DOCUMENT RESUME

ED 368 262 HE 027 278

TITLE	Managing Information Technology as a Catalyst of Change. Track III: The Impact of Quality.
INSTITUTION	CAUSE, Boulder, Colo.
PUB DATE	94
NOTE	78p.; In: Managing Information Technology as a Catalyst of Change. Proceedings of the CAUSE Annual Conference (San Diego, CA, December 7-10, 1993); see HE 027 275.
AVAILABLE FROM	CAUSE Exchange Library, 4840 Pearl East Circle, Suite 302E, Boulder, CO 80303 (individual papers available to CAUSE members at cost of reproduction).
PUB TYPE	Speeches/Conference Papers (150) -- Reports - Descriptive (141)
EDRS PRICE	MF01/PC04 Plus Postage.
DESCRIPTORS	Budgeting; *Change Strategies; Computer Software; Decision Making; Educational Change; Evaluation Methods; Higher Education; *Information Management; *Information Systems; *Information Technology; Leadership; *Quality Control; Strategic Planning; *Total Quality Management; Users (Information)
IDENTIFIERS	*CAUSE National Conference; Harvard University MA; Tufts University MA; University of Connecticut; University of Kansas

ABSTRACT

The 1993 CAUSE Conference included eight papers on the adoption of Total Quality Management (TQM), in its various forms, by information technology (IT) sections of colleges and universities. Papers have the following titles and authors: (1) "The Impact of TQM on an IT Organization: The First Eighteen Months" (Paul M. Morris), which outlines the experience of working with TQM at Tufts University (Massachusetts); (2) "Strategic Planning and Budgeting for Information Technology" (Charles R. Thomas and Dennis P. Jones); (3) "Implementing a New System on Time in Bad Times" (Elaine David), which describes the University of Connecticut's installation of a touch-tone registration system and improvement of other computer systems; (4) "Quality Software...but by Whose Definition? Is the End-User King?" (Louise M. Schulden); (5) "Guerrilla TQM or How To Infiltrate TQM into Your Institution" (Deborah J. Teeter and Jan Weller), which describes the implementation of TQM by the University of Kansas Department of Telecommunications; (6) "Change in the Trenches: Continuous Improvement of Service Processes" (Connie Towler and Douglas Renick), which explores service process improvement at Harvard University (Massachusetts); (7) "Establishing Trust and Building Relationships: Negotiating with Information Technology" (Scott C. Ratzan), which describes the COAST (Communication, Options, Alternatives, Standards, Trust) Model of Negotiation; and (8) "Assessing the Effectiveness of Information Technology" (Susan F. Stager and others). (Some papers contain references.) (JDD)

Managing Information Technology as a Catalyst of Change

Proceedings of the 1993 CAUSE Annual Conference

TRACK III

THE IMPACT OF QUALITY

December 7-10
Sheraton on Harbor Island
San Diego, California

TRACK III

THE IMPACT OF QUALITY

Coordinator: Constance F. Towler

Total Quality Management, in one form or another, is being adopted by many IT organizations today. The impact of this process can cause a dramatic change in the way we manage our organizations. How will we handle these changes?

The Impact of TQM on an IT Organization: The First Eighteen Months

Paul Morris
Tufts University
12/6/93

1. Institutional Background of Tufts

- Private university, research and teaching
- Decentralized: 7 Schools plus Central Administration
- 4,300 undergraduate; 3,000 graduate
- Budget: $270 million
- Central IT budget: $9.5 million, staff of 80
- Academic, MIS, data communications, telephone

2. Problems needing solution, which led us to consider new alternatives such as TQM:

- Declining real budgets
- Increasing demands for IT services
- TCCS not well perceived by users
- Staff under stress, felt unappreciated

3. My expectations of TQM as I started TCCS down that road

- Improve customer orientation
- Means of motivating staff
- Way of developing priorities
- Empowerment of staff through participation in decision-making about their jobs
- Set of tools for focusing on customer needs

4. Self-preparation before getting started

- Attended GOAL/QPC conference (12/91) (a local research, training and consulting organization)
- Read Walton's "Deming Management Method" (but did not adopt the Deming set of issues)
- Attended 3-day course at GOAL/QPC
- Attended 6-day course at CQM (Center for Quality Management: information-sharing consortium of local corporations using TQM)
- Joined CQM University Affiliates, which provided networking with other local universities and industry practioners

- (after a year) taught course in TQM

5. First TCCS (Tufts Computing & Communications Services) activities

- Started talking to TCCS managers about TQM
- Tried using some policy-level tools - failed, due to inadequate training of myself and managers. Should have started with something simpler (like KJ's)
- Focus Groups to analyze Voice of Customer (9/92)
- Used KJ analysis to identify major themes
- Analysis done by 14 TCCS managers and supervisors - to build their sense of ownership, and develop some faith in TQM tools

6. What the VOC analysis told us

- Not meeting customers' expectations for service
- TCCS does not understand customer needs
- TCCS does not understand customer environment
- Eight major themes for improvement (next topic)

7. Eight Task Forces

The mission of these are:

- Develop senior University-wide commitment
- Inform customers of services provided
- Establish feedback channels for customers
- Develop desktop support strategy
- Develop service-level agreements with customers
- Improve customers' access to transactions data
- Improve Help Desk/Customer Service Center
- Broaden skill set of staff

To illustrate the sorts of things we have been doing, I shall discuss the Desktop strategy group:

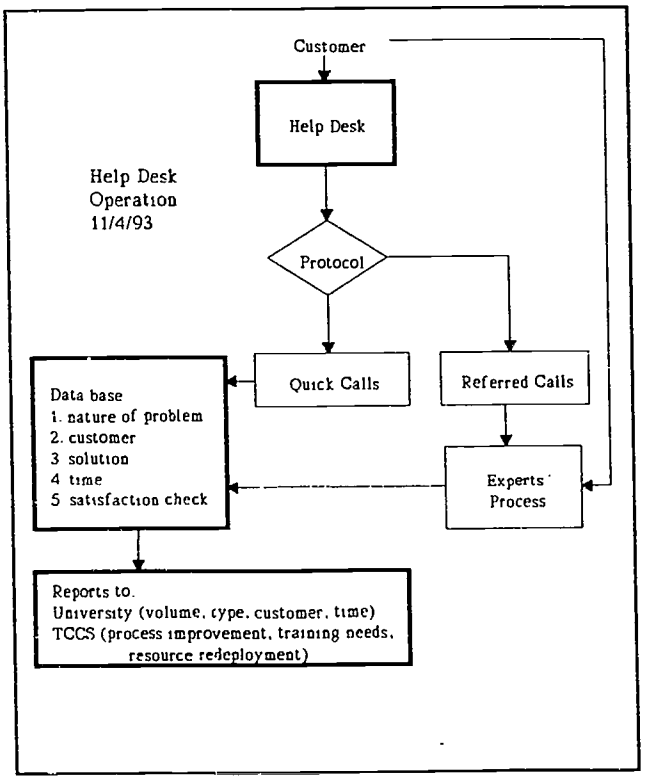
8. "Desktop Support" - the Problem

- Taskforce: 4 senior managers (this was not a multi-level TQM-style group, because the initial issues were policy-related rather than operations-related)
- Identified key customer complaints
  - who to call
  - do not like talking to a machine

- takes too many, different people to solve problem
- service unreliable, takes too long
- poor communication about what is going on

9. "Desktop Support" - Progress

- Developed process based on existing Help Desk
- Data collection
- Problem tracking
- Performance reporting



10. Help Desk

We have focussed on, and strengthened, the staffing, training and processes of the Help Desk:

- Transferred, trained staff from other areas
- Implemented Automatic Call Distribution - feature of telephone system
- System "knows" who is available (staff "log in")
- Data on volume, times, etc. automatically collected
- Made better use of existing tracking system (running on VMS now, will move to LAN)
- Monitoring and Reporting features now being used more actively
- Weekly management review of trends, unsolved problems.

11. "Desktop Support" - Issues Still to be Addressed

- Still no agreed list of "Things We Don't Do"
- Need FAQ list, solutions database as part of Help Desk resources
- Desirability of same process for all experts?
- We expect "Continuous Improvement", so we will always be looking for ways to do things better.

12. Central Administration's TQM Program

TCCS's TQM program is happening at a time when Central Administration (of which it is part) is also experimenting. Its program is named "TQ3" for its three objectives:

- Continuous improvement of services to customers
- Improve the way Central Admin operates (greater efficiency)
- Improve skill levels and job satisfaction for all levels of staff

13. TQ3 activities

The TQ3 Steering Committee has devloped a plan, and launched a number of initial activities within Central Administration:

- Senior management training
- Middle management training
- Communication via TQ3 Newsletter
- QI Team training
- QI pilot projects

14. TQ3 Pilot Projects

To test our ideas, and in particular the Seven Step Problem Solving method, a number of Pilots have been launched, one in each Division of Central Administration, and the last one in the Vet School:

- Reduce cycle time to generate P.O. from requisition
- Reduce cost of purchased vehicles by buying them used
- Reduce cycle time for hiring research assistants
- Reduce time required to answer payroll inquiries
- Reduce number of lost or incomplete records in Animal Hospital

15. Process for QI Teams

The QI teams are using the CQM methodology, assisted by Joiner's "Team Handbook":

- Seven Step Problem Solving
- currently at Steps 2 & 3: Data analysis, Causal analysis
- KJ analysis - qualitative data, focus on identifying underlying weaknesses

16. Results so far?

In TCCS:

- Customer needs now the official touchstone
- Projects, priorities try to reflect customer needs
- Help Desk project making encouraging progress
- Internal Training Committee making progress (on broadening staff's skill set)
- Management-by-fact making progress
- Still a long way to go

TQ3 Pilot Projects:

- Enthusiasm on QI Teams so far
- Results in January

Lessons Learned:
Warning: Anecdotal Evidence, sample size 1!

17. Cultural Change depends on New Processes

Talking about TQM, and asking for attitude change, will not work unless you give staff new processes to work with:

- If existing processes are not producing customer satisfaction, it really is management's fault
- Don't expect staff to provide better customer service without working with them to improve processes
- Focus on processes, not on individuals
- Listen to staff about why they are not meeting customer expectations
- Using the tools, and seeing them work, is critical to cultural change

18. Control IT Staff's Expectations

- TQM is not a panacea
- Change will happen slowly, over several years
- Stress "participation in deciding how to do your job"
- Stress **not** "participation in policy-making"
- Staff participation means some loss of control for managers
- Must tolerate other depatments who are not implementing TQM

19. Beware of Strangers

In trying to interest people in TQM, I have made the following observations:

- Very few people want to hear about the Japanese
- Japanese-style jargon offers an excuse for rejection
- Very few people want to hear about corporate successes
- Corporate-style justifications offer an excuse for rejection
- Most people want to hear that a school just like yours solved all their problems with TQM successfully, quickly and with no pain

19. A University-wide TQM Program Needs:

Based on a year's experience with the senior Central Administration managers, TQM needs:

- an agreed vision by top managers
- agreed expectations and objectives by top managers
- a link to local industrial practitioners (to be used discreetly)
- plenty of time and patience
- a variety of perspectives and expertises
- a balance of analytical tools and human relations skills

20. Difficulty of Bringing about Change

The following are obvious, but I have encountered all of them in the past year:

- Avoid unrealistic expectations on results, time, effort
- Expect progress to be S-L-O-W
- Constant re-inforcement needed
- Words, attitudes and actions must all be embody what you preach
- Not everyone will share your vision
- Not everyone will trust your motives

21. An Act of Faith: TQM is Worth Doing!

STRATEGIC PLANNING AND BUDGETING FOR INFORMATION TECHNOLOGY

prepared by:

Charles R. Thomas and Dennis P. Jones

INTRODUCTION

Strategic planning can enable an institution to take advantage of new and different opportunities in the future while minimizing the negative impact of unexpected challenges along the way. In this time of rapid technological change, strategic planning can also provide great opportunities in the use of information technology to support the mission and goals of colleges and universities. The planning effort must, however, be conducted within the framework of the institutional planning process and must consider the institutional culture, history and resources.

While many institutions engage in strategic planning activities at the campus level, few have extended those activities to the information technology units, and even fewer have linked them to budgeting and operations. The strategic planning process described in this paper is not revolutionary, in fact it has been used by dozens of institutions. The unique addition is the integration of *budgeting* at the strategic level. The purpose of this paper is to present a detailed framework for the implementation of a strategic planning and budgeting *process* for information technology that ensures policy level attention to the resources required to achieve strategic objectives. This approach involves close work with the appropriate institutional policy committee supported by staff work from the information technology unit. It is important to note that while outside assistance can bring a broad perspective and knowledgeable opinions to the process, and an outsider can serve as a catalyst to keep the process moving, the strategic planning process must be "owned" by the institution.

DIMENSIONS OF STRATEGIC PLANNING

The strategic planning part of the process described is based in part on "*Strategic Planning for Computing and Communications*[1] by Penrod and West, and generally follows the model developed by Dr. Robert Shirley[2]. The following important dimensions of planning for information technology cited by Penrod and West are based on a list compiled by John Moynihan.[3] and modified to fit the higher education environment. Planning for information technology should:

1. be a *formal continuous process*, have the support of senior administrators, use up-to-date planning methods, and result in documented output publicized to the institutional community;

2. be *eclectic*, choosing the best features from a diverse set of resources;

3. include a *review of the mission and the organization* of academic computing, administrative information systems, and telecommunications;

4. be *broad but bounded* in scope by economically and technically feasible solutions;

5. *involve* senior administrators, representatives of major client departments, and information technology staff members;

6. involve the identification of potentially *important technologogical developments* and recognize when those developments make the transition from "state of the art" to "state of the market";

7. address the *technical and managerial assets* of the information technology units through an analysis of strengths and weaknesses;

8. formalize an *organizational architecture* that addresses all departmental levels of the institution;

9. formulate an organization-wide *information architecture* on which all institutional application systems are based; and,

10. result in an organization-wide *technical architecture* that includes hardware and software platforms for voice, data, and image networks;

11. develop a collegial process for selecting an *organization-wide too! set* for both academic computing and administrative application systems development.

12. be *driven by institutional problems and opportunities* and by *client office needs* rather than by technological developments;

THE PLAN TO PLAN

Before undertaking to develop a strategic planning process for information technology, it is important to have the commitment and support of the institutional leaders. The best way to achieve this is to have a very understandable *Plan To Plan*, to communicate that plan to the appropriate individuals on the campus, and then encourage participation in the process. In the collegial environment, the involvement of the right people in the right processes at the right time can do much to ensure success.

An effective planning process should be consciously and formally organized. Both the administrators and the support staff should have formally assigned planning responsibilities[4]. To this end, a well thought out *plan to plan* can enable an institution to reach consensus on a planning process with a minimum number of false starts. In the follow paragraphs present a suggested set of activities for the plan to plan.

1. Conduct an on-campus workshop on strategic planning for top administrators and advisory committee members. The purpose is to establish a base set of knowledge about the state of information technology and strategic planning efforts at other colleges and universities. This workshop should follow the general model for strategic planning and emphasize the linking of strategic planning for information technology with the institutional planning process. The workshop should cover the basic concepts of *data versus information; the array of managerial actions; decisionmaking styles and the differing roles of information; and the application of a strategic planning model to a unit within an institution.* Other areas such as the external environment, both technical and non-technical should be covered, as well as the major strategic planning issues.

2. Gather strategic plans for information technology from other appropriate institutions to serve as examples.

3. Develop and summarize an overview of the strategic planning and budgeting process and the steps appropriate for the institution.

4. Develop a policy and advisory committee structure for information technology, including:

    a. Committees and specific charters. Gather and consider example committee charters from other institutions.

b. Determine committee chairs and representatives based on examples from other institutions of comparable complexity and size.

c. Develop committee appointment and operating procedures within the structure of existing institutional committee guidelines. Clearly document these procedures.

5. Develop an academic computing seminar agenda appropriate for the institutional culture, then identify topics for discussion, moderators, and participants.

6. Develop an administrative computing seminar agenda, then identify topics, moderators, and participants.

It should be obvious, but be sure to obtain approval for the *Plan to Plan* from the appropriate institutional administrators before proceeding with the orchestration of the full planning process.

THE STRATEGIC PLANNING PROCESS

The following paragraphs suggest the steps necessary to develop an ongoing strategic planning process for information technology for the institution. Institutional documentation and procedures for the process should be prepared in cooperation with institutional staff who will be responsible for accomplishing them.

1. *Establish the planning parameters.* This process determines who does what and how the planning process for information technology will relate to the institutional strategic planning process.

2. *Assess the external and internal environments.* Since these assessments may be conducted at varying levels of detail, it is important to determine the level of effort for appropriate the institutional culture. Analysis of the external environment should identify and assess major forces in the economic, social, technological, political and legal, demographic, and competitive areas that will present specific opportunities, threats, and constraints to the institution. Assessment of the internal environment includes identifying the strengths and weaknesses of the organizational resources such as human, physical, technological, and financial.

3. *Determine institutional and constituency values.* Include solicitation and documentation of perceptions of and expectations for both academic and administrative computing in the planning for this step. Conduct campus interviews with all of the major technology clients and document their opinions.

4. *Identify areas for strategic decisions.* The specific areas typically addressed in this step are: organizational mission, clientele, goals and outcomes, service mix, service areas, and, comparative advantage. Discuss the strategic decision areas in the planning committees, then review staff descriptions of alternatives in each of the six areas. Address alternative organizational structures, as well as the institutional hardware and software environments and the academic and administrative applications portfolios.

5. *Develop functional and operational strategies.* This step deals with how each of the strategic information technology issues will be addressed, by whom, and through what processes. Base discussion and suggestions for descriptions of the functional and operational strategies on successful models from other institutions. Develop and document specific action plans for each of the major information technology organizational units.

6. *Develop strategic objectives for the planning year.* The final step of the strategic planning process is to come to agreement on a set of strategic objectives for the planning year. These objectives include development and/or acquisitions of new information technology products and services as well as maintaining and improving existing systems. It is important to allow for iteration in the planning process, since many times other institutional units develop objectives that create information technology objectives that may well be unbeknownst to the information technology unit.

