

[Presentation Notes](#)
[Paper](#)
[Bio](#)
[Return to Main Menu](#)

PRESENTATION

F10

Friday, November 5, 1999
11:15 AM

BIRTH OF A TEST ORGANIZATION

Lisa Bresko

infoworks, A VIACOM Technology

INTERNATIONAL CONFERENCE ON
SOFTWARE TESTING, ANALYSIS & REVIEW
NOVEMBER 1-5, 1999
SAN JOSE, CA



Birth of a Test Organization

Lisa Bresko
Manager Business Application Certification

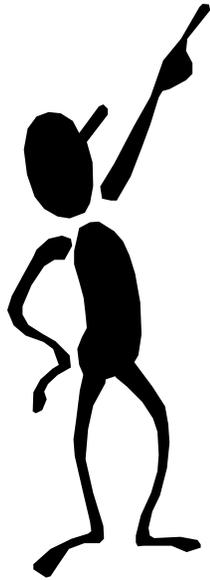
Lisa.Bresko@Viacom.com

Crawling - Who are we?



infoworks
a VIACOM technology service

Crawling - Make a statement



“Quality Management is dedicated to delivering stable, useable and reliable business solutions on time with maximal positive impact using a well-defined process”

Crawling - Set Expectations

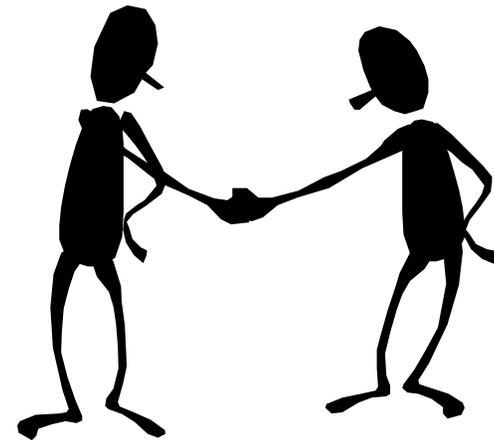
- Group Mix



- Primary QM Business Liaison
- Roles & Responsibilities

First Steps - Speaking

- Sell, Sell, Sell
- Communication Routes
- Establish Identity
- Application Assessment



First Steps - Writing

- **Methodology** - Define what you will deliver
- Create uniform documentation by using a **Test Plan Template**
- Standardize **Test Scenarios**
- Document **Test Assets**

First Steps - Playing

A's and B's

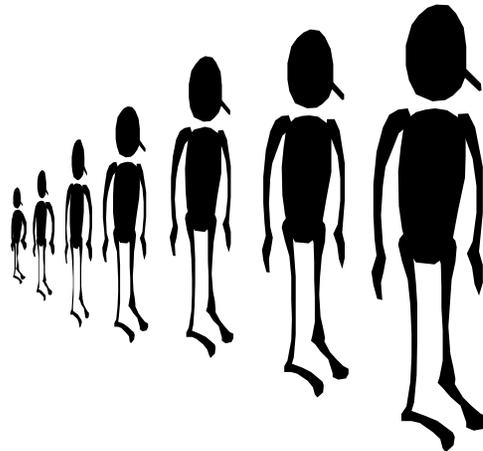
“You can't do it all” - Pick the players

“Do it all” - Prioritization

Walking - Plan

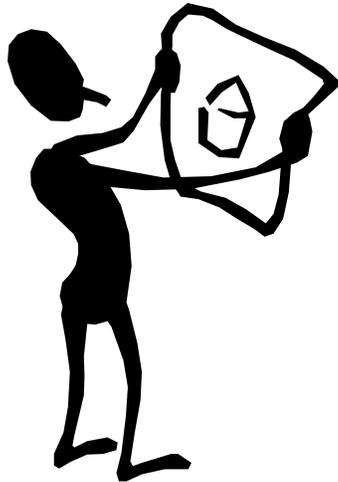
Who?

- Resources to test
- SME (subject matter expert)



Walking - Plan

What?



- Application Assessment
- Master Test Plan
- Test Scenarios
- Automated vs. Manual Scripts

Walking - Plan

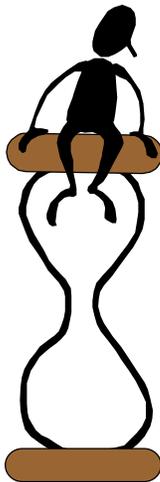
Where?

