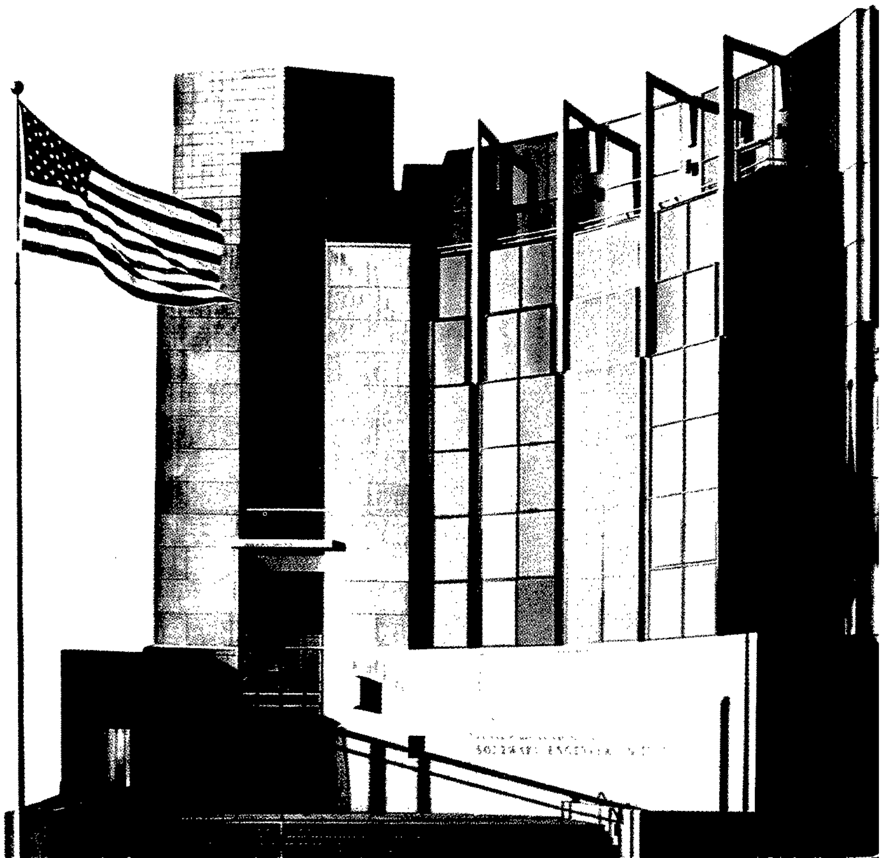
Carnegie Mellon University
Software Engineering Institute

A Turbo-Team Approach to Establishing a Software Test Process at Union Switch & Signal

Don McAndrews
Janice Marchok Ryan
Priscilla Fowler
April 1997

SR

Special Report
CMU/SEI-97-SR-002



Special Report
CMU/SEI-97-SR-002
April 1997

A Turbo-Team Approach to Establishing a Software Test Process at Union Switch & Signal

Don McAndrews

Janice Marchok Ryan

Priscilla Fowler

Transition Enabling Program

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Table of Contents

List of Figures

A Turbo-Team Approach to Establishing a Software Test Process at Union Switch & Signal

Abstract: Process improvement teams are often created and tasked to make complex changes in an organization, but are not provided with the tools necessary for success. This report describes what one team did to successfully install a software testing process in an organization. The principles of a turbo-team approach are introduced and a defined process for working in teams to introduce a new software technology is adapted from the prototype *Process Change Guide* developed by the Software Engineering Institute (SEI). Artifacts of this effort are included and described as examples for other teams to reference. Successes and lessons learned are described.

1. Document Overview

This report chronicles the experiences of an improvement team in introducing a new approach to software testing at Union Switch and Signal Inc. (US&S). US&S collaborated with the Software Engineering Institute (SEI) Transition Models team to follow the phased approach to introducing new technology that is described in the prototype *Process Change Guide.*[1]

1.1 Intended Audience

This document is written for improvement team leaders and team members (e.g., members of action teams, technical working groups, or software engineering process groups) who are introducing a new technology to their organization. A technology, in this case, is defined very broadly—it could be a new tool, such as a configuration management software program, or a newly defined process aimed at helping an organization improve some aspect of its business.