THE STRATEGIC BUDGETING PROCESS

Executive and top level policy committee involvement with the typical strategic planning process ends at the point of agreement upon objectives, leaving operational units to accomplish what they can within limited or reduced resources. Responsibility for achieving the objectives then shifts entirely to the operational managers

While it may seem relatively simple and somewhat mechanistic, this strategic budgeting process explicitly focuses executive attention on the activities and resources necessary to successfully meet the objectives. This is accomplished by using a series of steps that relate resources required for operational activities to agreed-upon objectives. The process allows value judgments on resource allocation and trade-off decisions to be made at a strategic level before operational projects are undertaken rather than being forced to make costly mid-stream adjustments when resources will not stretch to cover over-optimistic objectives, or when in-process operational failures occur.

The first step in the process is to briefly describe and identify all of the agreed-upon strategic objectives for the planning year. These objectives are then listed across the top of a standard spreadsheet. After agreement upon the objectives, all information technology activities required to achieve those objectives, as well as all on-going activities, are briefly described and identified, then listed down the side of the Objective-Activity Matrix. After constructing the basic matrix, a "1" is then placed in the spreadsheet cell under each objective supported by each activity as shown in Figure 1. The first pass at this exercise can be completed by information technology staff, then reviewed by the appropriate strategic planning committees.

	Objective-1	Objective-2	Objective-3	Objective-n	Total
Activity-1	1		1		
Activity-2	1			1	
Activity-3			1		
Activity-4					
Activity-n	1		1	1	
Total					

Figure 1: Objective Activity Matrix 1a

After all objectives and activities have been entered in the spreadsheet, the Objective-Activity Matrix is then summed vertically and the bottom line checked for totals of zero as shown in Figure 2 below. Any objective indicating zero supporting activities obviously cannot be achieved, so must either be eliminated, or have supporting activities added to the list.

	Objective-1	Objective-2	Objective-3	Objective-n	Total
Activity-1	1		1		
Activity-2	1			1	
Activity-3			1		
Activity-4					
Activity-n	1		1	1	
Total	3		3	2	

Figure 2: Objective-Activity Matrix #1b

After all zeros on the bottom total line have been eliminated, the Objective-Activity Matrix is then summed horizontally as shown in Figure 3 below. If any activity indicates zero objectives supported, either there is an unlisted objective, or there is some question why that activity exists. In most cases, an ongoing objective has been overlooked.

	Objective-1	Objective-2	Objective-3	Objective-n	Total
Activity-1	1		1		2
Activity-2	1			1	2
Activity-3		1	1		1
Activity-4					
Activity-n	1		1	1	3
Total	3	1	3	2	

Figure 3: Objective-Activity Matrix #1c

Once all zero totals have been resolved, the resources required for each activity are identified, both dollars and full-time-equivalent (fte) staff. Allocation percentages for activity resources are then estimated and entered for each objective supported as illustrated in Figure 4. These two exercises are usually accomplished by information technology staff, then reviewed by senior administrators and the information technology policy committee.

	Objective-1	Objective-2	Objective-3	Objective-n	Total
Activity-1					$ fte
Activity-2					$ fte
Activity-3					$ fte
Activity-4					$ fte
Activity-n					$ fte
Total					

Figure 4: Objective-Activity Matrix #2a

After resources are allocated and summed vertically, the estimated costs for each objective are displayed as shown in Figure 5 below. Value judgments can then be made by the information technology policy committee as to the costs and benefit of each objective. If the estimated costs shown in the lower right hand corner of the Objective-Activity Matrix exceed those available, value judgments can also be made as to which objectives should be modified, postponed, or dropped.

	Objective-1	Objective-2	Objective-3	Objective-n	Total
Activity-1	$ fte		$ fte		
Activity-2	$ fte			$ fte	
Activity-3		$ fte	$ fte		
Activity-4		$ fte		$ fte	
A\ctivity-n	$ fte		$ fte	$ fte	
Total					$ fte

Figure 5: Objective-Activity Matrix #2b

CONCLUSION

Recent technological developments in both computing hardware and software present dramatic opportunities for colleges and universities, but planning and preparation are required to capitalize on those opportunities. The current industry emphasis on campus-wide networking, client-server computing, and the graphic user interface require major changes in traditional institutional computing and communications environments, but these changes will not happen without executive involvement and leadership. The process of strategic planning and budgeting described in this paper can focus institutional attention on the appropriate institutional issues, and with institution-wide involvement, formulate a common vision for information technology.

Footnotes:

1. James I. Penrod and Thomas W. West, "Strategic Planning for Computing and Communications," *Organizing and Managing Information Resources on Campus*, (EDUCOM, 1989), pp. 117-139.

2. Robert C. Shirley, "Strategic Planning: An Overview," Successful Strategic Planning: Case Studies, New Directions for Higher Education, No. 64 (San Francisco: Jossey-Bass, 1988): pp. 5-14.

3. John Moynihan, "Propositions for Building an Effective Process," *Journal of Information Systems Management* 5, no. 2 (Spring 1988): pp. 61-64.

4. Donald Lelong and Robert Shirley, "Planning: Identifying the Focal Points for Action," *Planning for Higher Education*," vol. 12, no. 4 (Summer 1984): p. 4.

IMPLEMENTING A NEW SYSTEM ON TIME IN BAD TIMES

Elaine David

The University of Connecticut
Storrs, CT 06269

Abstract

In 1991, the Student Information area of the University of Connecticut Computer Center faced many problems. There was very little documentation, many jobs were not in production and being run as 'test' jobs from programmers' machines, and the staff had no overall knowledge of the projects under development. In addition, bad economic times had resulted in the loss of many knowledgeable personnel.

In the midst of these difficulties, the student information group was assigned the task to implement a university-wide touch-tone registration system.

In order to cope, we restructured our group to insure rapid development and first-time perfect operation of the new system. This paper will discuss our new standards and procedures, the problems we have encountered, and the progress we have made toward achieving the goal of installing a touch-tone registration system which would work perfectly the first time.

IMPLEMENTING A NEW SYSTEM ON TIME IN BAD TIMES

INTRODUCTION

In the Fall of 1991, in an effort to cut down on payroll expenses to the State of Connecticut, University of Connecticut employees were encouraged to take early retirements and voluntary layoffs rather than face mass firings.

The loss of staff within the University community resulted in greater demands on the computer center for additional computerization of University office functions to increase University efficiency; the loss of highly knowledgeable staff within the computer center made meeting these demands more difficult. Since there was no possibility of replacing lost positions for the foreseeable future, Administrative Services decided to consider the possibility of restructuring its staff in the hope of becoming more efficient.

In December, 1991 Administrative Services was restructured to consist of 3 teams of programmers each headed by a Team Leader. Each team would be responsible for multiple projects and team members would move from project to project within the team, depending on need. Team 1 was assigned student and academic applications, including the student information system, the auditing/advising system, and the new (yet to be programmed) touch-tone registration system. The team consisted of 2 senior programmer analysts (one of which was made team leader), 3 programmer analysts, and 1 programmer (transferred from production support).

Although the main impetus for using the team approach was the need to restructure due to the loss of personnel, Team 1 viewed this change as an opportunity to improve the overall student information system. Over the years, many of the team members had voiced concern about some of the ways we operated. With the formation of this team we decided to review the concerns we had, rank them and develop a plan for improving the way we worked.

CURRENT SYSTEM PROBLEMS

In meeting with the team, three main areas of concern were identified: personnel concerns, current system concerns and new system concerns.

The loss of 3 key members involved with student information systems in November, 1991, meant a loss of 60 years of combined experience. The manager who left had written many of the original student record systems programs which were still part of the newer system. His loss meant that any problems with or changes to these programs would create a problem for the computer center staff. The project leader who left was a trusted member of the University community. She served as a primary interface with the various departments, and was very knowledgeable in their needs. The primary analyst who left was the person involved with maintenance of the files, and who oversaw grade processing. Also, he was the person who had investigated the purchase of a voice response system for the new registration application.

By November, 1991, the morale of Team 1 was at an all time low. Not only did they have to deal with the added stress of increased work loads, frozen salaries, and lack of certainty about the future, but they also had to deal with the fear of failure due to lack of knowledge (regarding both specific tasks, and a general overview of the entire system).

Because of the prior stability of staff and the number of staff members involved in the student/academic systems, the computer center had allowed itself the luxury of permitting specialization. The staff member who was initially involved in a particular programming task was later the person to be involved with any modifications or problems dealing with that program(s). In short, we had permitted 'ownership' of information. This practice was beneficial in enabling us to do tasks quickly, but the lack of cross-training backfired when we lost programmers involved in some of the major areas. Given that there was no hope for new staff, and that no current staff member was familiar with the overall student record system, it was time for us to require a broader knowledge base of the staff, and to begin a program of cross-training.

In our team meeting discussions several major problems emerged. The first problem that we noted was that not all jobs were in production. (Only production jobs are scheduled by the user through the scheduling office.) Some jobs were still in the 'test' library and were being scheduled by a programmer at the request of a user. Other jobs were being run by programmers from a programmer's machine at a user's request. Also, many "errors" were being corrected "on the fly" without being logged in via a service request. This practice permitted undocumented modifications on user demand without factoring in other requests for programmers' time.

The second problem we encountered was the lack of documentation (or minimal documentation) of the system (jobs/programs/interfaces). This meant that anyone other than the programmer who was initially involved with the job/program/interface would have difficulty determining the nature of any problem and method for proceeding when a problem developed.

The third concern with the current system involved grade processing. This had always been a major effort by the computer center. It had been handled by two of the members of the staff who had recently left, and required all night overseeing by them. It was a process which rarely (if ever) ran smoothly, although the specifics of what went wrong were not known, as the procedures involved had not been documented. Grade processing was next scheduled for December 30, 1991 (1 month away from the time of re-organization), and would require the team's immediate attention.

Despite the financial problems at the University, the administration continued to maintain its strong commitment to the need for a touch-tone registration system. The then current system of processing pre-registration requests using a batch system and handling over 7000 students at add/drop using punched cards was no longer considered acceptable. Although some online capability already existed for the regional campuses and the continuing education office, this capability was not available at the Storrs campus. In addition, a 'promise' had been made by staff members who had since left that it was feasible to have a new registration system up and running by August, 1993. There was not a single computer center staff member remaining who had been involved with the touch-tone project. Although a general plan existed, no detailed analysis of any of the 'subfunctions' of the system was available.

To have any chance of meeting this new challenge, it was necessary to move immediately to ascertain what needed to be done, what could be done, and the resources required to get it done.

IMPLEMENTING THE TEAM CONCEPT (ASSESSMENT)

In December, 1991, a detailed analysis and plan were prepared for submission to the Touch-Tone Steering committee (consisting of the Associate Provost, members of the Registrar's office, associate deans from several colleges, and computer center staff). The analysis showed all of the tasks required to meet

the goal of the original project plan, and a time estimate (in hours) for each task. By reviewing the amount of time team members had spent on previous projects, it was estimated that the team could (at best) devote 70% of its productive time to this new project, and still manage its other functions. Using this information, it was clear that we could not provide all of the functions requested by August, 1993. However, a plan was proposed which called for a phased-in approach to introducing touch-tone registration. The plan called for an online method for handling add/drop for August, 1993, which would eliminate the need for punched cards and reduce the lines of students waiting to change courses. Also, the plan called for a touch-tone tone registration system to be available for add/drop with limited functionality for January, 1994, and a fully functional system available for January, 1995 which could handle not only the add and drop period but also the pre-registration period. It was imperative that the touch-tone registration system be operational within this new time frame; because of the high visibility of the project, it would have to work perfectly the first time.

To address the concerns of the team involving personnel issues, current system issues, and new system issues, and insure the success of the new registration system, the team decided that it would be beneficial to hold frequent working sessions to determine how we would proceed and to keep everyone informed of what was happening.

IMPLEMENTING THE TEAM CONCEPT (GRADE PROCESSING)

We began by reviewing the current schedule submitted by the system administrator from the Registrar's office and the schedule from the computer center scheduling office which showed dependencies and run times for each job. We decided that since we were sufficiently unfamiliar with the process, we would carefully monitor grade processing in December to insure that any problems would be detected as early as possible (hopefully prior to the printing of grade mailers and transcripts).

We determined potential places for failure within the process and decided to back up our files before the running of these jobs as a safety measure. We also discussed how we could know that a particular job was producing the correct results when the job ran successfully. For many of our jobs, summary reports were produced. However no one was looking at the reports until the following day, when the entire grade processing had been completed. These jobs would now be flagged to indicate that the process was not to continue until the reports were read and approved. Jobs which were not producing 'readable' reports were modified to provide better information.

In reviewing the current grade processing schedule we noticed that processing jobs and printing jobs were interspersed, so that jobs which required checking by the user might occur at 2:00 a.m. The schedule was revised to do the processing first and the printing of transcripts, mailers and letters later in the evening, with the expectation that if all the processing was correct then the outputs would also be correct.

Before running the grades, the team did a walk-through of the process and discussed how we would recover if a problem occurred at any stage of grade processing. These recovery procedures and the additional jobs needed for recovery were then documented.

Although we felt we had done a good job in improving grade processing the team decided to be available during our first trial. With pizza donated by our director to fortify us, we watched as job after job ran successfully. We checked all the output summaries, verifying the information reported and all outputs, giving special attention to grade mailers, probation and dismissal letters and transcripts.

The December running of grades was the fastest and best grade processing the University ever experienced. Our future goal was to have grade processing run smoothly without the need for the computer center programming staff overseeing the process. This goal was accomplished the next time we ran the grade schedule, in May, 1992.

Once grade processing was no longer a major concern to the team, we decided to tackle the problem of the current Student Records system. We realized that we could not undertake a major new project if we were going to be constantly pulled away to handle 'problems'. Therefore we needed to put together a plan for minimizing 'problems' so that we could focus on new tasks. It was important to insure that we would in fact devote 70% of our productive time to the new registration system, if we were to meet the deadline that was set.

IMPLEMENTING THE TEAM CONCEPT (OTHER PROBLEMS)

The first item in our plan was to continue our group meetings to discuss problems, issues and overall design objectives. We decided to schedule regular weekly two hour meetings to discuss general issues and to schedule other meetings as needed. The team set the agenda for each meeting, including any questions or concerns they had, and the agenda was distributed prior to the meeting. In addition, a running task list was maintained by the team leader and at the beginning of each meeting outstanding tasks were reviewed to determine their status. New tasks were added to this list as they were assigned to the team members.

We decided that our next priority would be to review every job that was part of the current system, and put all non-production jobs to production status. This review included jobs which were on programmers' machines, test jobs in the test library and pre-production jobs in the test library. During this review we found that in some cases we had several versions of a job. In the past this had created problems when we went to change the wrong version of a job. As a group we determined which was the correct version and deleted all other versions from either a programmer's machine or from one of the libraries. By May, 1993, we had cleaned up the test library and put all our jobs to production status. As part of the process of putting jobs to production status the team re-instated a policy of creating programmer and user documentation to accompany all production jobs. Also, a policy was established that all team members were required to spend 10% of their time creating documentation for 'old' jobs. We decided that this documentation would reside on a special machine to which we all had access, and that all job and program documentation would follow a specified format that we created. One of the team members was assigned the job of insuring that new documentation adhered to the standards and creating an index to the documentation. By July, 1993, we had created 400 pages of new documentation.

Several programmers had noted that they had created 'special' jobs/procedures for handling problems that they had to deal with. These jobs were located on their own machine. To improve the technical competence level of the team we decided that the procedures would be documented and the jobs would be put in the 'test' library. We came up with a naming convention for identifying these jobs and distinguishing them from test jobs which would eventually go to production. Ultimately, this permitted flexibility in assigning 'problem' tasks to programmers. If the documentation was well written and the job/program was available then anyone could solve the problem without the need to 'reinvent the wheel'. Each time documentation needed to be used team members were provided an opportunity to reassess the usefulness and accuracy of the documentation. A programmer who did not feel the documentation was sufficient for his/her needs went back to the programmer who initially wrote the documentation and asked for improvement. In several instances programmers would ask another programmer to review their

documentation before it was finalized, rather than have to redo it later. Although the team members initially balked at the tedium of having to create documentation, they have been relieved at knowing that they are now no longer the only people who can handle a given problem.

Another technique used to increase the versatility of the team members was to have one programmer work with another, more knowledgeable programmer on a particular problem. Team members were more willing to take criticism from their fellow team members than from the team leader who would be responsible for evaluating them. This process also promoted the team concept.

IMPLEMENTING THE NEW SYSTEM

In March, 1992, the team began tackling the new tasks for the registration system. Each month we monitored our progress, by using an in-house reporting system called Time Track. Much to our dismay, we discovered that we were not spending 70% of our time on the registration system as we had planned. In analyzing the data we learned two things. First, programmers were spending considerable time responding to *ad hoc* (telephone) requests from users. We also learned that the written service requests from the Registrar's office that were being submitted were not being ranked in priority order. Not only was the time we spent on other tasks affecting our ability to work on the new system, but the constant telephone interruptions from users were causing the programmers to lose concentration when they were working on the new system. Although we had improved our efficiency through proper documentation, cross-training and improved procedures, we now needed to improve the work habits of both the programming staff and users to meet our ultimate goal.

In meeting with the team, several problem areas were identified.

1. Users were interrupting the programming staff with telephone calls which were in fact service requests.

2. Programmers found it difficult to say 'no' to *ad hoc* requests which only took a 'couple of hours' of their time.

3. Because there was no paper trail for many of these *ad hoc* requests, specifications were not finalized prior to programming and therefore what a programmer initially thought would be a two hour task could actually take several days.

4. Usually service requests were not being properly prioritized by the users, so that a 'nice-to-have' enhancement would be sent in with a vital 'must-have', without being distinguished.