- QA server environment
- QA workstations



Walking - Plan

When?

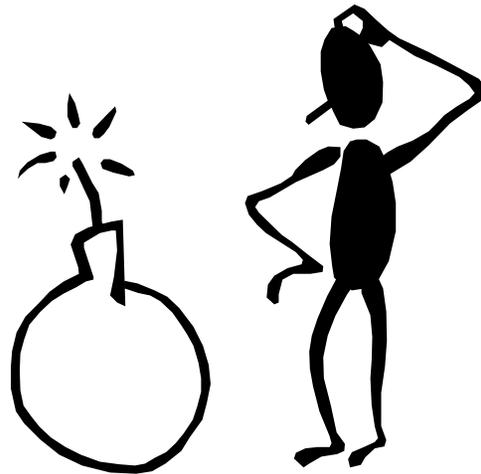


- Start early
- Initial walkthroughs
- Official QA period

Walking - Plan

Why?

- Identify risks
- User acceptance testing



We're Testing

- **Hugs for Bugs**
- Take pictures
- Send smoke signals

Walk don't Run

Wrap-up those test assets

- Review test plan
- Index to assets



Walk don't Run

Celebrate Successes



Review Improvements



Walk don't Run

What's Next?

- Evaluate your organization
- Look for improvement
- Get Started



Questions?



Thank You

Birth of a Test Organization

Lisa Bresko
STAR '99 West

Birth of a Test Organization

Lisa Bresko

STAR'99 West

Introduction

This paper aims to identify practical, easily implemented solutions to establishing a test organization. Often testers are put in a test group that is defined in name only. Regardless of what name your organization uses for the test function, you need to determine your group's identity and what you will and will not be responsible for. I liken this process to that of the growth stages of a child from crawling to first steps to walking.

The solutions defined are those that were tried and often worked for my team. My experience in various areas of IS&T organizations gave me a unique view when presented with the role of test manager and the task of defining a test group. As a team we have progressed through the difficult birth process and are now in the walking phase. Our future goal will be to run, although we aren't there yet.

The solutions implemented by my team are by no means the requirements for defining a test group. Hopefully they can serve as a guide for things to take into consideration when defining the test group to serve your specific organization. These solutions did enable us with almost no budget or increase in headcount to become a respected Quality Management group within our IS&T organization.

During this paper I may interchangeably use "test group", "team", "Quality Management group", "QM" each reference is to the one Quality Management Test Organization at infoworks.

Crawling

Who are we?

Before we could define ourselves as groups we needed to determine whom our customer's were and where we fit in the big picture. As a service division of Viacom, infoworks was developed almost two years ago by merging the IS&T organizations from MTV Networks, Showtime Networks and Viacom. Before infoworks was created Quality Management existed only at Showtime Networks. Due to the creation of infoworks, Quality Management now had a much broader customer base that was very varied. Not only were the technologies that we would need to support expanded, but the business groups and culture of each area was very different. We needed to determine how we would address this varied customer base and represent ourselves within infoworks.

Make a Statement

One of the first steps in determining your test group's identity is to clearly state who you are and what your overall goal as a group is. This is essentially a mission statement and although I question the importance and use of a mission statement in general, I think it is important for your group and for the overall organization to have a simple statement that reflects who you are. We developed our "statement" (the team refused to call it a mission statement) one afternoon in the park together.

"Quality Management is dedicated to delivering stable, useable and reliable business solutions on time with maximal positive impact using a well-defined process"

The statement itself should simply reflect what you want to achieve as a group. Our key words were "dedicated", "stable", "useable", "reliable" and "well-defined". Each member of the team participated in the development of our statement and agrees that it will reflect us as a group. The statement itself is only used in documentation for our methodology and in some presentations to management.

Group Mix

Because we were supporting such a varied customer base and infoworks itself was a new entity, we needed to determine what our group make up should be to best position ourselves with the customers and the existing infoworks organization. Unfortunately during the merger a number of Quality Management team members decided to leave and the existing team needed to be supplemented, but with whom?

Because our environment is so varied, the team itself became very diverse. We looked for technical skills that were lacking in the existing team which were now required and we looked for personalities that would work well within the varied cultures that represented our business customers. The goal was not to have a "cookie cutter" type tester. Different skills within different individuals gave the overall team depth. Since the existing team was very strong technically and came from development backgrounds we determined we needed more of a business analyst with excellent communications skills. The key was determining what was lacking and then rounding out the team with those skills.

