering management and the target organization.

We use the regularly scheduled Employee Communications Meetings to disseminate information about the training program. In January we review the approved plan and discuss the results of the annual survey. It is very important to deal with the expectations surfaced by the survey. Employees need a clear understanding of what resources are available, how they will be used, and why these decisions were made. Although it is never possible to meet all expectations, it is important to acknowledge them and put them in the context of the organization's goals.

During subsequent Communications Meetings, we discuss the status of training activities and budget and review metrics.

The training manager also prepares a column for each issue of our Software Engineering Newsletter. We make all training plans and documentation readily available on Lotus Notes.

One of the most important communication tasks is ensuring two-way communication. We make it a point to respond to all feedback, letting the employees know what actions we have taken to address their feedback. Knowing that we listen and take action based on their feedback has encouraged employees to improve the quantity and quality of their feedback.

3.4.4. Updating the Plan

At least monthly the plan is updated, at a minimum to record budget data for activities that are completed. Other revisions may occur due to changes in requirements. These revisions are handled in the same way as the original planning.

Following audits or independent evaluations, it may be necessary to revise the plan to address findings.

Finally, we have an Engineering Training Request database, so employees may request consideration for training opportunities not in the plan. These activities are responsible for expending most of the reserve budget.

3.4.5. Metrics

As discussed in previous sections, we set goals at every level of the organization and we produce metrics to measure our progress against the goals.

An example of the ongoing assessment of internally developed and delivered courses is Figure 2. The x-axis refers to questions on the feedback form. The y-axis measures the responses.

The Engineering Learning Team tracks progress against goals in the Engineering Functional Plan. Figure 3 is a metric used to assess the hours of training reported as indirect charges versus the hours of training recorded in the Human Resources Training database. Figure 4 tracks engineering progress in increasing involvement in training activities over the previous year.