Team members felt that while the current system was negatively impacting their productivity, they did not feel comfortable with denying users *ad hoc* requests, even though it was the policy of the computer center to require written requests. The programming staff had worked closely with many of these users, and a good rapport had been established. They felt that it was important to preserve these relationships, and they felt that by denying any request they would jeopardize the good relationships. A suggestion of having all telephone calls go through a central number so that they could be screened was overwhelming rejected by the team. Since many of the team members had young children, they felt it was important to have direct outside phone lines. It was decided that we first needed to educate the users about our policies and how they would be implemented, before we could expect the programming staff to adhere to the

policies. The following policies were restated and agreed to by the administration and Registrar's office staff.

1. Telephone calls to programmers would be limited to those required for the implementation of an already assigned service request. The users were informed that programmers would no longer be permitted to spend their time servicing undocumented requests.

2. All requests for service would be handled via a written service request. Any emergencies, which required immediate attention and could not wait for submission of a service request would go though the team leader. (Even emergencies were required to ultimately have a service request submitted and a number assigned to the task for auditing purposes).

3. Until we felt comfortable that we could meet the time line developed, we would only handle vital (emergencies and mandates) requests. Any other service request demanding our attention would require the signature of the Associate Provost before it would be done. This practice would insure even-handedness for the users.

At first the users were unhappy with the rigor that was being imposed, but as time went on, and the quality of our service improved, more and more users directed their call to the team leader and submitted proper written service requests.

ACHIEVING OUR GOALS - A PROGRESS REPORT

As we continued to develop the new registration system we implemented several new techniques which in retrospect were crucial to our initial success.

First, we kept a paper trail of all communication concerning the new system. This was a carryover from requiring a written service request. For the touch-tone project we decided to request sign-off of written design specifications for each subtask, be it a new directory of classes, extended security, development of a scheme for creating access time blocks for students to call, or developing an administrative online add/drop program. Programming did not begin until all of the specifications were determined for that subtask and we had a written sign-off. We wanted our programs to be of first quality and the only way we knew to achieve that goal was to avoid modifications to the original programs once implementation had started. What we offered to the users was our analytic skills in developing good specifications, clear documentation on the specifications, and a walk-through to insure that the programs (once developed) would serve their needs. In return, we demanded from the users full attention to the analysis and walk-through, as well as written acceptance of the specifications.

We never rushed the users to accept specifications before they were ready to do so, but the users knew that programming would not begin until they gave the 'go ahead'. In addition, the users were responsible for final testing and acceptance of the program as meeting the specification agreed upon. We agreed that unless the program did not meet the specifications, or there were policy changes which affected the program, we would not alter a program. Of course we would always make modifications due to programming errors, but we believed that these situations would be minimal once we received written acceptance.

Because our goal was to have a product which would be perfect the first time it was used in production, the team decided that it was important to provide for a large safety window for testing. This meant that at times we needed to 'perfect' the basic system before adding enhancements. Our programs were written with sufficient flexibility to allow other functions to be added in the future.

Not only did we require behavior modification from our users but the team underwent its own kind of behavior modification. The old attitude that it was all right to make mistakes as long as you were available to correct the problem when it was discovered was no longer acceptable. Not only could we not afford the time to correct programs, but too many mistakes had hurt the respect and trust from the user community. We began to improve our own testing techniques. For batch programs this meant running the production JCL with the minimum number of changes possible, and including the output from the test with the production run book (which contained the JCL, message library and programmer and user documentation and was available to the scheduling and operations areas). The team leader would not sign off on any production run book without the inclusion of the output from a test. For online programs, programmers, after doing their own testing, might enlist help from fellow team members. The same documentation that would be turned over to the user was given to a team member and the team member was requested to try out the software. Feedback from the testing was discussed at our weekly meetings, and any necessary changes were made prior to turning over the software to the user for testing.

One of our main concerns in using an online add/drop program, in which over 60 terminals would be available for add/drop and a larger number of terminals would be available to access the Master Schedule to provide student with information on the availability of classes, was that our system would not be able to handle the additional number of I/Os expected without severe degradation to the entire CICS system. To deal with this potential problem the team included Systems personnel in the early stages of planning performance testing. Not only did they take part in file design analysis and assist us in developing a test plan, but they also carefully monitored the CICS system during testing and after the programs had been put to production. The reports that they produced for us provided us with the information needed to define our VSAM files in the most efficient way possible and decide where to put files to minimize contention.

One of the contributing factors to the success of both the online administrative add/drop application and later the touch-tone registration application could be that in designing the requested applications the team also took into account user procedures. Often it was the team that first recognized that current procedures were no longer compatible with a new computer system. This was exemplified when we went from a punched card system for add/drop to a computerized system and the team recognized that a mechanism for advising the student of cancelled courses and courses which no longer had seats left would need to be developed, as the presence or absence of cards in a box would no longer provide this information. As a result the team developed a public access program to the Master Schedule which students could use. Since the Registrar's office and other administrative personnel already had online access to the Master Schedule, the importance of this addition to the project was not fully appreciated by the administration until add/drop was underway.

By March, 1993, we had completed the programming for the administrative online add/drop system. The documentation for the system was complete; key administrative personnel had been trained; testing had been done by both the team and the Registrar's office; and we had received a sign-off by the Registrar's office indicating that the system we had met all specifications to their satisfaction. We were now ready to devote more of our attention to the online application which would support the new telephone registration system.

The new touch-tone registration system (TTR) project provided the team with a wonderful opportunity to work in a new and better way. First, we decided that since TTR would be a separate system, we would abandon the standards developed by Sigma Corporation and used in all of our online student record system programs so far. These programs contained many lines of code which was not needed for our installation, and which many of our programmers did not understand. We decided that the programs would be written in COBOL 2 and make extensive use of the temporary storage queue capability. By making this decision we committed ourselves to being pioneers, since we were the only programmers within administrative services to use COBOL 2. Since we would be treading into some new territory, we decided to make all technical decisions as a team.

The team leader, acting as lead analyst for the project, worked with the Touch-Tone Steering committee to develop clear specifications. Most of these specifications (in broad terms) had already been determined when we put together the project plan. However, details were now needed. Once we had sufficient detail from the committee, the team reviewed the information to determine if additional information was needed which might affect the way we designed the system. These questions were documented and sent to the Registrar's office personnel for response. With the information we had, we began to develop an overall design of the system. This design, expressed both in words and as a flow diagram, was then submitted to the committee for review. The committee met, discussed the overall plan, and gave their permission for us to proceed. The team then proceeded to expand the overall plan into more detailed specifications. Since the voice response application would be written by an outside vendor, it was important to have clear documentation of the system, not only for ourselves but also for the vendor. The textual documentation was expanded first, and reviewed by several members of the team. This plan was then translated into a flow diagram by another team member. Meanwhile, other team members used the plan to develop screens and to begin programming the mainframe application. It was always necessary to keep consistency between the textual plan, the flow diagram and the programs. As the textual documentation became more and more detailed, so did the flow diagram and the programs. During this detailed design phase the Touch-Tone Steering committee was unavailable for meetings because of scheduling problems.

Before the beginning of the programming phase, the team made several design decisions, to insure consistency from one program to another. Several different members of the team were assigned different parts of the programming effort. While this meant that we needed to meet more frequently (now twice per week), the team felt that the time spent during these meetings was beneficial.

Once the programming was complete, it was time to begin testing - not only that the programs worked, but that they worked based on the documentation and flow diagrams that were developed. Each member of the team was given a part of the system to test which was different from that part of the system that he/she had programmed. Since we were sure that errors had to exist, the goal was to uncover as many of them as possible before we turned the system over to the user and the vendor. Rather than be embarrassed by errors which were uncovered, team members would thank each other for discovering an error.

For three months the team tested and simultaneously modified documentation, flow diagrams and programs. Although errors were uncovered, none of the corrections required a change to the basic design of the system. On September 21, 1993, the Touch-Tone Steering committee was given a demonstration of the mainframe application with simulated voice response, which would be used in the TTR system. It would have been nice to be able to report that the committee was completely satisfied with the new system. Unfortunately, although the system performed to specifications, the committee demanded changes that would affect the design of the system. This failing is clearly a result of the different time scales which the user community and computer center community operate. Users are embedded in their day to day concerns

while giving only partial attention to specific fragmentary questions of design while the programmers and analysts are completely stymied by questions which are either not answered or answered insufficiently. The users on the Touch-Tone Steering committee who were expected to act as consultants to the project were not released from any of their normal obligations; therefore they had very limited time to devote to the project. Critically reading detailed documentation and finding time to meet regularly as a group proved impossible. In our case when the users finally gave their full attention to the Touch-Tone project they discovered that assumptions made by the programmers and analysts although workable were not to their liking and they demanded changes. This particular problem is one that deserves consideration by the administration for future large scale projects.

CONCLUSION

At the time of finalizing this manuscript, we have not yet implemented the touch-tone registration system which is scheduled to take place in January, 1994. But whether or not we make the deadline the team feels that it has learned several important lessons, which have permanently changed the way we work, the way we interact with each other and the way we deal with the user community.

We recognize and have gotten the computer center management to understand the importance morale plays in our performance. Whenever possible we encourage users to write notes of appreciation. As a team we celebrate our successes, and we have come to realize that our individual successes are in fact team successes. Although the monetary resources of a state institution are limited, management has tried to implement reclassification in a more timely manner to insure that the staff is working at its potential.

In reflecting over the past two years, our team has come to appreciate how far we have come in improving our own work habits to provide better products and better service to our users. We turned a difficult situation into a window of opportunity to review our past practice, and explore innovative changes in methodology. Although meeting as a team is time consuming, we find that the time is well spent. The ability to discuss problems, issues and overall design objectives has cut down on possible errors, and has caused each of us to feel part of every project. Having clear documentation centrally located has enabled us to support one another and ultimately give better service to the users, who no longer have to wait for the availability of a specific programmer. Each of us, in one way or another has increased our technical competence, either through formal training, through workshops given by our colleagues, or through self-teaching. Each member of the team has a clearer understanding of the overall student information system.

We have made strides in improving our communication with the users by maintaining a paper trail of requests, specifications and desired enhancements. By requiring a sign-off of written specifications before programming and after job acceptance, we have increased the likelihood that we understand what the users wants and the users understand what they have been given.

As a team we have worked to install a new system, beginning with a needs analysis and progressing to a description of the system, flow charts, top-down design, programming and thorough testing of the system. We feel that as a result of the rigid standards which we agreed to adhere to, we were able to create a first-class product.

Team 1 wishes to acknowledge that the success they have achieved could not have come about without the support and backing of its management, for which we are grateful.

Quality Software ...
But by Whose Definition.

Is the End-user King?

By
Louise M. Schulden
Cornell University

Introduction

Software is playing an ever-increasing role in critical business processes. Yet software quality has not received the attention needed for such an important company asset.

Current software quality levels in the US result in software with approximately 4.5 defects per 1000 lines of executable code. This is an unacceptable level of quality. Japan is doing 3-fold better with 1.5 defects per 1000 lines. Motorola and IBM have launched quality programs striving for six sigma quality in software, or 4.3 defects per million lines of code. This may be excessive and addressing the wrong problem. How bad is the problem? A 1988 US Government Accounting Office surveyed the success, or otherwise, of software projects for their division and found that of a 6.8 million software budget the results were:

Software Projects for US Governmental Accounting Office 1988
47% (3.2 million) software delivered but never used
29% (2.0 million) software paid for but not delivered
19% (1.3 million) software abandoned or reworked
3% (0.2 million) software used after changed
2% (0.1 million) software used as delivered

Total quality management, quality improvement programs are common place in most industries, particularly manufacturing, and in most industries the payback has been incredible. The word quality is used in everyday speech to describe the degree of excellence of a product or service. But in the interum quality programs for software have been allusive. The first problem is a definition of software quality. There is confusion about what is meant by the term software quality. Part of this confusion may be caused by the different perceptions of software quality existing between people; software developers vs traditional quality assurance people vs end-users. There are different dimensions of quality which are important when considering the quality of a software product: performance and features, reliability, conformance, durability, serviceability, aesthetics, perceived quality, etc, It seems clear that quality is not easily defined, except arbitrarily, and that there are a number of dimensions to it.

This paper would like to present the software quality challenge. It starts with the important definition of what is the meaning of quality software to your institution and more importantly those who ultimately stand in judgement of IT (Information Technology) products and services, the end-users. Then how does a company organize a Information Technology quality improvement effort? What is the process for addressing quality trade-offs? What role does the customer play in all this? What is their definition of quality? What software and system attributes are important and to whom and how do we measure them? What tools or processes or ideals to use and follow will improve the quality of our software? Finally, how do we evaluate if our efforts are successful... worthwhile?

Misconceptions and ...

The first misconception about software quality is that IT management and staff know what quality is. When problems occur or customers become dissatisfied, it becomes immediately obvious that the software is of poor quality. Yet the IT response to the quality question remains essentially reactive rather than focused on searching for ways to build quality into software and services.

Second misconception, quality can be related to an "acceptable" level of failure. An old IBM advertising campaign asked: "if your failure rate is one in a million, what do you say to that one customer?" Unfortunately, all too often we are measured by our failures.

Third misconception, quality is an expensive luxury. The cost of quality in software is the cost incurred by delivering faulty systems. These costs encompass not only the cost of correcting the fault, but the costs incurred by the business due to the fault such as, lost orders, uncollected tuition, dissatisfied customers. The cost of detecting and repairing software failures after they have occurred usually far outweighs the cost of preventing them.

Fourth misconception, quality is free. Quality improvement efforts are by no means free. There are costs to efforts required to prevent mistakes, appraise work done, correct defects. However as long as these costs are less than the resulting benefits, they are worthwhile. The problem is that quality efforts require an investment up front, and it takes time before the benefits show themselves and can be assessed.

Fifth misconception, lack of quality in software is caused by poor quality staff. In fact, most people prefer to do a good job, but will deliver the quality they think is expected of them. If people feel that no one cares whether they produce quality work, they won't.

Sixth misconception, one can *test* quality into software...unit test, integration test, systems test, acceptance test, and finally quality is achieved. Testing does improve quality, but it is costly and still you can miss the mark.

Truths

First truth, users do not weigh equally everything that is right with software against what is found to be wrong. Unfortunately, we get judge by our mistakes. Software that works well is taken for granted. Software that is wrong for whatever reason, is remembered... and often talked about.

Second truth, users do not distinguish between problems caused by the application software itself, and those which are caused by faults in the hardware, system or communication software.

Third truth, whatever is wrong with the software, not meeting requirements, buggy programming, bad communications environment, slow response time, does not interface with vendor purchased or other software applications, etc.,etc. is the software developer's problem. It may not be his/her responsibility, but it will be their problem. It should be noted, this is getting better with more business partnering between the IT function and other business functions within the organization and team work across department, divisional, and institutional organizational boundaries. Still it has a way to go.

Fourth truth, "the best you can do as a computer professional is defend yourself." (DeMarco, 1980)

Why is software quality important?

The crash of a Boeing 767 in May, 1991 was attributed to malfunction of software that caused the plane's engines to reverse thrust in midflight. I expect the people on the plane did not realize when they boarded the significance of that software, but without question the quality of that particular software was of paramount importance to their very well-being. Computers and the software they run from microcode to standard 4th

generation programming languages touch every aspect of our life. For a university, the proper functioning and support of computer software makes sure we have an entering class, students get courses, students get tuition bills and financial aid, room assignments are made for classes, grades are recorded and tracked, and finally diplomas are issued. University software pays the bills (suppliers, employees), collects the revenue (tuition, outside support, alumni gifts), and even controls the heat in our buildings during our cold Ithaca winters. And yet software quality has not received the attention needed for such an important institutional asset. When facility building runs over budget, we cut moneys for parking to cover the loss. So software quality is the parking lots of many of our expensive high-rise system software development efforts. And yet because of hard financial times, institutions like Cornell must look to the strategic application of Information Technologies as a source of competitive advantage with other educational institutions in the future.

A high quality purchasing system will allow Cornell to pay bills in a more timely fashion and consolidate orders, thereby saving millions by taking advantage of volume and early payment discounts offered by vendors. Quality administrative systems free up not only staff but faculty, allowing the institution to save dollars in staff reductions and better utilization existing employees. Freeing up faculty time, allows them more time to go after grants and perform better research and teaching so more moneys flow to the school. Poor quality software or information technologies solutions COST BIG TIME. When an administrator says that they'd rather fill out a form than use the system, IT has a problem. When a faculty member calls, and complains they just wasted a day trying to send a document because of faulty communications software, IT has a problem. When 70% of your IT staff is spending all their time fixing bugs and maintaining software so it runs in production instead preparing for the new and future needs of the institution, IT better start looking for work in another field.

What is Quality? Quality Defined.

One of the early works to define quality resulted in Garvin's 5 approaches to defining quality. Garvin recognized that one approach to evaluating quality would not fit all situations. Consequently, the result was five approaches with the advice to follow the one that will most likely give you the result you seek. What you can see in computing is an evolution of the quality definition. Garvin's 5 approaches to quality include: the transcendent approach, the product based approach, the manufacturing approach, the user based approach, and the value based approach.

Transcendent approach is software is viewed as its innate excellence. In this case, the software would be viewed as a work of art: new, visionary, inventive. Quality is an unanalysable property. One can only evaluate on gut feel. Unfortunately, far too many computer professionals feel this way about their work. It is this path that has caused IT to find themselves in the predicament their in. For years, computer professionals were rewarded for reinventing the wheel. Now, there is just not enough time or money and there is far too much work, to encourage this behavior. Programming must stop being art, and be a business. If I have a print routine, writing another one that is unnecessary, is not excellence, it does not contribute to the quality of the IT function even if it is well written software. We very rarely have the resources to revisit the same problem or need twice. One step further, if I can purchase a print routine that meets the organization's needs for the optimal cost, then that is the quality thing to do. The transcendent approach may be how computer people judge each other, but is not an institutional approach to software quality. It was probably most applicable prior to the 1980's, when in fact computing was still in it's infancy and time of discovery.