In addition, we had no budget to increase headcount, but not enough resources to handle all the different projects identified, so we worked within project budgets to bring in additional consulting resources as needed. The consultants worked with the fulltime staff as a supplement to their testing. Each member of the team, staff or consultant worked as part of a team effort and followed our established methodology. We are beginning this year to budget additional consulting resources in advance instead of having to find the money within a project when it comes time for testing.

Primary QM Business Liaison

In an effort to support each business area's projects each QM group member was assigned a business area to support as the Primary QM Business Liaison (we were short one person so we combined two small groups). We identified the primary QM business liaison to the business area, the applications development group supporting the area and the other infoworks infrastructure groups. The primary QM business liaison is identified as a 75% resource for testing with the remainder of their time spent on information/communication, software configuration management and development of the overall QM process. For projects requiring more than the 75% testing resource, additional consulting dollars are budgeted within the projects when they are defined by the business and applications development. This ensures that the testing gets done and that the dollars are there when needed. I work with the business and applications

development during the project definition phase to help estimate the testing resources that will be needed.

Roles & Responsibilities

Because the Quality Management group plays a variety of roles within infoworks in support of the various business areas, I felt it was necessary to clearly communicate what role we will play and what we expect from the groups we interact with. We developed a matrix of roles & responsibilities for the Primary QM Business Liaison and presented it to infoworks and the business area management teams. Below is a sample of the matrix for each of the three areas of responsibilities:

Information/Communication Roles and Responsibilities:

QM Manager	QM Primary Business Liaison	Applications Development	Infoworks Infrastructure Support Groups
Identify a QM Primary Business Liaison for each applications development area	Attend appropriate meetings for supported applications development area	Invite QM primary business liaison and/or QM Manager to appropriate meetings	
Review prepared Application Assessment form for completeness	For each project involvement complete the Application Assessment form	Review prepared Application Assessment form for correctness	
	Forward completed Application Assessment form to Infoworks Infrastructure Support groups		Review Application Assessment form and follow-up with applications development regarding information affecting individual area

Test Management/Certification Roles and Responsibilities:

Application Development Project Manager	QM Manager	QM Primary Business Liaison	QM Consultant
Project Support			
Make requests for new project involvement	Receives requests for new project involvement		
Reviews and supervises all testing performed	Reviews and supervises all test methodology and standards employed	Performs testing and reviews test effort/status performed by QM or external consultants	Performs testing and reviews test effort/status with QM Primary Business Liaison
Provide copies of any available documentation, prototypes to QM primary business liaison	Review all master test plans, scenarios and test results presented to Application Development for proper methodology and standards	Provide documented master test plan, scenarios and test results to Application Development project manager for review	Provide documented master test plan, scenarios and test results to Application Development project manager for review

SCM Roles and Responsibilities:

Developer	Data Base Administrators	QM Primary Business Liaison
Tool Administration		
Creates appropriate PVCS projects as need	Creates PVCS projects to store database objects	Provide training and support as requested by development and DBA staff
Contact the DBA Staff to move updated DB objects (from archives) to the appropriate environment	Assign standard version labels to changed database objects, check-out changed database objects from PVCS archives and compile them in appropriate database environment	Provide technical assistance to DBA staff and assure that all file revisions have standard version labels assigned to them

This sample gives you an example of the different roles that were defined and how we identified what the Quality Management objectives were and what we required from the areas we interact with. The business areas are not part of the matrix since in our

organization they work as a team with applications development during a project's life cycle.

First Steps - Speaking

Sell, Sell, Sell

You must be your own advertising agent within your IS&T organization. In many cases test groups are the low men on the totem pole and it is up to you to get your team recognized for what they are doing and how they are progressing as an organization. Take every opportunity to get involved in overall IS&T process oriented meetings, applications development staff and/or project meetings and business events that can give your team exposure. Establish the test team as the ever friendly, always helpful and energetic people to call for assistance or information.