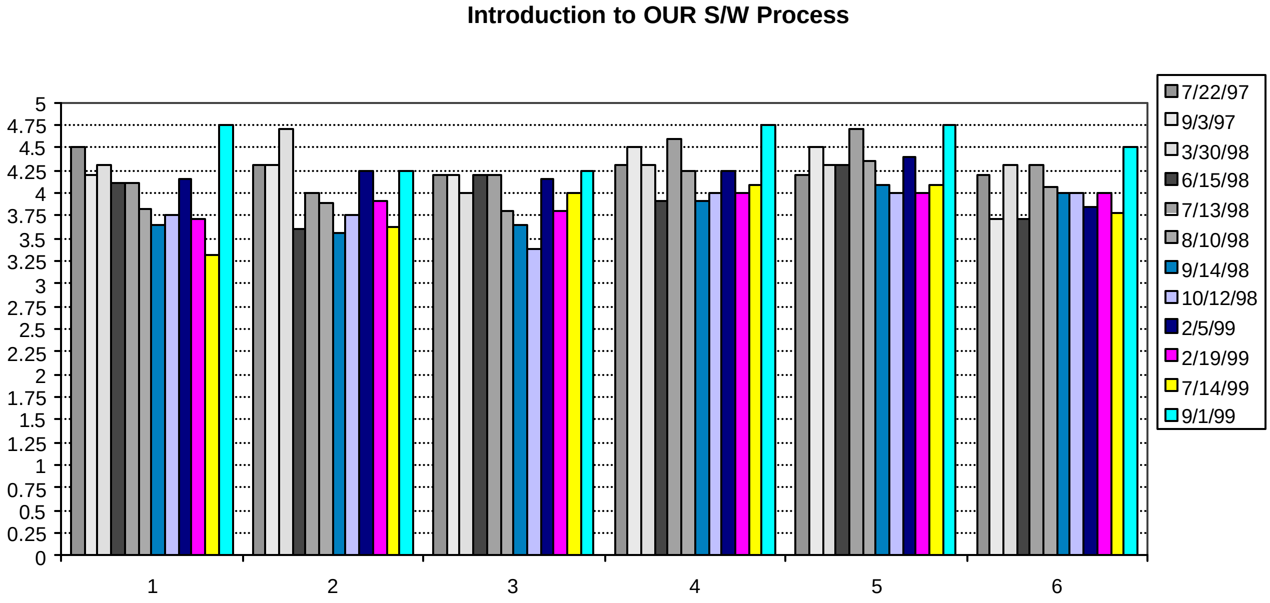
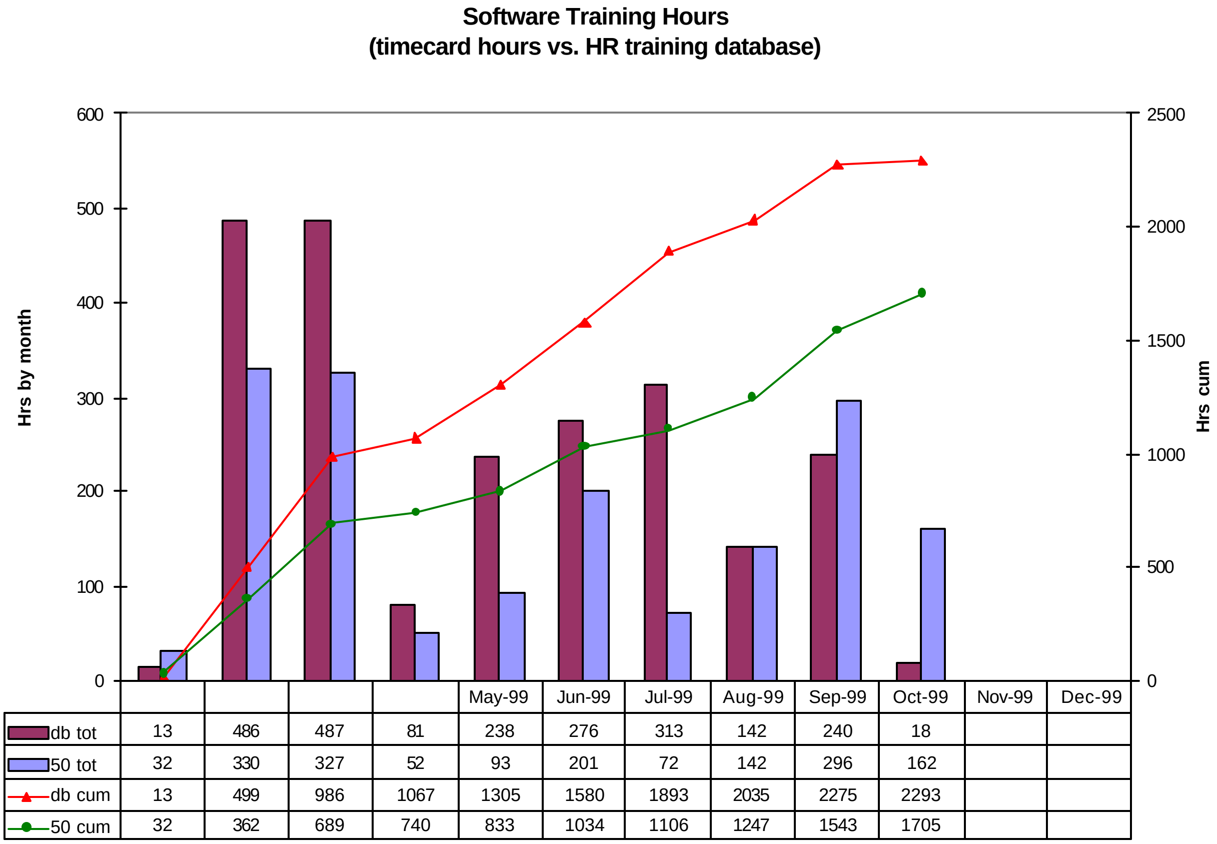
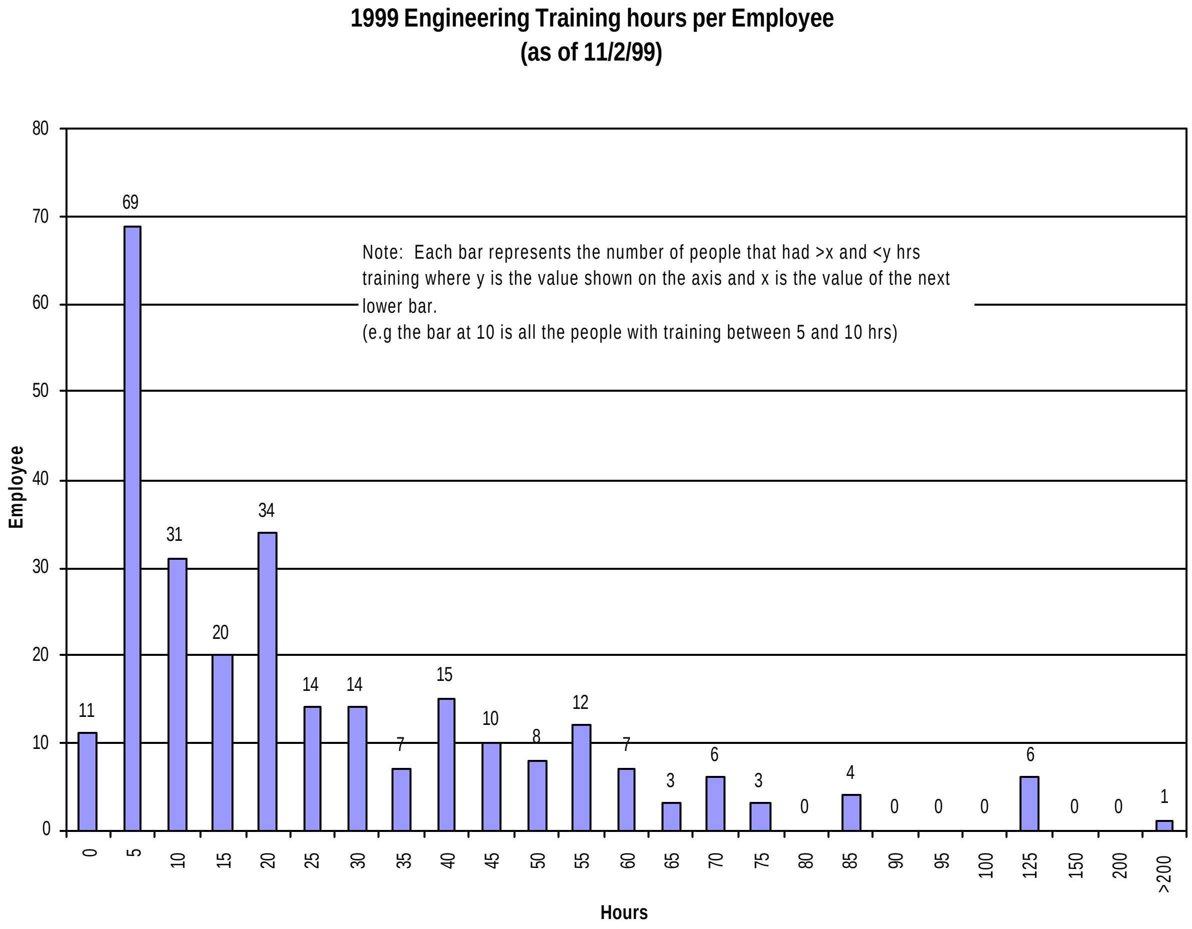