1980's Quality Definition - Product and Manufacturing Approach

Product based approach is software quality is related to the presence or absence of some attributes or characteristics and that these attributes can be objectively measured and consequently so can the software's quality. The manufacturing approach equates quality with conformance to stated requirements. The combination of the two, software that contained code possessing the professionally accepted quality software attributes/ characteristics and conformed with stated requirements was the goal of the 80's. It represented what 1980 programming shops consider acceptable and quality product.

Those of us who got our computer training in the 80's, were brought up on attributes or software characteristics that were signs of quality programming.

Quality Software Attributes and Characteristics

What attributes or characteristics are relevant and traditionally have been considered when considering the quality of a software product? The software literature is full of the attributes such as: correctness, flexibility, efficiency, reliability, usability, extendability, portability, testability, understandability, re-usability, maintainability, interoperability, integrity, and survivability. Top of the list is performance and features.

Performance relates to the primary operation characteristics of the software.

Features refer to the secondary characteristics that supplements the software's basic functions. (NOTE: Both performance and features are measurable, but it does not follow that the user perceives differences between different software as significant in quality terms).

Efficiency ,the amount of computing resources and code required by a program to perform a function.

Usability ,the effort required to learn, operate, prepare input for, and interpret output of a program.

Reliability ,the extent to which a program can be expected to perform its intended function with required precision or the probability of a software product failing with in a specified period of time. Unlike a manufactured product, software is more difficult to evaluate on this front due to the fact it doesn't "physically deteriorate".

Extendability /flexibility ,the effort required to modify an operational program.

Portability ,the effort required to transfer a program from one hardware configuration or software system environment to another.

Testability ,the effort required to test a program to ensure it performs its intended function.

Understandability , the effort required to understand the code and what it is doing.

Re-usability ,the extent to which a program can be used in other applications, related to the packaging and scope of the functions that the programs perform.

Maintainability ,the effort required to locate and fix an error in an operational program.

Serviceability ,the ease with which the supplier of the software accepts responsibility and rectifies.

Interoperability ,the effort required to couple one system with another.

Integrity ,the extent to which access to software or data by unauthorized individuals can be controlled.

Conformance, the extent to which the software meets the specification. (This must be measured before and after acceptance of the software by the customer. Deviations may become apparent only after the software has gone into service.)

Correctness ,the extent to which a program satisfies its specifications and fulfills the user's mission objectives.

Durability/survivability, the measure of the length of time that software can be used before replacement.

Aesthetics, yes software can be beautiful.

Perceived quality , the user opinion of the quality and usefulness of the software. This may in fact be the most important. Individuals may not have full information to judge by, but judge they will. Their judgement may also include price and reputation of the software supplier.

As one can see there are many characteristics that contribute to the quality of software, and this list is certainly not exhaustive. These actual represent high-level attributes that can be shown to depend on other characteristics. For instance, if a piece of software is to be maintainable it must be understandable, testable, and modifiable. Given the state of the art in software engineering, growing the the tree in this way until the characteristics at its leaves are objectively measurable may not yet be possible but it is a necessary goal if software quality assurance is to develop. In 1987, Kaposi and Kitchenham proposed a quality profile model as a way of structuring the analysis of the quality of a piece of software. The quality profile of the software is specific to an individual and the application, but has the advantage of separating quantifiable and non-quantifiable factors. It provides a good basis for an explanation of why different people can simultaneously hold different views about the quality of the same piece of software. The quality profile categorization follows:

Quality Profile for a Person,Application

Transcendental Properties (Non-quantifiable)

Quality Factors (Objectively measurable)
- Quality Metrics (Quantifiable)
- Quality Attributes (Indicate presence or absence of a property)

Merit Indices (Subjectively measurable)
- Quality Ratings (Quantification of value judgement)

It should be noted that some of these characteristics are mutually exclusive. Quality is a trade-off. Which attributes should be emphasized?

Quality is a Trade-Off

In addition to identifying the "quality" characteristics there is a problem with conflicts between the quality attributes. After quality attributes of software for an application have been defined, the next major concern is determining which of the quality attributes to emphasize. It is impossible to optimize all quality attributes because of conflicts between the quality factors such as, maintainability being at odds with speed of execution or minimization of storage. A system that is easy to use requires easy access and system openness. By contrast, high integrity requires limited access and a closed system. In a trade-off environment, one must decide whether to emphasize the correctness characteristics (internal controls, data entry, and validation) or the maintainability characteristics (user documentation and simplicity of design). It is important to emphasize the qualities appropriate for the application.

To add to the difficulty, is the issue of cost. People say quality is free. That's not exactly true. Total quality-related costs are often subdivided into four groups: 1)prevention costs (quality planning, employee training, supplier education, etc), 2) appraisal costs (reviews, walkthroughs and other forms of testing), 3) costs of correcting defects discovered before acceptance,and 4) costs of correcting defects discovered after acceptance which have to be borne by the developer. This complicates the cost of quality issue because the cost of quality assurance activities such as appraisal and prevention are more easily estimated than the expected savings.

Over the years, depending on the software and its application some attributes have taken a back seat to others. For example, in the 80's portability was of little importance. Most administrative shops were running large mainframe applications. There was little thought to moving the applications to other platforms. Now with hardware cost plummaging, micro- and mini-computers competing with mainframes on raw computing power, and communications software and networks propagating and improving in reliability, portability is a very desirable software attribute. The type of application effects the ranking of relative priority of the characteristics. An application used by hundreds of decentralized users will place more importance on the quality of useability and nice GUIs, than a system used by a well-trained few.

Motivation to undertake quality assurance activities may be to produce a good product, but usually not. More usual reasons include cost effectiveness or good customer relations or marketing. And despite the definition of quality characteristics and their prioritization, quality software and systems alludes us. What is missing from our definition?

1990's Definition of Quality

In Garvin's user-based approach and value-based approach we may find a definition of quality that we can successful apply in the 90's and next century. User-based approach where quality is related to its fitness for use in a particular application. Quality is related to the software user's satisfaction. Value-based approach combines quality, which is a measure of excellence, with value, which is a measure of worth, by defining a quality product as one which provides performance at an acceptable price or conformance at an acceptable cost.

Sample definitions reflecting this philosophy...

"The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs (ISO 8402 standard)."

"The totality of features and characteristics of a product or service that bear on its ability to satisfy a given need (BSI 1979)."

"The degree to which the attributes of the software enable it to perform its specified end item use (DOD 1985)."

What is not present in these widely accepted standards is the acceptability of cost. Now with finances being tight, acceptable cost must be added to the definition.

Organizing for Quality

Now we know what it is, how do we achieve it? First there must be management commitment. This can not be over-emphasized. Few things are more damaging to quality initiatives than a stated quality policy which is immediately contradicted by short-term imperatives and unrealistic deadlines. Without a highly visible commitment to software quality from management, no quality program will succeed.

A separate functional group with responsibility for quality should be created within the IT function, being careful to make sure the achievement of quality remains the responsibility of every person involved in the delivery of software products and services. The quality group is to advise on procedures, techniques and tools, and provide external, objective quality assurance. The quality specialists must be viewed in a support role of assisting staff in the achievement of quality, rather than a policing role. Management must be the police, so the seriousness of this quality initiative is reenforced. The quality function should aim to prevent problems before they occur through education, and the introduction and support of appropriate procedures, standards, techniques, tools, and training. One of the key functions of this group is to take the customer's satisfaction pulse regularly.

A second group will be needed to spearhead the quality improvement effort. This group would be comprised of members representing different roles in the IT function: business modelists, front-line consultants, analysts, designers, programmers, technical support staff, and operators. Their responsibility should be part-time, and a rotation through this group is advised. These people will define and plan the quality improvement effort, represent their concerns to the quality team, and the quality team to their function. They will be instrumental in the implementation of quality initiatives within their function.

Metrics

It will be difficult to register any improvements in quality unless some measures of quality are established. "You cannot control what you cannot measure" (DeMarco, 1982). The identification of suitable measure, and the assessment of the actual values of each of these measures, is an essential component of any effort to improve quality. One must be careful when selecting measurements. Selecting the wrong measurement could give undesired results. For example, measuring lines of code could result in the illusion of increase productivity, but more likely it will result in extraneous, inefficient code and reduce use of reuseable modules. Measurements might include:

* number of problem reports, change requests received per period of time
* problems or change requests outstanding at the end of each month
* time taken to respond to problems
* number of errors detected and type design, specification, misstated or misunderstood requirement

There are numerous possibilities that should be limited only by imagination, need, and resources.

It is important to implement the right measurements. Common sense and monitoring the results will tell you if you are measuring the right things to get the desired outcome. Secondly, it is important not to select too many measurement (5-6 is sufficient). Remember quality improvement is an iterative process and a long-term commitment. Too many measurements will distract and confuse the direction of the quality effort and be overly costly. Pick largest problem areas first.

It may be difficult to obtain measurements. Inability to take needed measurements in itself is a quality problem, and should be attacked as such. The second step in a quality effort may be developing the means to collect needed metrics after identifying what metrics are needed. Whatever means is used, keep it as simple and unobtrusive as possible!

You'd be surprise how much development and maintenance records you are probably already keeping can tell you. For example, the costs of development efforts is usually readily accessible information. Currently, 79% of all development efforts are viewed as going significantly over dollar and/or time budgets. System usage records are also usually readily accessible due to IT's need to account for machine usage. Currently national usage statics show 45% of all systems never get used. During development: track costs, milestones, the success of unit testing, the amount of reusable code exercised, track record for user acceptance testing. The effectiveness of your systems/software development life cycle methodology can be seen in the number of changes made at each development stage to: the business model after its acceptance, the logical design after its acceptance, the physical design after its acceptance, file changes after physical design, and program changes after unit testing during user testing.

Maintenance Metrics

Maintenance tells you an incredible amount about the quality of existing software. Maintenance can fall under 4 categories: corrective, adaptive, perfective, and preventative. If your organization is doing a great deal of corrective maintenance, fixing bugs, etc. then it is a good indication your IT function needs better systems development cycle methodologies, or modeling tools, or programming standards, or testing procedures. Adaptive maintenance is due to a changing user or computing environment. Some of it is inevitable, but too much is again an indication that user requirements were not defined adequately during systems development. The user requirements required the ability to change and the specifications analysis lacked the quality to anticipate this need resulting in undesirable system inflexibility. Perfective maintenance is often referred to at Cornell as enhancements. Often, our enhancement list is longer than the original specifications. This is a combination of user not recognizing needs and analyst not discovering all user needs prior to software release. It is often a sign of an unrealistic implementation schedule, that was too rushed. Finally, preventative maintenance which is the periodic review of the system to uncover or anticipate problems. If your shop is doing mostly preventative maintenance you are probably running a quality environment and have control of your computing.

Maintenance will tell you alot about the quality of the IT work. Track maintenance costs by system, by program, by programmer. Measure to number of failures per program. Calculate number of hours spent on maintenance and whether it was corrective, adaptive, perfective, or preventative in nature and emergency, urgent, or routine. Develop a profile

of the most common maintenance requests and problems encountered. Look to correct these first in the development process. Quality improvement is incremental improvement.

Tools

The clear statement of quality requirements in the requirements specification is a major step towards the production of good quality software. Software developers must plan and implement software development projects with the objective of building in quality. A desire to produce a high-quality product must be supported with a willingness to commit resources to the three disciplines needed for the activity: development disciplines (such as analysis, design, and unit testing), product assurance disciplines (such as quality assurance, test and evaluation), and management disciplines (such as project and general management). It seems to be generally agreed that this involves activities in the following areas:

1. Establishment and maintenance of a requirements specification. This also serves for the basis for acceptance tests.

2. Establishment and implementation of a process for developing the software. This would include shop design and programming standards. A methodology.

3. Establishment and maintenance of an evaluation process. This involves the production of standards defining what must be done to complete a task successfully and also how the work should be done.

Fifth truth, the biggest single problem encountered in the computing industry is the specification of requirements. Organizations seem to find it exceedly difficult to express what they want in clear and unambiguous terms. The fuzziness particularly is evident where a institution's own administrative function and information are concerned. If the business has problems in this area, computerization is often seen as the way forward. In such cases computerization only succeeds in producing more convincing chaos, not sense. The evidence for this lies in the hundreds of abandoned projects throughout the industry.

If the goal or definition of quality is meeting the user's needs than IT will have to more closely align itself with the customer. Cornell has implemented Business Modeling to separate the software development process from the business analysis. Business modeling has help at Cornell better synchronize IT with the business and in many cases better synchronize the user's with their own business. Business modeling involves everyone who has a part to play in an elemental function that is being studied. It breaks down the function to its smallest parts. With everyone having a solid understanding of the business, it is a wonderful opportunity for reengineering and doing a critical study on where and what kind of support information technologies can best provide. The advantage of this, is time is taken to identify what IS the business. It is an opportunity to reengineer and optimize necessary activities and obliterate worthless activities. All this is done prior to thinking about computerization.

Quality is in the eyes of the user. To understand what the user values, IT function has to move closer to the business philosophically to understand what's important. Computer people know what they value in a quality system, robustness, maintainability, etc. What they don't know is what the user values. To find this out the user should be asked. A simple software characteristic evaluation form filled out by the user will help communicate user defined quality. Early on communication is key, if for no other quality goal than a satisfactory user perception.

Communication must be viewed as a key tool to insuring quality software. Business modeling assists that communication. Early survey sheets from users on what they are seeking and postmortem user survey sheets on satisfaction on prior software products move communication forward. Taking up residence with the user community is also appropriate. Communication is unfortunately often an under utilized tool. Aides to communication such as surveys, e-mail, structured modeling tools (DFD,ER diagrams) are invaluable to the quality effort.

An important tool to system/software development is structured development approach. Often referred to as system development life cycle or SDLC or system development methodology. At Cornell, we have our Systems Development Methodology. This methodology describes:

* the phases of the system life cycle,
* the purpose and goals of each phase,
* the items to be delivered and for each deliverable item, who is to prepare it, what it consists of, some idea of the methods and tools available to create it, and the review process by which the item is accepted,
* the approval process for each phase, how we know it is completed.

Tools to assist this process include modeling tools: data flow diagrams, entity relations diagrams, structure charts, and business function diagrams. These provide several benefits. First they act along with the methodology guidelines as a communication tool between IT and the customer. In many cases these days, automated modeling tools can serve to check for consistency and completeness of the model. And finally, the model serves as documentation.

A glossary or data dictionary of terms for data elements and other items in the system is a must. Redundant and and inconsistent data definitions may exist throughout an organization's procedure manuals, source program documentation, data files, and in the minds of those in the organization. Ambiguity of what things mean is the makings of software disaster. One can not hope to build quality software, or purchase it, when there is ambiguity of the meaning of the data in the system and how it is used.

Standards are of utmost importance to insuring quality. Well trained software specialists know the best practices for analysis, design, programming, implementation, documentation, and maintenance. Quality demands consistency. Consistency is insure by standards. Many systems development groups operate without standards or have standards they do not use. The most common reason for a lack of standards or not following them (though often not admitted to) is that standards inhibit creativity. I haven't met a systems analyst or programmer yet who did not believe they could determine a better way to do a task than the process proposed by the standards manual. When computer professionals are allowed to do this they have performed two jobs instead of one. They have develop the process that they follow AND follow that process to solve the user's needs. Waste.

Programming standards come in all shapes and forms. Not everyone with a 4 year computer science degree knows how to program well. Standards can help teach good programming techniques. Modular design and reuseable code allows one to create the best code possibly and use in multiple places.

Invest in new automation techniques. Case tools can be used not only to increase programmer productivity, but to institute shop programming standards. CASE generates code faster and reduces code variability. The biggest problem with CASE is sometimes

getting IT staff to use it. Most of them got into this business because they enjoyed programming. The other problem though not as obvious, is customers wanting their own signature systems. With CASE tools you get a standard interface. The resulting purchasing system looks like the resulting budget system, etc. Many users through years of IT lacking standards, have gotten as use to creating their own world as the supporting IT staff have. Both sides of the house must be sensitized to the cost of individual interfaces built for functional area preference vs a common interface for the institution.

One of the best tools, though often overlooked,for improving quality is staff training. Proper training in quality, programming, analysis and design, modeling, software tools, and communication skills is invaluable.

Variability is the enemy. The obvious problem with variability is that the user and other IT staff never know what to expect. Each system operates differently. Maintenance may be easy in one system and difficult in another. It makes user training and new IT staff training a nightmare. And with variability, visible and invisible to the customer, IT credibility for knowing what they are doing suffers.

Testing does improve quality, but it is a costly method of accomplishing the quality objective. Testing procedures should exist for all levels of testing: unit, module, integration, systems, and acceptance. It is important to have a good test environment, that mimics the production as closely as possible. All project plans must include adequate testing time.

Post- Review and Evaluation are Important Tools

The only tangible that matters is dollars. Does the system save more money than it cost to develop (or purchase), maintain, and run. The only intangible that matters is customer satisfaction.

Though these are simply they encompass a world of sins. Money, where it is spent, how much is received, or how well it is utilized is just not that easy to track down. But scrutiny of the business, it's inputs and outputs, and an honest look will show you. Taking an honest look however is not easy. Pet projects, pet agendas, and business processes steep in tradition and sometimes mysticism stand in the way. It is IMPORTANT to quantify as many savings and costs as possible. Look for signs.

* Staff working less/more hours or Staff reductions/increases
* Reduction in costs or budgets
* Reduction in identifiable waste
* Reduction in paper
* Fewer reports or sources of information without completeness or sufficiency of information suffering
* Lower stress or anxiety levels among staff and service receivers
* More business or higher quality business. In a university that might mean more and better applicants to admission, larger alumni gifts, better faculty.