Communication Routes

Communication needs to be started as early as possible and be consistent throughout a project's life cycle. Have your team sit in on as many project meetings, staff meetings and business meetings as possible. If there aren't formal meetings or you're getting resistance from the applications development area, then have them make it a habit of scheduling ½ hour with some of the development team for review of what they are doing and to try and get as much information as possible. I've found scheduling ½ hour meetings with each of the business area applications development and infrastructure vice presidents once a month to be a great time to just let them know what we are doing as a group. Even if the conversation doesn't revolve around business topics, you are being recognized as part of the overall IS&T organization. I use this meeting to briefly status hot projects we're involved in and to identify any potential concerns or issues that have filtered up to their level from their respective teams. In general most people don't mind a ½ hour of their time being taken if you show that your main concern is to make sure they are informed and that you want to catch any potential problems early.

Establish Identity

Make sure your applications development teams know who supports them and attempt to establish a team approach to the project delivery. I know how difficult this can be, but it is essential to your success as a test group. Perseverance is the key and holding up your end of the team to stay on schedule even if that means cutting corners in testing, can earn you the partnership you need. Go for lunch, drinks, whatever it takes! The goal is to become a team with applications development and then get them to do things your way!

Application Assessment

Establishing yourself as a resource of information for the infrastructure departments in your organization can also enhance your image. We developed a simple form that we use as an application assessment and can be completed from design documentation, system documentation or if there is no documentation, which is often the case, by simply asking a series of questions. The information is helpful when planning your test effort and we also pass it along to all the other infrastructure groups (Network & Data Center Ops, Database Administration, Tech Standards, Logistics and Training) as a heads up. Often these groups are not given any advance notice during a project's development and this information helps them contact the necessary people to prepare for that phase of the project development. Below is a sample from the application assessment form we use:

Environment:

Assessment Question	Assessment Answer
General environment:	
What operating system(s) are the users running?	
Equipment:	
Are you planning on using existing equipment (servers)?	
If No – Has equipment been ordered?	
Application environment:	
What is the user environment (Laptop/Desktop)?	
User specifics:	
Number of expected users?	
Location(s) of expected users?	
Special considerations:	
Please specify any special environment requirements for this application.	

Application:

Assessment Question	Assessment Answer
General Application:	
Is this new development or changes to existing application?	
Application Deployment:	
If existing application – has it been deployed to Production in the past?	
Application Specifics:	
What is criticality of application?	
Special considerations:	
Please specify any special application requirements:	

Documentation:

Assessment Question	Assessment Answer
Business requirements document?	
Application design documents?	

First Steps – Writing

Methodology

It is a good idea to develop a basic test methodology, which can be expanded and revised as your test group matures. Your methodology should indicate how you receive and prioritize the requests for testing, how you go about testing, what results you keep, automation versus manual test methods and how you will report during the test process. This can be as general as a business flow diagram or as complex as a fifty page detailed methodology. My advice would be to keep it simple, but consistent. Our methodology consists of the following sections:

- Change Request – how are requests for QM services managed
- Prioritization – what determines the test effort
- Test planning – test plan template
- Test design – test scenarios
- Automation versus Manual testing
- Defect recording and tracking
- Completed test results – how is it documented
- Index to test Assets – where is everything

We employ the same methodology for all projects and produce the same documentation regardless of whether we are doing a full system test or simply testing a new module to an existing system. We are still in the progress of expanding this methodology and it will be an ever-evolving part of our group as we grow and mature as a test organization.

Test Plan Template

During our Y2K certification effort we developed a master test plan template which has been modified and evolved to the Test Plan template that we use today. This provides for a uniform way of documenting the test effort and enables us to get the approval of the appropriate business and applications development staff on what will be tested. The template currently captures the following sections and is used for all testing done by the QM group:

1. Change Summary
 - 1.1. File Location
 - 1.2. Version Notice
2. Introduction
 - 2.1. Purpose/Scope
 - 2.2. Responsibilities
 - 2.3. Tracking
 - 2.4. SQA Repository and Project
 - 2.5. Resolution
 - 2.6. Severity Criteria
 - 2.6.1. Defects
 - 2.6.2. Software Promotions
 - 2.7. Status Reporting
 - 2.8. Hardware Problems
3. Test Overview
 - 3.1 Project Plan
 - 3.2 Software Overview
 - 3.3 Scope
 - 3.4 Hardware/Software Overview
 - 3.4.1 Prerequisites
 - 3.4.2 Test Tools
- 4 Test Requirements
- 5 Test Deliverables
 - 5.1 QM Test Scenarios
 - 5.2 Scenario Template
 - 5.3 Artistic Testing
 - 5.4 Defect Recording
 - 5.5 QM Certification Statement

The template covers many of the standard categories that are described when reading about what should be covered in a master test plan. The whole team agreed upon these sections and the format was developed as a group effort. As we evolve as a test organization this template will grow with us. The most important sections for any template are the Software Overview, Scope and Test Requirements. The team will often develop these sections upon becoming involved in a project and fill in the remaining sections toward the end of the test effort.