Figure 2. Cumulative Feedback Results for Internally Delivered Training.



					May-99	Jun-99	Jul-99	Aug-99	Sep-99	Oct-99	Nov-99	Dec-99
▬db tot	13	486	487	81	238	276	313	142	240	18		
▬50 tot	32	330	327	52	93	201	72	142	296	162		
—▲—db cum	13	499	986	1067	1305	1580	1893	2035	2275	2293		
—●—50 cum	32	362	689	740	833	1034	1106	1247	1543	1705		

Figure 3. Engineering Learning Team Metric for Software Engineering Department



Note: Each bar represents the number of people that had >x and <y hrs training where y is the value shown on the axis and x is the value of the next lower bar.
(e.g the bar at 10 is all the people with training between 5 and 10 hrs)

Figure 4. Engineering Training Span

4. And Now, Back to the War…

In February of 1999, at the Software Management Conference during one of the general sessions, Tom DeMarco commented on training. He said (paraphrased), “Training is allowing the inexperienced person to do slowly what the experienced person does quickly.” This comment resonated for me throughout the entire year. The money, the planning, the communication involved in delivering training to a technical organization are all directed at arming employees for the competition in the business theatre of operations. And these things are necessary. But I have not forgotten that they are merely the prelude to empowering the inexperienced to master new technology, to practice until they are perfect.

Appendix A. Mapping The Training Program to the CMM

This appendix is applicable to those organizations seeking CMM compliance. Planning and executing training as described in the preceding paragraphs and creating the necessary process documentation described in Table I, the CMM Level 3 Training KPA Compliance Matrix, will ensure satisfaction of the requirements of the Training KPA.

The goals of the Training Key Process Area referred to in the 'Goal' column are:

1. Training activities are planned.
2. Training for developing the skills and knowledge to perform software management and technical roles is provided.
3. Individuals in the software engineering group and software related groups receive the training necessary to perform their roles.