Customer satisfaction is also allusive. Customers sometimes do not know what they want until they see it. They almost always know what they do not want. And alot of the satisfaction depends on the expectation. It is important in addition to systems project management that the IT management manages customer expectations. Customers must be kept abreast of progress all along the way and be the center of system development. This is sometimes difficult. It is not always obvious who the customer is: is it the sponsor, who may never use it such as a VP of Finance, the heads of the function the system serves, such

as the heads of budget for a budget system, or is it the clerk who actually USES the software. This is where business partnering becomes the key. Ultimately the clerk is the customer, but it should be kept in mind the VP and functional head has a broader picture of what is trying to be achieved. Both needs must be reconciled for success. This sometimes requires the administration to align their goals across the organization. But this is not a topic for this paper.

To determine customer satisfaction...ASK THEM often, repeatedly.

Quality of Outsourcing Service

With many of our organizations seeking to cut costs by outsourcing all or part of the IT function, I feel a need to make a comment on outsourcing quality. For many of us the outsourced pieces of IT will become an integral thread of our institution's IT function. The same quality guidelines apply. The quality of products and services, cost-effectiveness, timeliness of deliverables, reliability of performance, flexibility, and responsiveness to the customer, and customer needs being met to their satisfaction is still the measures of quality. Project deadlines must be met or penalties imposed. Measures of performance reflecting customer priorities are part of the contract. In short, the quality guidelines of the IT function should apply to vendor supplied software.

Software Quality is Only Part of Information Technologies Quality

Software does not run in a vacuum nor do customers judge it solely on it's own merits. Unfortunately, this is truly a case of one bad apple can spoil the bushel. Software that runs in a poor quality communications environment is worthless. Bad response time or machine downtime reflects poorly and causes customer dissatisfaction no matter how good the software is. To insure Information Technologies quality one must look at the entire IT picture. Diane Wilson (MIT,1988) identified seven IT assessment methods to evaluate the IT function:

Productivity. Efficiency of expenditure of IT resources.
User utility. Customer satisfaction and perceived value of IT services.
Value chain.. Impact of IT on functional goals.
Competitive performance. Comparison against competition with respect to infrastructure components of business measures.
Business alignment. Criticality of the organization's operating systems and portfolio of applications to business strategy.
Investment targeting. Impact of IT investment on business cost structure, revenue structure, or investment base.
Management vision. Senior management's understanding of the strategic value of IT and ability to provide direction for future action.

In her research, it was discovered only a third of the organizations studied measured the business value or strategic impact of information technology on the business. The dominant measures were ones of cost reduction, increased productivity, and reduced head count. Although more than 70% of the organizations use surveys to determine user needs, about one-third used formal procedures to assess user satisfaction with IT services.

Measurements are also needed to evaluate the value of the system/software in relation to it's performance. What is the strategic value of the software to the business? How does it contribute to the institution's competitiveness? Finally what may be the best expression of the two previous questions, is how satisfied is the end-user? These are the harder, but more important questions. How can these attributes be measured?

IT Quality Assessment

Attribute	Instrument or Metric
Organization satisfaction	End-User Surveys
Meeting Business needs/priorities	End-User Surveys
Contributions to Business Competitiveness	Revenues Gross margin Reduced costs Improved productivity Improved cash flow Cost avoidance-Be careful of this one Improved transaction response time Improved receivables payment Improved departmental performance Impact on end customer (students,etc) Return-on-investment calculations Return-on-asset calculations Impact on products and services
Strategic Value to Business	Establish competitive barrier Create defendable market position Improve service level Introduce technology-based products Introduce technology-based services

Summary

I contend quality is simply meeting the user's requirements both expressed and implied for an acceptable cost. It is not an intangible or subjective factor of rightness or good design. IT IS directly measurable through the user's perceived satisfaction with the product or service, and its tangible costs and benefits can be calculated. The wider context of the service offered by the information technology function should be part of the formula, the majority of users do not distinguish between problems cased by the application software, and those which are caused by faults in the hardware or system software. What we need to aspire to is tight performance measurement linked directly to important business consequences.

Quality is, above all, about people: a continuing commitment to produce quality software and provide a quality service is needed from people at all levels and in all parts of the IT organization. End-users perceive quality in software products and services which most meet their requirements and continue to do so. Techniques, tools, and procedures can improve software quality, but only if they are deployed in an environment which encourages every person to make a long-term commitment to the achievement of that quality as part of a team committed to quality performance. Of all tools, communication is the most valuable. If communication with users is poor, then the perceived quality is likely to be low. The bottom line. The user will judge the quality of the software, the system, the Information Technologies function.

GUERRILLA TQM OR HOW TO INFILTRATE TQM INTO YOUR INSTITUTION

Deborah J. Teeter
Director of Institutional Research and Planning

Jan Weller
Director of Telecommunications

University of Kansas
Lawrence, KS 66045

ABSTRACT

Various models for pursuing TQM are emerging on college and university campuses. Most models and TQM gurus insist on a top down approach for TQM to succeed in transforming an organization. This paper explores the experiences in an organization in which TQM devotees pursue the principles and concepts in their own sphere of influence but without official sanction or resistance from the top. Information technology is part of the guerrilla movement.

Almost by definition, information technology (IT) organizations are accustomed to being change agents in their institutions since they constantly cope with changing and improved technology. Since most all units in an educational institution are touched by information technology whether it be computing, telephones, or other services, IT can play a key role in transforming an institution into a total quality environment. This paper shares experiences of an IT organization and how it demonstrates the possibilities of TQM through customer focus and reliable, responsive services.

GUERRILLA TQM OR HOW TO INFILTRATE TQM INTO YOUR INSTITUTION

Various models for pursuing Total Quality Management (TQM) are emerging on college and university campuses. Most models and TQM gurus insist on a top down approach for TQM to succeed in transforming an organization. This paper explores the experiences in an organization in which TQM devotees pursue the principles and concepts in their own sphere of influence but without official sanction or resistance from the top.

The key principles of TQM and three models for adopting TQM in higher education are presented. The *guerrilla* model for pursuing TQM at the University of Kansas is discussed and is followed by a case study from the Department of Telecommunications.

KEY CONCEPTS OF TQM

Total Quality Management is a managerial philosophy with many names--continuous quality improvement, statistical process control, statistical quality control, among others. Regardless of the label you choose, the key concepts that underlie the *quality* philosophy include:

- focus on customers,
- focus on process,
- use of scientific method to continuously improve processes,
- employee/staff involvement.

Many other issues are involved for an organization interested in adopting TQM principles but are not discussed in this paper.

TQM places a premium on customers and recognizes their central role in determining quality. The satisfaction of an organization's customers--both those external to the organization and those within the organization--is a key driver of TQM. An understanding of who the external customers are and what they need is critical to carrying out the organization's mission.

The customers internal to an organization are partners in accomplishing the organizational mission. In particularly complex organizations like higher education institutions, many subunits of the organization serve one another and receive service from one another as internal customers. For example, the department of telecommunications exists to provide voice, data, and video communications needs within the university. The internal customers--the departments and offices to which they provide services--will determine or judge the quality of those services.

To provide quality services, TQM focuses on the activities by which we do our work--*processes*. To accomplish a goal or perform a task, the means are processes. Telecommunications has processes to provide new communication services (install a phone), to relocate existing services (to move a phone), to upgrade or expand services (to add voice mail), to provide data communication services (install local area networks), to bill for services, to train departmental and office personnel in use of various communication features (e.g., voice mail), among a host of other processes.

To improve these service-oriented processes, we use *systematic analysis*. TQM has an array of tools and techniques to help understand how processes function

and to develop alternatives to improve them. "Controlled" experiments are used to test alternatives and to evaluate the success of suggested changes. This scientific method to improve processes is also known as the Shewhart cycle, named for W.A. Shewhart who applied statistical quality control techniques to manufacturing processes in the 1930s while associated with Western Electric (AT&T) Bell Laboratories. The steps of the scientific method or plan-do-check-act (PDCA) cycle include (Sherr and Lozier, 1991, p. 8):

Plan: Identify a process in need of improvement, analyze the problems, and develop a proposal for change that will cause some type of improvement.

Do: Run an experiment with the proposed change.

Check: Collect data to determine whether the experiment produced the desired change.

Act: If the experiment is successful, implement the idea more broadly; if not, learn from the mistake and try an alternative.

Processes targeted for improvement are systematically studied using the PDCA cycle and data collected about the process is used to determine the viability of proposed changes to the process.

Who recommends process improvement changes? Those most closely associated with the process, often referred to as the "owners" of the process, are in the best position to suggest improvements. TQM recognizes the critical human element in the execution of processes and involves staff in the improvement of those processes. It is the owners of the process who best understand how a process *actually* operates. This knowledge is critical to the improvement cycle since the focus is on how a process actually works, not how someone removed from the process thinks it works. How a process could work better is the outcome of the process improvement effort.

The role of management changes from being directive to coaching as it "empowers" staff to assume greater responsibility for how their work is executed. Staff development is essential to prepare staff for these expanded responsibilities. An understanding of organizational mission, knowledge of customers served, and an understanding of tools and techniques to improve processes are a part of necessary staff development. Furthermore, a sharing of responsibility and credit for the improvement of organizational processes is an obligation of management in a TQM organization.

This synopsis of some of the key tenets of TQM provides a backdrop for the ensuing discussion for an organization adopting TQM principles and concepts. For a more extensive discussion of TQM foundations in a higher education setting read "Six Foundations of Total Quality Management" by Lozier and Teeter (1993).

ORGANIZATIONAL MODELS FOR PURSUING TQM

TQM gurus and industry leaders pursuing TQM in their own organizations insist there is only one model for adopting TQM and that is "top down." While many early adopters in higher education have heeded that advice, there is evidence of other models of pursuit. It is too early to determine their ultimate success but important to note them.

In 1991, Seymour and Collett reported the results of a survey of twenty-two institutions with a TQM initiative. They found three distinct models for

adopting TQM:

- cascade model,
- infection, and
- loose-tight or combination of top down/bottom up.

Each model is briefly described below.

The cascade model (or trickle down) involves master planning from the "top down." The senior officers of the organization study TQM principles and tools; the leadership develops a vision for the organization and a three or five-year plan for implementing TQM; education and training are provided; and pilot studies are initiated.

In the infection model (or bubble up) there is top level involvement but not necessarily commitment; the implementation takes place through voluntary pilot programs whose successes generate interest and are used to garner interest throughout the organization.

In the loose-tight model, institutional leaders need not be zealous nor have a sharply-defined five-year plan; there is some involvement at the executive level and some general map of where the journey is headed with a loosely-developed plan; local champions pursue fundamental transformation of their unit or area; the pilot projects not only focus on the improvement of a targeted process but also a basic change in the unit's culture.

The focus of this paper is on a fourth model--the guerrilla model--with attributes of both the infection and loose-tight model.

GUERRILLA MODEL FOR PURSUING TQM AT THE UNIVERSITY OF KANSAS

Interest in TQM at the University of Kansas (KU) was spurred by faculty teaching quality concepts in the School of Business. In the fall of 1988, two senior administrators in the financial area attended a five-day professional development program on TQM that the business faculty present each semester for business and industry leaders. As a result of this experience, a pilot project to improve the payroll process was undertaken that resulted in the elimination of signatures (other than the appointing department) on student appointments under $6 an hour. This reduced the complexity of the process, reduced errors, and improved timely payment for hours worked.

In May 1989, all senior administrators attended a session on the principles, concepts, tools, and techniques of TQM conducted by Lawrence A. Sherr, Chancellors Club Teaching Professor and Professor of Business Administration. Sherr conducted an expanded version of that session in 1990 first for the directors and then the staff reporting to the University Director of Information Resources (academic and administrative computing, human resources, institutional research, and telecommunications). A pilot project in telecommunications was initiated. During 1991 Sherr presented several seminars on TQM for the Unclassified Professional Staff Association.

After several years of presentations on TQM to a variety of administrators and staff, there was no top level "push" to formally adopt these principles. Changes in administrative leadership and other issues diverted the attention of the senior management of the institution.

But the grass roots involvement in the pilot payroll project and subsequent seminars for mid-level managers and staff who found the TQM principles and concepts appealing provided the real impetus for pursuing quality concepts. The guerrilla movement began to form in 1991 as individuals who shared an interest in pursuing TQM concepts in their own spheres of influence began meeting to learn more about TQM concepts and to consider how to pursue the practice of these principles in their own organizations. They referred to themselves as the Ad Hoc TQM group and included the following:

- Associate Vice Chancellor for Administration and Finance,
- the directors of facilities operations, telecommunications, and facilities planning,
- the director and assistant director of institutional research and planning, and
- associate director of human resources.

The group has expanded to include other directors committed to pursuing these concepts. The group does not formally report to any university officer; the group was not appointed by anyone and is not formally accountable to anyone.

The ad hoc group was motivated to pursue these principles by the simple desire to be more customer-friendly, provide higher quality services, and be more efficient through the adoption and practice of quality principles. This common interest in pursuing shared goals galvanized the TQM Ad Hoc Group to develop plans and devise strategies to make TQM an operating philosophy.

Initially the TQM Ad Hoc Group recognized that an investment in training was essential if the principles of TQM were to become an operating philosophy. The group sponsored training of prospective team members, team leaders, team facilitators, and team sponsors that built upon introductory sessions presented by Sherr. These first efforts were funded by members of the ad hoc committee from their departmental budgets, a real demonstration of commitment in a time of constrained resources.

The guerrilla movement advanced with the formation of six teams in 1992 to improve administrative processes. This action step signaled that the movement was beginning to realize the goals that brought together the members of the TQM Ad Hoc Group. The teams reported on their activities in March 1993 to the senior management.

While these teams worked, interest in TQM grew. Training in the principles and concepts is now conducted by members of the ad hoc group. Over 300 staff and faculty have been introduced formally to TQM principles and as the interest in TQM expanded, the effort to coordinate and support the formation of teams grew beyond the volunteer capacity of the ad hoc group. Subsequent to the presentation to the senior management about the activities of initial teams, funds were identified to support a full-time coordinator/trainer. This support has enabled the effort to grow by expanding the training and by providing assistance to units to help identify processes for improvement and to form more teams to address new issues.

The vision of the early proponents of TQM was that through championing the principles within their own organizations, their successes would capture the interest and attention of others. The strategy that developed from this vision bears a strong resemblance to those used by political movements (Goodwyn, 1978); the activities may be considered as guerrilla tactics. The five phases

to the strategy are:

- Movement Forming - Create awareness of and interest in a new managerial philosophy that recognizes that the pursuit of quality is customer focused, data driven, process oriented, and empowers faculty and staff.
- Movement Recruiting - Form an ad hoc group which shares an interest in furthering the principles espoused by the new managerial philosophy.
- Movement Educating - Educate faculty and staff about the principles, concepts, values, tools, and techniques of the new managerial philosophy.
- Movement Activated/Embraced - Create a mechanism for the pursuit of these new principles, concepts, and values utilizing the tools and techniques, e.g., teams.
- Movement Realized - Integrate these concepts, principles, and values into the daily work life of faculty and staff.

The objective of this five-phase strategy is to transform the university into a quality-driven, customer-focused institution in all aspects of the organization.

In summary, the principles and concepts of TQM are intrinsically appealing to those desiring to provide high quality services and the tools and techniques provide a means. The challenge is in the pursuit of the philosophy. Initially senior management was neither a supporter nor a barrier. The proponents took it upon themselves to pursue these principles in their spheres of influence. As experience grows and interest builds, other units of the university are targeted to "join the movement." The expansion process is slow but deliberate. With limited resources, the ad hoc group wants to be sure newcomers are adequately trained and supported. The effort is still in its infancy (ad hoc) with the hope of becoming institutionalized over time--the movement fully realized.

DEPARTMENT OF TELECOMMUNICATIONS IS INFECTED WITH TQM AND BEGINS THE TRANSFORMATION PROCESS

The Department of Telecommunications is a key player in the guerrilla TQM movement at KU for several reasons. At the time the guerrilla movement was forming, the department was undergoing considerable growing pains and suffered a variety of image problems that TQM could help address. There were many opportunities for improvement. Additionally, since all units in the university use telephones, if the telecommunications department successfully practices quality principles, it has the potential of impacting most all units in the institution and could spur interest in TQM. Furthermore, information technology (IT) organizations are accustomed to being change agents in their institutions since they constantly cope with changing and improved technology. The following case highlights how telecommunications became a part of the guerrilla TQM movement and describes the transformation process and its various impacts in the evolution of telecommunications into a quality organization.

Background

The Department of Telecommunications is one of the newest departments at the university. Once a service provided by facilities operations (physical plant), telecommunications became a department reporting to the senior officer of

information resources in 1986. During this period, the university was in the final stages of wiring the campus, installing the new PBX, providing new phone sets, and hiring staff. Processes multiplied and became extremely complex in response to human, technical, or system failures/needs/regulations.

Exasperation, frustration, and dissatisfaction multiplied in the client community. Life--in terms of telephone service--had once been simple: if a department wanted to add a phone, move a phone, or disconnect a phone, they had only to call the local Ma Bell to take care of everything. Now the client was faced with new forms to complete, people to deal with who were not telephone experts, new rules, and higher costs.

At the same time, staff in the telecommunications department were faced with hostile, frustrated clients who yelled at them; had trouble finding information in the files; did not fully understand how the PBX worked; and did not know how to pass information to one another in a meaningful way. In response to this chaos and uncertainty, the staff sought to gain some control by creating new processes, modifying old processes (sometimes combining the processes), adding new forms, and attempting to document the ever-changing procedures.

The Beginning of the Transformation...

Informal and sporadic discussions about TQM occurred between the coauthors of this paper for over a year, but interest and commitment were undeveloped until 1990 when the annual retreat of the Information Resources units was devoted to an introduction to Total Quality Management principles presented by Sherr. The telecommunications department director left that session with a commitment to explore the possibility of using TQM to evaluate some of the processes that appeared to be badly broken or in need of a "fix."

Continuing discussions between the coauthors ended with our agreement to put together a pilot TQM team in telecommunications--the first since the guerrilla movement began. We established meeting dates and times and met for several months in the fall 1990/spring 1991 before cancelling the project. Why? In short, we did not yet have the tools or training to properly deploy a team.