Scenario Designer

Just as the test plan template gives us a uniform way of documenting our test effort, we wanted our test scenarios (or test cases) to be uniform as well. We developed an internal VB application that resides on the SQA repository that we use for all automated testing to define our test scenarios. Each test requirement, identified in the master test plan, requires a test scenario to be entered in Scenario Designer regardless of whether the test will be automated or manual. Scenario Designer documents a purpose, the detailed method, expected results and notes for each test designed. It also provides us with a history, which is helpful, when one member of the team joins a test effort and modifies an existing scenario. The goal of Scenario Designer is to uniformly record the steps necessary in duplicating any test. It becomes the heading of all automated tests in

the SQA .rec file, but would be just as useful as a template residing in MS Word if the majority of your test efforts are manual at this time.

Test Assets

With the goal of our team to automate whenever possible most of our test assets are located in the SQA repository. For those tests that are not automated we generally use screen shots, screen cams or printed reports to document the test results. In either case it is important to determine a standard directory and directory structure to hold all of your test assets. Within our SQA repository project directory we create a docs directory to hold the application assessment form, master test plan and any test results that are in the form of screen prints, screen cams or reports. We create a high level document called an Index to Test Assets that points to the location on the server of all test assets and identifies what was done for a particular project. The goal here is to enable your team and anyone else who is interested, to quickly and easily find all documents pertaining to a particular project.

First Steps – Playing

A's and B's

Generally a test group is told “Do it All” or “You can’t do it all” in either case you need to determine which applications will be the players for your particular group to test. If your management is of the “Do it All” mentality then have them prioritize and make sure their expectations regarding time necessary are realistic. If the reality is that time will not permit you to “do it all” then within the application identify the (A) areas – those that must be tested and the (B) areas – those that will be tested if time allows. Perhaps you need to validate the data through the GUI (A), but if you have time it would be good to also validate your data results against the database (B). Just be sure to have signoff from applications development and/or the business that this level of testing is acceptable. The “You can’t do it all” mentality works the same way, but on the application level. Have the business and/or applications development identify which are the (A) applications – must follow the standard test process done by the QM group or (B) – those that can go directly to a user test after applications development does their own version of QM testing. By attempting to get the business/applications development to determine what is critical and therefore requires the more rigorous testing effort, you enable your team to be effective and follow a process that makes their efforts more valuable.

Walking – Plan

Who?

Determining the resources necessary for testing is always difficult, but if you determine what will be tested that estimate becomes easier. You need to do take the First Steps before you begin to Walk. Once you have determined the testing resources make sure you identify any other resources necessary to the test effort. These can include applications development as well as other infrastructure groups. In our organization the applications development group coordinates with the infrastructure groups, but we found we didn’t always know whom to speak to in applications development, hence the SME. The SME a.k.a. Subject Matter Expert is the key contact person for the project and can either be someone from the business or the applications development group. They coordinate with our team and serve as the knowledge base for the application being tested. By implementing this simple process of identification we save lots of time when questions or issues arise.

What?

Identify what will be tested and then begin assembling your essential pieces to the test process. For us this includes beginning the application assessment form, creating a master test plan, identifying test scenarios and determining if the test effort will be manual or automated.

This is where you're actually getting ready to test!

Where?

Unfortunately this usually doesn't mean that you can take off for the beach to begin that test effort, but you do need to make sure your test environment needs are being addressed. Ideally you should have a separate database environment (essential if you are automating your tests) and a pristine desktop environment that mimics the business user's desktop. Because budget issues prevent our group from having it's own lab we require from applications development either a QA or UAT database environment dedicated to us during our test effort. We've managed to maintain different business group user's desktops by implementing imaging software, which saves us the need for multiple test workstations. The imaging software is relatively inexpensive and we use JAZ drives to boot a workstation with the desired image when ready to test.

When?