1.2 Expected Use

This report and the detailed description of experiences should provide ideas and examples for future improvement teams and others introducing technology-based changes in an organization. The US&S artifacts included in the Appendices should be considered examples rather than best practice, since they reflect only one organization's experience.

[1] The *Process Change Guide* prototype version was co-developed by the SEI and Xerox Corporation during 1995 and 1996.

The US&S lessons learned should provide some guidance about what teams should watch out for as well. Readers should plan to adapt any solution described here to their organization's specific requirements.

2. US&S Overview

This section provides a brief overview of US&S and its history with process improvement. It sets the stage for describing the software testing improvement effort.

US&S was founded in 1881 by George Westinghouse to develop and manufacture railway safety and signaling devices. For more than 115 years, US&S has developed technologies and products that have made railway operations safer, more reliable and more efficient. Today, US&S products and systems are in service in a multitude of countries on all major continents, including virtually every railroad and transit property in North America. The company employs over 1,000 people, and annual orders are measured in hundreds of millions of dollars.

Some recent major undertakings include

- CSX Consolidation, Jacksonville, Florida: the first Class I railroad to control its operation from a single office location, dispatching and managing the movement of 1,400 trains over 19,000+ miles of CSX railroad track.
- Union Pacific Consolidation, Omaha, Nebraska: designed and built the Harriman Dispatch Center in 1991, enabling Union Pacific to control its entire 20,000-mile network from one central office.
- Massachusetts Bay Transit Authority (MBTA) Operations Control Center, Boston, Massachusetts: the new operations control center consolidates everything from train and bus dispatching to the control of public address and ventilation systems; the first fully integrated metropolitan transit management center in the U.S.
- Los Angeles County Metropolitan Transit Authority (LACMTA) Green Line, Los Angeles, California: the most sophisticated transit line in the world, featuring a fully automated control system and phased-in driverless operation.

2.1 Focus on Software Testing

As the systems at US&S have grown in complexity and size, the organization has recognized the need to develop more formal, structured processes for developing software.

The demand for improvements in the area of software testing came from more than one direction. First, individuals in both management and engineering were recognizing the need for emphasizing improvements in testing as part of the challenges they were facing with growth in applications and advances in technology. Additionally, some of the systems that were being fielded were experiencing too many problems, to the point where customers were becoming dissatisfied with US&S products. Finally, customers were beginning to require that a software test team be in place to provide an independent validation of the systems before shipment.

Several years before the effort described in this report, a software testing working group was formed so that the project engineers dedicated to testing could share methods and lessons learned across projects. The testing working group was successful in some areas, but failed in others. One of the successes was the documentation of a standard template for system/acceptance test plans. Another success was the development of some expertise in the area of testing through regular meetings to discuss common problem areas and attendance at external training courses. However, the group never really established a process that could be installed on all projects, nor was the development of testing established as part of the overall software process. System validation-level testing was identified in US&S's Software Engineering Handbook, but was not defined. The process was highly dependent on the individuals implementing it. If those individuals left the company, the testing expertise left with them.

Other problems resulted from the fact that the testing working group never had a clear focus and direction. Initial charters for this group included objectives or activities focused in the following areas:

- defining and documenting the process for testing
- developing a proposal for a formal independent test team
- evaluating and selecting tools to automate the process
- developing training

These areas were to cover all levels of testing, from unit testing through integration and system testing.

This charter was too broad. In addition, the team was assembled from projects across the organization, with varying needs and expectations from the testing working group. The testing working group of 4-8 individuals met periodically (weekly or bi-weekly) for over a year, with varying degrees of support from the organization. This working group dissolved in early 1995.