The meeting format was no different than the format established for a staff meeting. Too many people were involved; the staff had no idea of what we were trying to do or what their role should be; the director controlled the meeting and had specific outcomes in mind; the staff had no stake in the outcomes; and, particularly important, the staff was intimidated by the director's presence and the majority were extremely reluctant to participate. It became apparent that this process was not working. Training in leading and facilitating teams and how to approach process improvement was needed. Contrary to the advice of some to "Just Do It," we learned we did not know enough to "Just Do It."

Next Phase

The commitment to TQM, however, remained. And, fortunately, the TQM Ad Hoc Group arranged for team leader/facilitator training in January 1992. Shortly after this training, six teams formed from units represented on the TQM Ad Hoc Group. One of the teams was from the telecommunications department; the

director was the sponsor and one of the department's assistant directors was named team leader. Both had participated in the team leader/facilitator training. The team was charged to improve the telephone work order process.

Maintaining Momentum

What helped maintain and sustain the telecommunications director's interest in TQM as well as reinforcing support of the departmental team, was the regular meetings of and discussions with the TQM Ad Hoc Group. These meetings reinforced the TQM principles that:

- the university is a collection of processes;
- units are responsible for the creation and maintenance of many of its processes--they are not imposed by others;
- the unit must ask how they perform a specific process, who performs specific tasks in the process, and why;
- the unit must not only be willing to change or delete specific processes, but also to continuously evaluate the changed process to maintain gains or seek further improvements.

The critical element is that process analysis, change recommendations, change implementation, and change evaluation are conducted *by the individuals who perform the tasks in a process or who are responsible for the complete process.*

TQM Impact on Staff

The work order team was comprised of staff from accounting, purchasing, billing, customer services, operations, and management. The facilitator was from another campus department. The team scheduled weekly meetings and established attendance, format, and general behavioral guidelines. The team completed their initial task in seventeen weeks and, one year later, has regrouped to analyze the original changes and determine corollary processes that are candidates for improvements.

Three team members enthusiastically wrote an article about their experience for the *ACUTA News* (Association for College and University Telecommunications Administrators). Published in spring 1993, the article begins:

> "This year, the department had the opportunity to apply Total Quality Management techniques to the improvement of the telephone work order process. It found that by employing the TQM philosophy--by coordinating all departmental areas and drawing on the insights and talents of all staff--it was able to isolate problems and to create effective solutions."

The article ends:

> "Our most obvious benefit from our TQM process is the new work order.
>
> Another benefit is a greater sense of teamwork, as each area within our department communicated and worked with others. Through TQM, Telecommunications staff gained a greater understanding of our department and an increased appreciation of how we can pool our abilities to improve the way we do business.
>
> Another benefit was that it placed the decision-making process on the level of the users of the form--both internal and external."

From their initial team formation, through completion of the initial team

effort, and team reformation, significant, but often subtle changes have occurred. For example, the majority of the team initially expressed skepticism regarding the process and concern that their "real" work would be delayed. It took approximately four meetings before they began to work together, setting aside their work group identities (i.e., accounting, purchasing). Correspondingly, they began to look forward to their weekly meeting as an opportunity to complete work.

Team cohesiveness really took shape after an intensive three-week effort on form design. Feeling quite good about the work they had completed and the changes to be made, they looked forward to finalizing the format of the new work order. The facilitator, perhaps frustrated with the lengthy team struggle, on his own devised a format outside the meeting and presented it to the team. Reportedly, very few members commented and the meeting ended. The team informally regrouped and sent the team leader to discuss with the sponsor their reaction--demoralized, undermined, devastated, and frustrated. At the next meeting (after a one-week cool down period), the team successfully confronted the facilitator. The result: the team members drew closer, with a stronger commitment to function as a team.

Impact on Management

TQM poses many challenges to management. Management is charged with the maintenance and creation of processes and some may view the examination of processes as a challenge to their authority. Furthermore, it can be difficult to relinquish to the staff the authority and autonomy to change processes. The staff must recognize that when they have been provided with the authority and autonomy to improve processes, they also assume the responsibility for the success or failure of the processes they are empowered to change. As the boundaries for management and staff change, everyone needs to understand the implications of those changes. This is an educational process and, in some cases, a struggle for all that requires constant monitoring.

Management must recognize that not every staff member may fully understand a process even though they may be a critical player. For example, mail delivery/pickup in telecommunications has historically posed problems. To clarify the process, instructions were written and are continually modified to simplify the process. For example, a list of technical reading material with the designated recipient of each has been posted at the receptionist's desk. Yet, month after month the director's "in-box" was filled with material that should have been directed to others. Frustrated with the failure to follow guidelines, I (Jan Weller) went to the front desk and, self-righteously holding up the misdelivered magazine, asked the receptionist if she had instructions on where this magazine should be delivered. She paused and then said brightly, "Oh yes, but I thought YOU might like to see it before I sent it on to the right person." If individuals don't understand the process of which they are a part, they may, with the very best of intentions, feel free to change the process.

Impact on Customers

We do know that the customers in the external work order focus group like the changed form. A 62% reduction in call backs indicates success. The number of clients involved with this process, however, are less than 3% of the total

faculty and staff at the University.

Internal customers, however, indicate that problems exist with the new form in terms of billing and cable plant database updates. While we do have some informal feedback, telecommunications does recognize the need to systematically collect data to assess the impact of changes on all customers. In collaboration with the Office of Institutional Research and Planning, an assessment tool is being developed.

Parting Thoughts...

The principles and practices of TQM are driving the staff to become more customer focused. Staff now see themselves as clients and know what they want and how they want to be treated. The staff are beginning to practice thinking about what they would expect as a client of telecommunications. The "we" versus "them" mindset is shifting as demonstrated by conscious effort by customer services staff to view irritable clients as a challenge. Recently, the manager said she timed how long it took to turn around a client from negative to positive (or at least neutral).

The staff is looking at what we do and how we do it as a series of interconnected processes. They are asking whether they should look at a specific process and, if so, should the evaluation process be formal or "quick and clean." The degree of perceived process complexity and the time to formally study the process are the determinants. Some process issues can be addressed informally using TQM principles and tools rather than a formal team process.
Learning about and practicing TQM opens us to new ways of doing old things. At every opportunity we are asking our technical and administrative colleagues how they perform tasks, why, and the results. This, perhaps as much as anything, is what infects the staff. There is excitement that tools exist that allow us to look at old tasks in fresh, new ways--and the staff will be the ones who will assess whether a new way can or will work.

TQM is inclusive if staff is provided with basic training in the principles and practices. It is essential for management to articulate why it is important to incorporate TQM into everyday work habits and visibly practice tenets of TQM. Staff who are not trained in TQM basics, or who have not had the opportunity to develop a TQM mindset through participation on a team or other reinforcing activities, can inadvertently subvert a unit's pursuits of being a quality organization.

TQM is not a panacea and it is not easy to practice. To learn new ways of thinking and doing can be daunting, and it may seem easier to return to the old way of doing business. But doing things the old way is what drew us to TQM.

References

ACUTA News, Association for College and University Telecommunications Administrators, April 1993.

Goodwyn, L. *The Populist Movement*. New York: Oxford University Press, 1978.

Lozier, G.G. and Teeter, D.J. "Six Foundations of Total Quality Management." In D.J. Teeter and G.G. Lozier (eds.), *Pursuit of Quality in Higher Education: Case Studies in Total Quality Management*. New Directions for Institutional Research, no. 78. San Francisco: Jossey-Bass, 1993.

Seymour, D. and Collett, C. *Total Quality Management in Higher Education: A Critical Assessment*. Methuen, MA: GOAL/QPC, 1991.

Sherr, L.A. and Lozier, G.G. "Total Quality Management in Higher Education." In L.A. Sherr and D.J. Teeter (eds.), *Total Quality Management in Higher Education*. New Directions for Institutional Research, no. 71. San Francisco: Jossey-Bass, 1991.

Change in the Trenches: Continuous Improvement of Service Processes

Connie Towler and Douglas Renick
Harvard Quality Process
Harvard University
CAUSE '93

Getting Started

Over the past three years of implementation of the Quality Process at Harvard, we have worked with managers, supervisors, and staff to help them understand the concepts of TQM and how they might apply at Harvard University. This work began in the Office for Information Technology (OIT). We have put some of these techniques in place and many problem-solving teams have been successful in improving productivity and saving money. We were, therefore, able to build on this positive experience with key managers in the organization and when we were ready to launch the Service Process Improvement program, they were ready and, in some cases, waiting for this next step in our journey.

OIT began its training component of implementing TQM with a basic Problem-Solving Team Training module. Added to that in the following year and one-half were Team Facilitator training, a workshop on Benchmarking, and Customer Service training modules for front-line staff and management staff.

A module on Service Process Improvement was added this year. This module, unlike Problem-Solving Team Training, was designed using the concepts of JIT (just-in-time) training, with participants working on actual processes. While the pilot program was quite successful in that the teams achieved their goals or at least made significant progress, they were mixed in their reactions to the training. Some felt that the pressure to meet the training deadlines inhibited them in their process improvement. They wanted to work at their own pace and not on a schedule determined by others. Accustomed as we are to listening and reacting to our customers, we re-designed the program to use a case study in the training instead of actual processes.

What is Service Process Improvement (SPI)?

We chose the title ***Service*** Process Improvement to emphasize that all processes are driven by customer satisfaction. In this program, teams of participants learn how to identify and specify their customers' requirements and how to determine their customers' current satisfaction levels. These customers might be internal or external to the organization.

The program emphasizes Service ***Process*** Improvement which requires that participants understand:

1. processes and systems,
2. how to map or flow chart them, and
3. how to measure them, not only in terms of results, but in terms of key variables upstream from the results.

The aim is to teach people to build quality into the process, which eliminates the waste of rework and ensures customer satisfaction.

Problem-Solving Team Training, a Pre-Requisite:

Problem solving is a part of any service process improvement. So that we would not have to stop the flow of the work once we started the process improvement, we made Problem-Solving Team Training a pre-requisite to the Service Process Improvement Training.

Included in the Problem-Solving Team Training is practical application, using a case study, of many of the basic quality tools: Pareto, histogram, fishbone, force field analysis, et cetera. Some refresher might be necessary as we proceed with process improvement, depending on the length of time between these two programs; however, it would probably not be a significant delay. By making the Problem-Solving Team Training a pre-requisite, we also avoided a facilitator's worst nightmare of some people having had the basic training and some not.

Objectives of the Program

As teams began the Service Process Improvement program we established these objectives:

> At the end of the program the teams will have **improved the effectiveness of one work process** and will have learned:
>
> - how to identify and flowchart a work process
> - how to measure the capability of a process
> - how to obtain customer requirements and to specify them
> - how to measure customer satisfaction with the output of the process being improved
> - the interaction between service process improvement and problem solving
> - re-learned the importance of creating a high performance team through the use of good interaction skills, good meeting management, and consensus decision making

Selecting a Team to Attend SPI Training

To help managers understand the selection process they might use to identify who should attend this program, we formed the following guidelines.

Guidelines for Selecting a Team to Attend SPI Training

> The team you select to send to SPI training might be formed by thinking about the work process or processes you would like to see improved. You want to include people on the team who have ownership and responsibility for a process or set of processes. You are empowering your team to document a process, to discover how it does and does not work, and you are asking them to interact with the customer to discover the customer's current level of satisfaction and exact requirements for the output of the process. The team should have 4 to 6 members.
>
> We are using as a resource for the training a booklet by Richard Chang called "Continuous Process Improvement." Chang suggests the following guidelines for choosing a work process to improve.
>
> "Processes selected for improvement should typically be considered critical to the organization, ones where customers are not satisfied with the specific outputs

being produced. In addition, some or all of the following process characteristics may exist:

- Problems experienced by external and/or internal customers
- Complaints received from external or internal customers
- High degree of non-value-added effort involved
- High maintenance costs, i.e., too complex, too many people and/or functional areas involved, requires ongoing fixes
- More advanced technology available than is currently being used.

"To increase your chance of success, select a process that:

- The customer benefits from or cares about
- You have a moderate to high level of control over
- Is important to the ongoing performance of the organization
- Is stable enough to analyze, measure, and improve.

"In addition, the organization should be able to dedicate the appropriate financial, and human resources for improving this process."

You may want to pick a team that has responsibility for several processes, giving them the freedom to choose a process to improve. We will be asking teams to keep managers informed at each step of the improvement process.

The Original Design

This module began using the concepts of JIT (just in time) training, with participants working on actual processes from their every-day work life. The teams were given time between sessions to collect data from the customer and perform tasks required in the process improvement effort. The outline for our first training effort looked like this:

First Design
Service Process Improvement Training
Program Schedule

Session 1 - 3 hours

Overview of SPI
Review of processes chosen by teams
Methods for identifying customer requirements and levels of satisfaction
Teams plan to identify requirements and satisfaction level

2-1/2 week break

Session II - 4 hours

Reports on requirements and satisfaction
Flowcharting the process
Identifying strengths and weaknesses of the process
Measuring the process

2-1/2 week break

Session III - 2 hours

Reports on process performance
Review of process improvement methods and problem-solving process
Write "as is" and "desired state" statements

3 week break

Session IV - 2 hours

Reports on solution(s) chosen, plans for implementing, tracking, and evaluating
Standardizing the improved process

4 week break

Session V - 2 hours

Reports on results and standardization
Evaluation of training and application process
Graduation and celebration

Program Re-Design

The re-designed program is different in the following ways:

- Length of the program:

  The program is shorter: the training time for the original program was 12 hours, the new program is 9 hours. The program is also more compact: instead of spreading the program out over 12 weeks, we now conduct SPI training in 3 half days spread over 3 weeks.

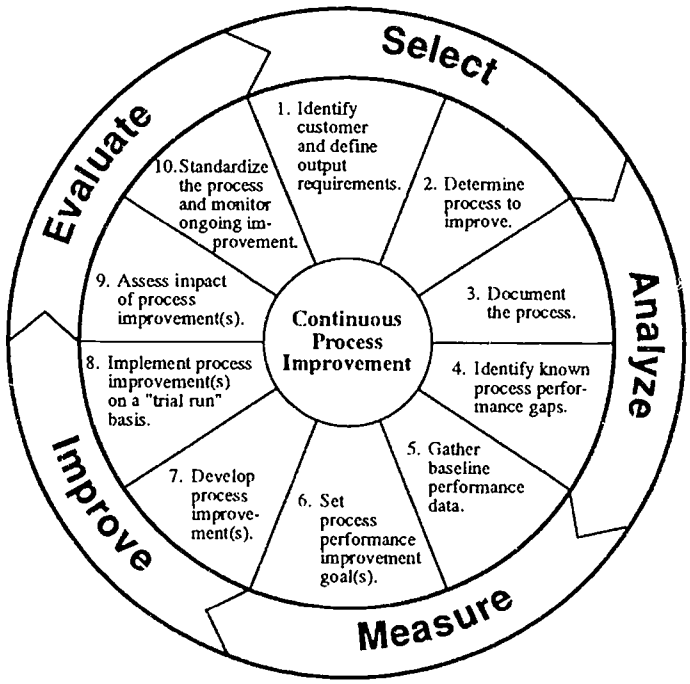
- Approach

  We are using a case study in the re-designed program, not real processes. This, again, was a request from the participants in our pilot training program.

Re-Design
Service Process Improvement
Session I - 3 1/2 hours

- Conceptual overview of SPI

  SAMIE (Select, Analyze, Measure, Improve Evaluate) model and its relation to Problem Solving



- Program Objectives
- Simulation of Process Improvement
  - Collating Exercise
- Review of key concepts and practices in TQM as they relate to SPI
- Introduction to the case study
- Team Meeting 1 SAMIE steps 1 and 2
  - Case--Part 1: the organization or department, its context and situation, services provided, customers and identification of something about the process that will be the focus of the case.
- Questions for the team:
  - Who is the customer(s)?
  - What is the service provided?

- What is the process to be improved, its boundaries--inputs and outputs?
- Who is the process owner? Is this a process that is a candidate for the SPI approach? How will you find out the customer's requirements and the current level of customer satisfaction?
- Clarify concepts covered:
    1. a process--its owner, its boundaries
    2. SPI and processes
    3. services provided
    4. customer(s)

• Presentation on how one identifies customer requirements, and how these are translated into the work unit's specifications. Measuring customer satisfaction. The basics of flow charting.

• Team meeting 2 SAMIE steps 1, 3 and 4
- Case--Part 2: Results of interviews with customers and a focus group. Results of a customer survey.

• Questions:
- What are the customer's requirements? Work unit specifications?
- What is the current level of satisfaction? Is it worth improving the process?
- Case--Part 3: The current process in narrative form. Task for the team:
    - Create a flow chart of the process.
    - Identify any known performance gaps (ways in which customers requirements are not being met.)

• Clarify concepts: - customer requirements and departmental specifications
- Flow charting

• Assignment on measuring

Session II - 3 1/2 hours

• Team Meeting 3 SAMIE step 5

• Questions:
- How will you measure output?
- What measures could you create upstream in the process?

• Presentation: Brief review of the problem solving process.

• Team Meeting 4 SAMIE step 6
- Case--Part 4: Baseline performance data.

• Questions: Are there performance gaps? If so, where in the process?

• Tasks: List performance gaps. Choose one gap as a problem to be solved. Write AS IS and Desired State statements. Fishbone potential causes of the problem.

• Team Meeting 5 SAMIE step 7
- Case--Part 5: Data on causes.

• Question: What are the key causes of the problem?

• Team Meeting 6 SAMIE step 7
- Tasks: Brainstorm solutions. Clarify. Support and disagree. Bracket. Combine. Name the categories of solutions.

Session III - 2 hours

- Team Meeting 7 SAMIE step 7
  - Tasks: Select a solution using weighted voting, discussion and consensus.
    Plan the solution using a Gantt chart.
    Talk through implementation and evaluation--SAMIE steps 8 and 9
- Talk about standardization--SAMIE step 10.
- Summary and review of participant objectives.
- Teams meet with facilitators to plan their first meeting.