Begin your test effort as early in the project life cycle as possible. Sit in on design meetings, review documentation, talk to users and request demos or access to prototypes. The earlier you begin the more efficient your actual testing will be. We've implemented an early pre-release application walkthrough, which gives us access to an early release of an application, perhaps before all the business logic is complete and we use it for creating our master test plan and test scenarios. We don't consider it an actual test and we don't report defects, but it saves time when we do begin the test effort because we are aquatinted with the application and have already completed a draft of the test plan and scenarios we will use to test. Make sure you are clear when the official turnover of an application is and when actual testing will be scheduled to occur.

Why?

This is the reason for your existence so you need to be clear on the risks in not testing. Work with the business to identify how they will be affected if an application doesn't perform as expected. Don't be afraid to make recommendations the worst that can happen is they ignore the risk, just make sure YOU don't ignore the risk. Encourage a User Acceptance Test and if resources permit help define the UAT tests that will be conducted. Be an advocate of the business users as well as applications development on delivering a quality product.

Walking – Test**Hugs for Bugs**

How do you get a developer to fix bugs? Give Hugs – Hershey's Hugs that is. The Hugs for Bugs campaign was developed when we began seeing a large number of repeat defects. Defects that were identified supposedly fixed and when re-tested appeared the same as before. We started giving Hershey's Hugs for each bug that was successfully re-tested in the next version, if the bug appeared in a later release the developer was deducted one hug. Once we finished testing a release the developer would be delivered a handful of hugs for all completed defects. We've modified the program to suit different tastes and now include fruit, nuts and assorted candies. The developer's think it's great and we don't have the frustration of re-testing the same defect over and over again.

Take Pictures-Send Smoke Signals

Document! Document! Document! Be sure of what your testing and if possible get confirmation on your results. Take plenty of screen shots and/or screen cams while testing even if the majority of your testing is automated. If a user wants to see the results, a screen print is much clearer for them than the test log of your automated test tool. Make sure you send up smoke signals if necessary while testing. Reporting defects is not enough if the impact of those defects is not readily apparent to the user.

The golden rule here is involve your applications development team in all communications and NEVER point the finger!

Walk don't Run

You've finished testing and now it's time to compile your documentation, evaluate your test plan and determine what the results are. Make sure everyone on the team can access your documentation and knows where to look for the test assets so that while you're on the beach in Hawaii someone can step into your shoes if necessary.

Celebrate your successes! This is the time to make sure management knows . . .
How the test effort went . . .
Did you stick to schedule. . .
How many defects avoided production. . .
See the results . . .

You need to spread the word and make it known that your team worked efficiently and produced a desired result. Don't wait for anyone else to come around to singing your praises!

Review improvements! This is the time to make sure management doesn't know . . .
What went wrong . . .
How behind schedule were you. . .
How many defects did you miss. . .

This is the time for improvement, not pointing fingers or attempting to push the blame off onto other organizations. Evaluate what worked and what didn't and make the appropriate adjustments. Get feedback from the groups your team interacted with and make positive plans for avoiding problems in the future. This is a time of growth and each member of your team needs to provide input into the process and make recommendations for overall improvement.

Conclusion

The ultimate goal is to develop a team that can work together toward a common goal and feel an element of success in the overall process. Developing a test organization takes time and if you use even a handful of the suggestions we've implemented you'll be on your way to defining your unique test group. Remember that it is a growth process and you need to start at the beginning to Walk like a real test professional. Our group is still in the walking phase, but I have hopes that with time and a team oriented approach we will be running soon.

If you would like to receive any of the full templates being used by my group or have any questions please feel free to contact me via email: Lisa.Bresko@Viacom.com

LISA BRESKO

Lisa is currently manager of business application certification for Infoworks, a Viacom technology service. She leads a small test team that has made a large impact on the way testing is viewed by the business groups they support. The team is responsible for testing of critical business applications on a variety of platforms supporting the dynamic entertainment businesses of MTV Networks, Showtime Networks, and Viacom.

With 15 years in the information services industry, Lisa has been in a variety of roles including applications development, database administration, project management, and finally, quality management. Working in a variety of roles and responsibilities within various IS&T organizations has provided her with the basis for the implementation of a series of basic strategies for creating a successful, respected quality management organization. She started her career in quality management as a tester and has progressed into a management role with a very team-oriented approach. Her goal is to foster the team relationship between her group, applications development, and the various infrastructure groups in her organization to produce quality software they can all be proud of.