Table II. CMM Level 3 Training KPA Compliance Matrix [4] [5]

Key Practice/Description		Goal	Documentation		
			GDIS	**Project**	**SWE Process**
C0-1	The organization follows a written policy for meeting its training needs.	1,2,3	Policy		
AB-1	A group responsible for fulfilling the training needs of the organization exists.	1,2,3			1) Software Training Procedure specifies SW functional manager responsibility for SW staff training. Training is an agenda item at weekly management staff meetings.
AB-2	Adequate resources and funding are provided for implementing the training program.	1,2,3			1) Software Training Procedure documents the requirement to assign a training manager. 2) Software Engineering Training Plan documents the plans and budget.
AB-3	Members of the training group have the necessary skills and knowledge to perform their training activities.	1,2,3			1) Software Training Development Procedure documents the process to “train the trainers”. 2) Software Process Lotus Notes Database contains training and certification records for trainers.
AB-4	Software managers receive orientation on the training program.	1,2,3			Software Process Improvement Procedure describes bi-monthly SEPG meetings where training is a standing agenda item and SW Managers are members of the SEPG.
AC-1	Each software project develops and maintains a training plan that specifies its training needs	1	HR Training Database	Project Training Plan	1) A documented project training plan is required 2) Software Training Procedure defines software engineering roles, required training for each role, and the process used to document project training status.

Table II. CMM Level 3 Training KPA Compliance Matrix [4] [5]

Key Practice/Description		Goal	Documentation		
			GDIS	Project	SWE Process
AC-2	The organization's training plan is developed and revised according to a documented procedure	1	Procedure and Training Plan		Software Training Procedure documents the processes for development and maintenance of the training plan.
AC-3	The training for the organization is performed in accordance with the organization's training plan	1,2			1) Software process documents the process and the training materials. 2) Software engineering training is updated monthly with actuals.
AC-4	Training courses prepared at the organization level are developed and maintained according to organization standards	2	Standards for Internal Training Courses		1) Software Training procedure documents the development process for internal training courses, and contains the Software Training Architecture. 2) Software process documents maintenance of training materials in the software asset library.
AC-5	A waiver procedure for required training is established and used to determine whether individuals already possess the knowledge and skills required to perform in their designated roles	3			1) Training procedure documents the process for waiving process or technical training required for a specific SWE role. 2) Training Waivers are maintained in the Software Process Asset Library.
AC-6	Records of training are maintained.	3	HR Training Records		
ME-1	Measurements are made and used to determine the status of the training program activities.	1,2,3	Status Reports, Engineering Metrics		1) Training budget is updated monthly and budget metrics prepared. 2) Measurements of Engineering Training status are compiled and reviewed by Engineering Management.
ME-2	Measurements are made and used to determine the quality of the training program.	2,3			Training evaluations and feedback form results are measured and tracked for all classes.
VE-1	The training program activities are reviewed with senior management on a periodic basis.	1,2	Status Reports To Senior Mgmt.		Project Status Reports include training and are reviewed at monthly by senior management.

Table II. CMM Level 3 Training KPA Compliance Matrix [4] [5]

Key Practice/Description		Goal	Documentation		
			GDIS	Project	SWE Process
VE-2	The training program is independently evaluated on a periodic basis for consistency with, and relevance to, the organization’s needs.	2,3			1) Software training procedure documents the requirement to perform an annual independent evaluation of the training plan. 2) Results of this evaluation are maintained in the software process database.
VE-3	The training program activities and work products are reviewed and/or audited and the results are reported.	1,2,3	Audit Reports		1) Monthly SQA Discovery Report metrics are collected and distributed, including the SEPG and training program audit details. 2) SQA audits all projects, including the SEPG of which the training program is part.

BIBLIOGRAPHY

[1] Brokaw, Tom, *The Greatest Generation*, (New York: Random House, 1998).

[2] WWWebster Dictionary, http://www.m-w.com/cgi-bin/dictionary

[3] Joiner, Brian L, *Fourth Generation Management, (*R. R. Donnelly & Sons Company, 1994)

[4] Capability Maturity Model for Software, (Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 1993)

[5] Humphrey, Watts S., *The Capability Maturity Model*, *Guidelines for Improving the Software Process*, (Addison-Wesley Publishing Company, 1995)

Anntoinette Gurvin

Anntoinette (Toni) Gurvin received her bachelor's degree in computer science from the University of Maryland in 1987. She also has bachelor of arts degrees in English and communications and is working on a master's of science in software systems. Toni has over twenty years of experience, predominantly in military systems development, encompassing software development, software test, software quality engineering, process improvement, and project engineering. Her professional experiences have led her to focus on process as the key to success in any endeavor. Toni is the manager of tactical avionics software engineering for General Dynamics Information Systems in Minneapolis, Minnesota. She also manages the training program for software engineering.