Results and Learnings

It has been stated above that the JIT aspect of the first SPI program did not work well. Teams working on improving real processes proceeded at very different paces. The training became a distraction. Teams felt that they had to stop the real work on the process to complete the training assignment for the next session.

Yet one team, the telecommunications team, felt that being asked to measure their process was the real breakthrough that took them beyond the problem-solving process. They never would have measured - because they felt that they knew what was wrong! The results of the measurements confirmed their hunches, but gave them confidence that they were on the right track. This team also felt that the constant emphasis on listening to the customer and applying what you hear leads to looking at things in a different way. The team began by asking: "how can we improve our voice mail system." With this focus the team bogged down. Listening to the customer opened up the possibility of doing away with the voice mail system except when it was needed to handle the large volume of calls in peak periods.

A team looking at the return policy at the Technology Product Center discovered that process improvement didn't apply very well to a discreet policy decision. Another team working on the maintenance system for the local area network in a particular building did not have a process in place on which to make improvements. The process had to be created. The training was only somewhat useful for this team. The learning here is: be careful to apply process improvement only where it is applicable and helpful.

One facilitator reported:

> "The quality process helped drill in the notion that we must be customer focused both individually and organizationally. SPI goes a step further and helps us to view what we do for customers--almost everything we do--as a process.
>
> "Recognizing that what we do in serving customers is a process is a very powerful new awareness. Reifying the process, making it an object that can be measured, studied, tweaked and gradually improved as we measure it, brings us to a very different mindset and into a very different relationship with our work. SPI gives us the knowledge along with the right tools for controlling our work processes, for changing them in ways that make a difference to us and to the customer. We become the subject instead of the object ... the actor instead of the acted upon.
>
> "Whereas before the job seemed like random human interactions that cannot be controlled or improved, SPI helps us to see it differently, to understand the inputs and the outputs, to focus on the process variation and to measure the resulting gulf between what we are providing and what the customer is really telling us she wants.

"As a facilitator, though, I learned that success does not come easily. If the suggested improvements were not made, participants ended up believing that SPI was just another training effort that in the final analysis could not make the bureaucracy adopt needed changes."

Teams reported that it was an obstacle to effective work together that some team members had been trained in problem solving, team interaction skills, and good meeting management practices -- and some had not. This situation makes it difficult for the facilitator and for the team members. And it takes more time for teams to make progress.

Summary

In summary:

- Process improvement training can have significant payoffs for teams and the organization.
- Use a case study of process improvement for the training.
- Flow charting and measuring processes are valuable skills for teams.
- Help managers choose appropriate processes to work on.
- Be sure that all team members have been trained in problem solving, team interaction, and meeting management skills.
- Be sure managers understand the importance of empowering teams to make improvements.

Primary Resources for Service Process Improvement

Leadership Through Quality: Concepts of Quality, Quality Improvement Process Reference Manual. Stamford, Connecticut: Xerox Corporation, 188.

Chang, Richard, "Continuous Process Improvement," in *InfoLine*. (Issue 9210). Alexandria, Va.: American Society for Training and Development, 1992.

Process Quality Management and Improvement Guidelines. Indianapolis, Indiana: AT&T Customer Information Center, 1987.

Harrington, H.J., Business Process Improvement. New York: McGraw-Hill, Inc., 1991.

Establishing Trust and Building Relationships: Negotiating with Information Technology

by

Scott C. Ratzan MD, MPA, MA
Assistant Professor
Division of Communication Studies
Emerson College
100 Beacon Street
Boston, MA 02116

617/578-8745

Establishing Trust and Building Relationships: Negotiating with Information Technology

The growth of information technology has advanced McLuhan's "global village" into a global community capable of communicating efficiently and rapidly across a large and heterogeneous landscape. Computers have played an tremendous role in transforming the life of citizens all over the world, as millions of people in more than 100 countries have been affected by the way in which they communicate, learn, govern, manage and make decisions (Stefanik, 1993).

Once universally available and operational, computer networks and other types of information technology: 800 numbers, videotapes, cellular phones, fax machines, electronic databases, cable and satellite television, talk radio/television, have minimized language and geographic barriers while providing the world's residents with the tools to learn from and communicate with each other (Stefanik, 1993).

Yet even without universal accessibility and utilization, information technology has a great capacity to serve as a catalyst for problem-solving and facilitating effective group communication (Stefanik, 1993). In fact, as leaders consider the use of total quality management techniques, the use of computers and other forms of information technology for qualitative and quantitative analysis require ethical and practical considerations on their respective utilization.

The challenges of management and decision making in today's political, business, health and education sectors demand dynamic negotiation perspectives. As such, effective negotiation and shared decision-making is built upon a communication foundation; as it is basciallly the *sine qua non* of negotiation. The entire process originates with the initial communication act. As the negotiation develops, options are presented and discussed, along with appropriate alternatives, all within a context of mutually agreed upon objective standards which imbue the process with trust, in the joint effort to reach a satisfactory and successful outcome while developing an effective ongoing relationship. Such an approach [shared decision-making] has been described in a*New York Times* (1992) editorial regarding its implication in health care as "something big -- big enough to change the way U.S. medicine is practiced."

The intent of this article is to apply a shared decision-making model to satisfy common goals and objectives, employing information technology to build relationships and establish trust between individuals in management and decision-making capacities.

The COAST Model

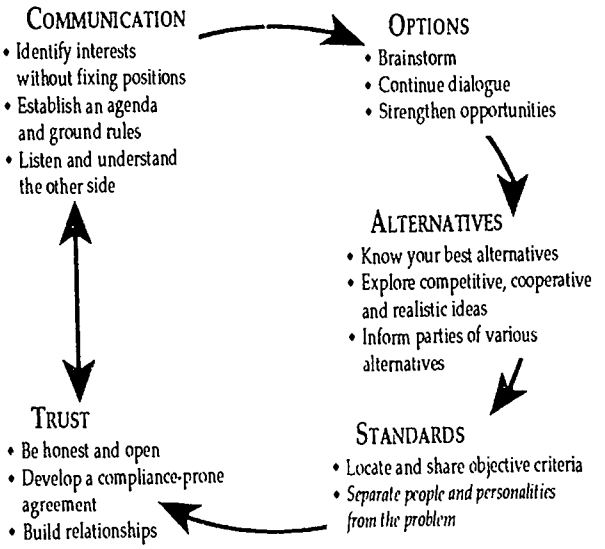
COAST -- Communication, Options, Alternative Standards, Trust -- is an approach to negotiation rooted in the essential elements of Aristotle's classical views on rhetoric. Accordingly, negotiation is a communication process based on shared similarities derived from ethical and caring relationships between people in pursuit of common goals. Its goal of identifying differences and facilitating optimal solutions among the alternatives is designed to forge an overall trusting relationship among participants necessary for long-term success in future encounters.

COAST suggests a dialectical approach, characterized by a replacement of the unidirectional flow and power relationship with a co-active encounter based on trust and a free flow of information employing various means of information exchange (e.g. interactive video, printed matter, computerized interactive databases, etc.).

To build upon agreed alternatives and enhanced trust, while advocating and implementing specific actions to improve the public good, is the abiding ethical goal of communication in the COAST model.

The COAST Model of Negotiation for management is rooted in an ethical and effective co-active *communication* process. The initial communication encounter involves parties who communicate particular management/decision-making interests. Subsequently, the brainstorming of all available *options* - regardless of the viability and effectiveness of those options. Following this important phase and through intensive communication acts, the focus of the encounter is to identify *alternatives*, agreed upon by involved parties, that could/should be employed in reaching a common goal. Such alternatives are selected based on application and analysis of the options to specific *standards*, objective criteria oftentimes defined by a third party, group or organization which has credibility among those involved in the encounter. The essential element of the COAST model, and an element that should be pervasive throughout its various phases is *trust*, the transactional product of open and honest sharing of information and credible, expected feedback among the involved parties. The degree to which trust exists within the encounter is a positive prediction of the degree of compliance with action plans and overall satisfaction of the parties involved.

Insert COAST Shared Decision-Making Model Card Here



Communication

Effective and ethical communication are key ingredients of any communication encounter. These include: *exploring/discussing the interests of each party; exercising effective listening skills; understanding personalities, cultures, backgrounds, attitudes, values, and beliefs; establishing an agreed upon agenda; setting ground rules; and asking/answering pertinent questions.*

Communication is information technology's main application with traditional and alternative means such as telephones, fax machines, computer networks among others which are designed to assist people in communicating and learning. Modern economic systems cannot compete in the global village without far-reaching telecommunications and global knowledge banks (Stefanik, 1993).

The opportunities that information technology holds for management and decision-making within the public or private sectors are great. One of information technology's greatest assets is its high reach: that it provides educators with the invaluable potential for reaching an enormous population, otherwise difficult to contact.(Arkin, 1990) This view is echoed in another researcher's report:

> "Worldwide, everyone is potentially connectable to everyone else through a newly evolved global web of interlinked telephone-computer networks. In theory at least, more abundant information communications technologies ... should create new opportunities for previously disconnected people ... to talk to each other, (and) exchange information."(Annis,1992, pp. 587-8)

Effective communication is based on the traditional communication act. Beginning with the foundation of oral and written communication , appropriate mediated communication can redefine the encounter especially if the ultimate joint decision opportunity is terminated due to issues of power and authority. Clearly, a communication-based negotiation model centers on a multi-agent decision -making approach determining the respective parties' best interests rather than employing a unilateral "substituted judgment" decision.

Although the agenda and ground rules often are impacted by constraints, the imprecision of language, the inability of parties to communicate clearly, prior attitudes, beliefs, role expectations and religious perspectives, those involved in a negotiation must make the effort to establish an open relationship, thereby ushering in a two-way process of decision-making (Quill, 1983). Without an active agenda objective within the communication component of COAST, participants in effective management decisiion-making will reduce the potential for the most effective outcome.

Options

Options, the second component of the COAST negotiation model, invites the participants to engage in generating and brainstorming potential solutions that could satisfy each parties' interests. At the onset, however, there must be an understanding that while working together, the first effort should be to create as many options as possible, without criticism or analysis of said options (Fisher & Ury, 1981). The advantage of the brainstorming process is that it provides a wider variety of options to be considered by the decision-makers that may not be considered in normal discussion. Furthermore, it results in a strong bond and identification with the decision, a product of a joint decision-making effort (Ballard-Reisch, 1990). It also increases satisfaction in the decision making process and hence increases compliance (Beisecker, 1990). The ability to generate ideas through dialogue, with inductive and deductive thinking of possibilities further enhances outcome potential. Inductive empiricist Francis Bacon (1625) stated, "a wise man makes more opportunity than he finds."

Options ahould be determined with the final goals of management and decision-making in mind, regardless of the viability of those options. Options tend to be more intangible, based for example, on values and beliefs. The options stage can be problematic if the step is not merely viewed as continuing the dialogue of parties' possible actions to address common interests. The risk is that parties often cannot withold judgment that often leads to participant withdrawal in the decision-making process.
The impersonal nature of information technology, demands an emphasis for the creation of options by all parties to be discussed and considered in the alternatives phase of the COAST model.

For example, in the field of medicine information technology assists in numerous ways; alternatives to oral communication between patient and physician such as artificial neural network systems, can be used to increase the number of options from which a physician and patient may choose. These systems analyze patterns in large data sets

where computational answers are not useful but where decision making and problems solving can be enhanced by recognizing recurring patterns (Rootenberg, 1992).

Alternatives

During the decision-making process, one should protect against hasty selection of an inappropriate course of action. Alternatives, unlike options, tend to be more tangible, leading to actual actions that could realistically address the interests. Each party should consider viable and workable alternatives in the effort to strengthen the satisfaction level of those involved in the communication encounter. Frequently this includes conferral with family members, friends and colleagues. The major focus in this part of the negotiation is to continue the dynamic flow of communication among the dyadic participants. Employing traditional formal and informal decision analysis (probabilities, reasoning, heuristics, etc.) with frank discussion of advantages and disadvantages regarding each alternative can aid involved parties in eliminating weak alternatives and strengthening the ultimate appropriate decision.

During the negotiation procedure/act, it should be reaffirmed that there is nothing permanent nor obligatory in the communication encounter. If either party views the encounter from an unsatisfactory perspective, barring a resolution of the differences, potential termination of the relationship remains, however unpleasant, an alternative which could increase ultimate compliance with realistic/rational decisions.

Standards

Another pertinent component of the COAST model is *standards*, criteria by which alternatives are measured and assessed. The agreement on and use of standards -- objective criteria -- in the decision-making process is a crucial component in enhancing the efficacy of the communication encounter.

Over two thousand years ago, the Greeks identified a speaker's character to be of crucial importance in effective leadership. Today, amidst an array of technological capabilities that can instantaneously transmit an image throughout the globe, the bottom line for the effective leadership still remains unchanged -- credibility of the source and his or her ability to establish standards and to embody personally such principles in management.

Of course, there is room for compromise and acceptance; the key step is for both perspectives to be communicated and agreed upon. For example, a patient might be motivated by moral standards and a physician by professional standards. According to Fisher and Ury (1981), each must realize that "one standard of legitimacy does not preclude the existence of others." The consideration of appropriate standards from which to refer in the mediated encounters of great importance.

The realization of the importance of information technology in the immediate future in the United States and the world has led to partnerships in business and education. In 1989, the University of North Carolina at Chapel Hill and IBM entered into a partnership to further develop information technology development in higher education. The Institute for Academic Technology (IAT) was designed to create more technological classroom experiences and streamlined the ability to disseminate information within the university system. The program was estimated to have reached 40,000 academics this year through a system of shared seminars, workshops and planning sessions. Similar partnerships are appearing in the medical school community as well.

Eighteen schools in the United States and Canada participate in the Health Care Interactive Videodisc Consortium formed in 1987 in conjunction with IBM, which allows members to collectively produce course-ware for the field (Rootenberg, 1992) The Northeast Medical School Consortium's 11 participating schools shares resources via Apple Computer technology, and the Shared Decision-Making Foundation program at Dartmouth Medical School in Hanover, N.H. in conjunction with the Sony Corporation of America helps patients make more informed decisions about their own health through a totally mediated communication process with alternatives and standards (Rootenberg, 1992).

In the medical field, the growing number of collaborations between academic institutions and information technology vendors demands guidelines due to the fact that these partnerships often affect patient care (Rootenberg, 1992). The Integrated Academic Information Management Systems at the National Library of Medicine assists institutions that wish to participate in studying several different aspects of information technology management systems in the hope of meeting such standards (Rootenberg 1992).

Trust

Trust, one of the most important elements in the COAST Model, is a reciprocally enhanced product of each of the aforementioned areas. As open *communication* is encouraged, all possible *options* and *alternatives* discussed, and objective *standards*

agreed upon, both parties already have begun the process of establishing an abiding *trust* - building an effective relationship in the dyadic encounter. Communication with disclosure of information further enhances *trust* and the relationship - (that part between the communication and trust in the model illustrated by a double arrow line).

The COAST paradigmatic dialectical design presents an added by product of disclosure by interested parties further deepening trust. If present, trust imbues the encounter with honest and open dialogue. (Bromberg, 1981; Deutsch, 1958; Jabusch & Littlejohn, 1990; Kremenyuk, 1991; Lindzey & Aronson, 1969).

With ethical communication leading to trust of both the source and the message, computer networks can enable the world's residents to communicate with and learn from each other (Stefanik, 1993). Information technology provides management and decision-makers with the tools to streamline the process and facilitate better decisions leading to more effective government, more productive business, and better-quality service. Because of its growing ubiquity, those involved in the use of information technology must ensure the accuracy and integrity of their data by properly learning how to use and manage the technology (Rootenberg, 1992)

Ultimately, the COAST model is merely a theory which builds *trust*, a necessary objective for its practical and efficient application transcending the initial encounter. Relationships are formed over time with trust built from disclosure and effective communication between parties (Silvestri, 1987). The relationship - the double arrow- is perhaps the unquantifiable resource employing the COAST negotiation model. With a strong relationship (communication and trust), future outcome efficacy of the management encounter is enhanced, adding positive human factors which often are the most important indicator to a plan's success (Fisher & Brown; 1988; Norfolk, 1990).

Applying COAST

Ironically, the importance of communication with information technology was so eloquently descibed some 65 years ago by John Dewey (1927):

> " The highest and most difficult kind of inquiry and a subtle, delicate, vivid and responsive art of communication must take possession of the physical machinery of transmission and circulation and breath life into it. When the machine age has thus perfected its machinery it will be a means of life and not its despotic master. Democracy will come into its own, for democracy is a name for life of free and enriching communion."

The idea of using all the available means of communicating -- appropriate media -- elicits unique options to expand the effectiveness of the encounter. However, the open

communication of alternatives offers individuals the opportunity to apply different standards, whether scientific, religious beliefs or other areas deemed important to participants in the encounter. In place of the traditional communication patterns, the negotiation model clearly emphasizes the joint importance of mutual decision-making of target audiences through ethical and effective management and decision-making.

Overall, the application of COAST - Communication, Options, Alternatives, Standards and Trust - to messages with information technology can result in a win-win situation for all parties involved. The ethical application of the COAST model of negotiation by business leaders, health care educators, politicians, and such could potentiate the plight for appropriate social responses including individual behavior and attitude change as well as institutional and policy making to reach appropriate audiences adequately .

Within any communication and negotiation encounter, information must flow both ways. Rather than solely expounding information technology into a community and expecting the recipients to listen, understand and adopt the message, educators should take a more transactional, holistic approach. As the use of information technology grows and new generations learn to improve upon it; as its accessibility changes the face of politics and the mass media by giving individuals more access to information; as it encourages people to organize and become active within their respective communities and presents the option to learn about and communicate with other countries, we learn that other cultures have much to teach about managing, conflict resolution, negotiation and compromise (Stefanik, 1993).

The application of COAST --Communication, Options, Alternatives, Standards and Trust -- to messages with information technology synergistically enhance the information technology/interpersonal encounter with an advantageous by-product of a *relationship* with the message/messenger. Hence, a sense of public and private empowerment to be involved and responsible participants in attaining the goal becomes a welcomed qualitative benefit, resulting in a win-win situation for all parties involved.

REFERENCES

Annis, S. (1992). Evolving connectedness among environmental groups and grassroots organizations in protected areas of Central America. World Development, 20, 587-95.

Arkin, E.B. (1990). Opportunities for improving the nation's health through collaboration with the mass media. Public Health Reports, 105(3), 219-23.

Ballard-Reisch, D.S. (1990). A model of participative decision making for physician-patient interaction. Health Communication, 2(2), 91-104.

Beisecker, A.E. (1990). Patient power in doctor-patient communication: What do we know? Health Communication, 2(2), 105-122.

Bromberg, M.D. (1981). The new medical-industrial complex. New England Journal of Medicine, 304, 233.

Charting the Right Direction (1993). Advertising Supplement to *The Chronicle of Higher Education.*

Deutsch, M. (1958). Trust and suspicion. Journal of Conflict Resolution, 11, 265-279.

Fisher, R. & Brown, S. (1988). Getting together: Building a relationship that gets to yes. Boston: Houghton Mifflin.

Fisher, R. & Ury, W. (1981). Getting to yes. New York: Penguin Group.

Jabusch, D.M. & Littlejohn, S.W. (1990). Elements of speech communication. San Diego: Collegiate Press.

Kremenyuk, V.A. (Ed.). (1991). International negotiation: Analysis, approaches, issues. San Francisco: Jossey - Bass Publishers.

Linzey, G. & Aronson, E. (1989). The handbook of social psychology: Volume 5. (2nd ed.). Reading, MA: Addison-Wesley.

Mccombs, M.E., & Shaw, D.L. (1972). The agenda setting function of mass media. Public Opinion Quarterly Summer:36.

Norfolk, D. (1990). Think well, feel great. London: Michael Joseph.]

Quill, T.W. (1983). Partnerships in patient care: A contractual approach. Annals of Internal Medicine, 98, 228-234.

Rootenberg, John D. (1992). Information Technologies in U.S. Medical Schools. Journal of the American Medical Association 268, 21, 3106-3107.

Stefanik, Nancy. (1993). Sustainable Dialogue/Sustainable Development . Brecher, Jeremy, John Brown Childs and Jill Cutler (ed.) Global Visions: Beyond the New World Order. Boston: South End Press, 1993.

Weinstein, Michael M. "Dr. Video: How to Best Decide What Patients Need." *New York Times*, December 13, 1992.

Assessing the Effectiveness of Information Technology

Susan F. Stager, James G. Williams, Polley Ann McClure, and John W. Smith

I. Evaluation Needs of Chief Information Officers

The time is past when information technology leaders could boldly promise that our solutions would cure all ills (if indeed there ever was such a time). Today college and university executives are more knowledgeable about the real promise of information technology; at the same time they are under serious budgetary pressure. This means it is increasingly important for us to be able to document authoritatively the outcomes of investments in information technology. We must also demonstrate that the benefits of those investments outweigh their costs. In the past we've done neither job well.

The benefits of information technology interventions should be evaluated in terms of their effects on the academic process. These can be direct effects, such as interventions aimed at improving student learning or enabling an analysis needed for important research. The effects can also be indirect, as when an intervention is intended to improve the efficiency of a support process or to improve some element of service to customers. It is important to tie indirect effects to academic outcomes, as, for example, when an improvement in efficiency of a support process enables funds or staff to be reallocated to the direct support of instruction and research. The most important failure, in my view, of efforts to evaluate information technology projects is that we evaluate the technology itself and whether people like it or use it, but we do not often enough take the next step—demonstrating that the project made a difference in academic outcomes for the institution.

Nor is it enough simply to demonstrate effectiveness in terms of academic outcomes. For example, a growing number of excellent software packages have been shown to improve some aspect of student learning. I know of no example, however, in which we have measured an improvement in academic outcomes per unit of cost of these packages. An important first step may be the collection of very expensive "boutique" applications that show some improvement in learning; but when our institutions look to us for help in improving productivity, we must begin to include the denominator of cost in our assessments.

A critical requirement for the type of assessment I think we need is some definition of measures of academic outcomes. Unfortunately we are dependent upon others—faculty and academic leaders—for these definitions. They will not be easily formulated. But until we have some agreement about the way to measure the numerator of the productivity term, our efforts to do so will always be subject to disagreement by way of definition. We need to challenge faculty and academic leaders to define the objectives for educational improvement initiatives in terms that can be measured. Then we need to use those definitions to assess the effectiveness of technology initiatives.

These efforts to evaluate the "bottom line" effectiveness of our activities are essential if information technology is to "grow up" and become a mature component of the higher education enterprise. Five or ten years ago we may have been able to promise the world at any cost (and many of us did). But today the very existence of our institutions may depend on whether or not we can deliver on the promise of improving academic productivity. Our obligation is to determine honestly where we can and can't do this, and to give evidence to support our case.

The purpose of this presentation is to review Program Evaluation techniques for acquiring data about technology innovations and to provide recommendations on the format for communicating these data to senior administrators in higher education and to state legislators.

II. Evaluation Tools and Techniques

> "When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind: it may be in the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of science."
>
> William Thomson, Lord Kelvin
> Popular Lectures and Addresses (1891-1894)

The question any evaluation seeks to answer is: "Did the intervention or treatment have the desired effect? Did it cause a change? Did it make a difference?"

To begin an evaluation it is critical to determine exactly what was the intervention or treatment in question. Following that, we seek information on the effect. We must determine what the effect was and find ways to measure it. Even if an evaluation proceeds no further than this, thinking about these key points will lead to more effective programs and investments.

To provide some context, consider the following hypothetical intervention. In the past year, your university has invested $1 million in constructing a new network. You wonder if this has been a good investment. The questions above—about intervention, effect, and measurement—become real, and their difficulty becomes clear. What do you mean by "good investment?" How do you measure the effect of a $1 million investment versus, say, a $900,000 investment? To make any progress we will have to make this scenario more concrete.

Imagine that in the past year 1,200 faculty at your university have been connected to the network at a cost of $1 million. Has this intervention (the network connections) had any effect on:

- Faculty productivity as measured by the number of peer-reviewed articles they have published?
- Faculty satisfaction with the computing environment at the university?
- Faculty access to information (as measured by what?)?

To answer these questions we need information derived by either an experiment or an evaluation. We make the following distinction between these two sources of information:

- To control unknown effects (sources of variation), an experiment uses sampling, experimental design, and random assignment of subjects to treatments.
- An evaluation is usually non-random. Without the powerful effect of randomization, great care must be taken to attempt to control unknown sources of variation that could mask or exaggerate the intervention and lead to incorrect conclusions.

The primary issue in both experiments and evaluations is the appropriate assignment of unknown sources of variation.

To continue the case study, in an experiment you would assign faculty members at your university to a networked or non-networked group. After a reasonable period of time, you would compare the publication rates in the two groups. If the productivity of the networked group was significantly higher than that of the non-networked group, you might conclude that the intervention had had a positive effect. There might be other factors, such as department/discipline affiliation or length of tenure at the university, that could have an effect on number of publications. But, in theory, the random assignment of faculty to groups balances these effects.

In an evaluation you would have made no random assignment of faculty to groups. Ideally, faculty members would have been given connections to the network based on some non-random plan. To conduct an evaluation of productivity in this context, you would have to take into account all variables that could have affected the outcome of the experiment. For example, it would be reasonable to assume that faculty who demanded network connections immediately were more involved than others in collaborative research with their colleagues at other institutions. Such faculty might be inclined to publish more as a group, whether or not they had network connections. The key thought to keep in mind is that in an evaluation context, randomization does not control for these effects. The evaluation must take them into account, or risk drawing incorrect conclusions.

Given the non-random nature of evaluative studies, and understanding that uncontrolled, non-random variation is the chief source of ambiguity in evaluation results, what might we do to minimize the risk of error? Here are several possibilities:

- We can give careful thought to the types of variables that can cause unwanted effects. If we collect information on these variables, we can use statistical procedures to eliminate them from the analysis.
- We can analyze information collected on the key variable of interest before and after the intervention. But we must beware of the time effect—a variable can change simply through the passage of time, and not from the intervention.
- We can collect data on a similar, parallel group not exposed to the intervention. In effect, a control group can be identified and examined after the fact.

Revisiting our case study once more, we can apply some of these ideas. We know, for example, that it is standard university procedure to collect publication rates for faculty each year. We can examine the overall rates of publication from the year prior to the network installation and one year after installation. A significant difference would indicate the effect of our intervention, all other factors being equal.

In a similar evaluative procedure we could examine and compare to our own case information collected from a university that had not deployed a network in the past year, ensuring that the demographic characteristics of the two groups are matched as closely as possible (size and mission of the institution, departmental affiliations of the faculty studied, their years in the department, etc.).

III. Criteria for Evaluating Evaluations

Whether the intended audience for your evaluation is the faculty, other administrators, the university president, or board of trustees, that audience will make value judgements about the worth of the evaluation. The criteria for judging an evaluation study are the same criteria that administrators use daily when judging the value of any information presented to them: the

information must be of high quality and must come from reliable sources. These evaluation criteria may never be verbalized, but they will be consciously applied. It is important, therefore, that you "think" like the audience and evaluate your own efforts during the planning stage when you can still modify the evaluation process.

Here is a list of criteria that represents both the professional and lay judgement of such efforts:

1. Was the evaluation carried out by a competent, trustworthy, objective staff?

> An eloquently written evaluation by a staff member with a history of inflating numbers, blaming mistakes on others, or falsifying information will not be taken seriously by your administration. An eloquent evaluation written by the project leader will carry less weight than one conducted by an external evaluator or an internal evaluator without a vested interest in the outcome of the project.

2. Were the relevant stakeholders involved in the evaluation?

> At the very least, an evaluation of a collaborative project is considered incomplete if one or more stakeholder units were not included in the design of the evaluation and review of the results. At worst, the evaluation may be suspect. The nature of the involvement of stakeholders is also an issue. Was involvement coerced or voluntary? Do the evaluators have supervisory responsibilities for some of the individuals participating in the study?

3. Did the evaluator take the context into consideration when reviewing the results?

> A document written from the perspective of the technology organization risks appearing naïve if it fails to account for factors operating within the higher education context. Technology programs are not implemented in isolation from factors such as the increase in nontraditional students, fiscal problems of higher education institutions generally, the distance education movement, and deferred maintenance problems.

4. Were reliable and valid measures used?

> In many respects, administrators are less concerned about the reliability and validity of the instrument used than in the other factors mentioned above. The reputation of the author of the document often colors administrators' judgement about its reliability and validity. Of special concern to any evaluator should be the accuracy with which s/he quotes faculty, students, and staff participating in the study.

IV. Case Study: University of Virginia School of Architecture

By the beginning of 1993 the School of Architecture at the University of Virginia was at a stage many organizations go through as they embrace information technology. The computing environment had been created and managed by a few interested faculty and administrators. Two key players had left within the last year, and the computing environment was in disarray. The administrative and academic areas were uncoordinated, reflecting the structure of the university computing organization. University and school administrators felt increasing internal and external pressure to significantly increase the use of information technology in instruction. Resources were extremely limited, with little money budgeted for information technology and no internal technology support staff.

The planning that had been done was too local and limited in scale to serve as a guide for the extensive, complex, networked environment that was evolving. The plans presented to the

university administration had shifted significantly as the key people in the school changed. The provost wanted future investments to be guided by a more long-range and consistent plan. One of the university's responses to this problem was to commit an internal consultant to the school for an initial period of six months. This person had two primary charges:

- Bring the current computing environment to a reasonable level of reliability and service.
- Help the school create a set of strategic and tactical plans.

In both of these areas the dominant factor was extremely limited resources, both fiscal and personnel. The vision that was developing in the school would require an order of magnitude more resources than were currently available. It was clear that:

- There was very little margin for error—the technology had to work.
- All acquisitions and activities had to contribute to the future as well as the present.
- The technology expenditures had to maximally benefit architecture, not just technology.

To deal with these criteria the university had to develop a scheme that provided some assessment of the impact of a particular technology intervention upon the instruction and practice of architecture, as well as one that helped determine which interventions were most cost-effective.

Information technology has become so entwined with the practice and teaching of a discipline that people find it difficult to separate content from technology. When asked, "What is the problem?" the instructor's answer is likely to be "We need more memory," rather than "Our students need to be able to create clear and concise project proposals."

To deal with this, UVA devised an evaluation scheme to direct thinking into three distinct areas: discipline content, computer literacy, and infrastructure. This scheme provides a direct link from discipline requirements to infrastructure design and expenditures. For example, the discipline need "A landscape architect must be able to compose a clear layout plan for a site" links to a computer literacy requirement that "Students must be proficient in at least one computer-aided design and drafting program." This in turn leads to an infrastructure specification that "LAND CADD will be available on all workstations connected to the School of Architecture network."

The foundation of the scheme is a collection of knowledge statements. These statements can be converted into questions that can be used to establish a baseline, to set goals, and to measure progress. Examples of the different forms are shown in Figure 1. It is also possible to adjust the resolution of the question depending upon specific needs and the amount of effort the organization is willing to spend on the evaluation. They can be phrased to gather simple yes/no answers, choices from multiple alternatives, or precise numbers.

Statement: An architect should be able to write a clear and concise project description.

Baseline: What percentage of third-year students can create a clear and concise project description?
Goal: What percentage of third-year students should be able to create a clear and concise project description?
Progress: What percentage of third-year students have subsequently demonstrated an ability to create a clear and concise project description?

Figure 1. Knowledge Statement and Derived Questions

It is important to emphasize that this collection of statements is not meant to exhaustively define an

area. It is impossible to get agreement on such a list. Such a focused statement is more like a null hypothesis: if a person does not understand the statement, most others would agree that that person is not knowledgeable in the area. For example, if a student can not "efficiently create accurate and detailed schematic drawings," that student is not competent to practice architecture. This relates to technology planning in the following way: if access to a CADD lab does not increase the number of students who satisfy this criterion, it may indicate that information technology is not a cost-effective solution to this particular problem. Or it may indicate that the facility is deficient.

Associated questions in the computer literacy and infrastructure areas can help determine if the problem is with the facility or instructional process, and can lead to corrective intervention.

The major effort in this method is the generation of the knowledge statements. People in the discipline must generate the content statements, although they often need coaching to keep them from drifting into the areas of computer literacy or even technology infrastructure. Technical staff should generate the infrastructure statements, and the collection of computer literacy statements is best generated by a combination of discipline and technical personnel.

For evaluation, the first step is the determination of the current knowledge of the target group (students in this example). A questionnaire, formulated as shown for the baseline statement in Figure 1, should be given to administrators, faculty, and students. The second step is to establish measurable goals. A rephrased questionnaire should be presented to administrators and faculty. The results of these two questionnaires should then be used to determine literacy requirements and as the basis for infrastructure design and development. There are four basic considerations for this stage:

- What are the most important goals?
- Where do we find the greatest discrepancies between the current state and the goal?
- In which areas will technology have the greatest impact?
- In which areas will technology have the lowest cost/benefit?

Additional questionnaires can be used to measure progress. These have limited utility, as will be explained in the conclusion.

This scheme is already producing useful results, though it has not been tested in its entirety. The process of generating the knowledge statements:

- Brings content issues to light so that they can be rationally examined.
- Enumerates the core content of the discipline as it relates to technology.
- Starts the process of establishing priorities.
- Indicates the areas where technology intervention is important.

The initial questionnaires have been used to generate a rational framework for the design of the technology environment and as a guide to optimize technology investment. Additional questionnaires can be used to measure progress toward goals. In reality, the rapid evolution of technology and the steepness of the assimilation curve are likely to cause the goals to change significantly during the time of the intervention. Thus, although it would be possible to conduct a summative evaluation, there would be little meaning in the results. As a formative tool, however,

the technique addresses many of the special problems of technology planning. It provides a direct link between discipline requirements and technology interventions, offers guidance in setting priorities, and provides a rational basis for resource allocation decisions.

As with all such methodologies, this one is not the ultimate answer. It is one more tool to be shaped and applied by those of us trying to manage information technology in higher education.

V. Case Study: Evaluating Classroom Technology

Indiana University was concerned that freshmen were not fully engaging in freshmen-level courses. We undertook a project to help reinforce a "culture of learning" at Indiana University by targeting three large lecture courses and assisting their faculty as they worked to engage freshmen more fully in them. The courses were introductory psychology, 100-level mathematics, and business law.

A number of the innovations in these three courses had technology as their cores. In introductory psychology, students responded via computer to questions about course concepts, allowing the instructors to modify the next lecture to clarify the concepts. In addition, students completed computer-based quizzes that tested their understanding of readings and course materials. Students could re-take the quizzes until a 70% success rate was achieved. In mathematics, students independently completed practice problems using interactive software and modules prepared by the professor. In business law, lecture discussions were extended through e-mail, thus increasing faculty and student contact.

From the beginning, we were aware that there would be statistical limitations to the evaluation. Course assessment measures were by necessity non-intrusive. Each innovation was evaluated from three perspectives: the faculty member's perceptions; observations by external personnel; and student responses to tests, course evaluations, and focus groups. In the introductory psychology class, students scored significantly higher on a common final exam than their counterparts in six other sections. Typical comments included, "The course stimulated me to think in a better, different way." In mathematics, the students in the targeted section had higher median scores on both the midterm and final examinations, although the differences were not statistically significant. In business law, daily attendance reached the 93% mark.

The results of this evaluation were reported to four audiences. A technical report, written by the evaluation team, was distributed to the evaluation team and the participating faculty. A narrative description was distributed to the university board of trustees during its regularly scheduled meeting. An oral presentation of the project and evaluation were presented to the senior administrators of the university. There was a definite logic to this distribution plan. In each case, an effort was made to report the data of interest to the audience in a format that was comfortable for that audience, without compromising the integrity of the report.