Some key lessons were learned from this group relating to establishing and staffing teams, focusing their efforts, and executing plans for success. One major lesson learned was the importance of a well-defined charter. Although team members met regularly, their effort was fragmented and unproductive because too many issues were being worked with no clear goal defined. In addition, it was also observed that for the team to be involved in improvement activities, the members of the team must have a stake in the results—i.e., they must be able to apply the results of the team to their own jobs. And finally, the team recognized that their efforts would only be useful and recognized with management support. This was exemplified by the fact that the managers that supported the team the most had the most critical needs for testing on their projects.

2.2 Collaborating with the SEI: The Prototype Process Change Guide

In late 1995, an increased emphasis was being placed on system testing on projects. Some major projects were nearing the peak testing periods, and several new projects were starting up. It was clear that it was time to focus once again on establishing a formal test team. Some of the individuals who had been part of earlier testing working group, and were still actively involved in testing on projects, were chosen to charter a new team focused on testing.

This time the team used a prototype model from the SEI for transitioning technologies—the model described in the prototype *Process Change Guide*. This guide had evolved from work conducted by the Transition Models project at the SEI and was co-developed with Xerox Corporation. US&S and the SEI established a cooperative research and development agreement (CRADA) to collaborate and use the prototype *Process Change Guide* materials to help define a testing process and install it in the organization.

The collaboration involved one SEI person working half-time at US&S on the software test team effort. Another SEI person provided coaching and technical support along the way based on her experiences co-developing the prototype *Process Change Guide* with Xerox. A consulting engineer at US&S was selected to lead the software test team and act as the interface between US&S and the SEI. The manager of the US&S Software Engineering Process Group (SEPG) sponsored the activity.

The primary activities conducted as part of this collaboration were

- testing installed, the process documented, and a functional group established to perform software testing at US&S
- feedback on the use of the prototype *Process Change Guide* provided to the SEI

This report documents the experiences of the collaboration, including the successes and lessons learned and all artifacts created by the team.[2]

[2] Some artifacts are incomplete because of space limitations.

3. Why a Structured Team Process?

At US&S, it is now recognized that successful teams rely on the use of a structured, systematic team process. The software test team used a structured team process—referred to as the turbo-team[3] approach—to systematically work through the steps described by the prototype *Process Change Guide*. This section discusses the rationale and need for a structured team process, introduces the turbo-team approach developed at US&S, and describes how the prototype *Process Change Guide* was used.

3.1 The Reality of Teaming in Technical Organizations

Creating teams to address process improvement issues introduces a change to project dynamics. Engineers who have 40+ hour a week responsibilities developing software are asked to support a team focused on doing something else. Regardless of whether the engineers are highly motivated or resistant to working on the team, it represents extra work to them. Often it is seen as extra work for something that is not even going to benefit them on their job. For this reason, when people are asked to support an improvement team, it must be very clear to them why they were selected, and team activities must become part of their job.

While the members of the improvement team may work on a project that is well planned, controlled, and effective in developing software, improvement teams often come together without strategies for decision-making and conflict resolution, or even basic planning and organizing as a team. The improvement team setting presents an environment and situations that differ from their regular project team.

Further, "all-stars" of the organization are often asked to help solve a problem in a team situation. Although they may be the expert in a particular engineering or management domain, they may not be very proficient at working on a team, or solving complex organizational problems.

For these reasons, improvement teams should be provided with a structured process for working as a team. With a structured process, the team can come together, charter itself, identify operating procedures, define the problem, and solve it in a systematic manner. A detailed plan clearly identifies to the team member who is an engineer or manager what is expected of them, and the plan provides inputs to modify their already full project schedules. And finally, using a systematic, well-defined process for working in teams provides data to the organization on the successes and lessons learned. This data is then useful to the organization when planning future teams.

[3] See McAndrews, McDonough, and Matvya, "Turbo-Teaming Toward Improvement" [McAndrews 96].

3.2 Turbo-Team Approach

This section describes the principles of turbo-teams. US&S has applied these principles to several types of teams solving a diverse set of issues, from process improvement to project crisis to customer satisfaction. A brief definition of the turbo-team approach as applied at US&S is as follows:

> *Turbo-team approach: a quick, quantified approach to working in teams that emphasizes empowering the right mix of people to focus rapidly on a well-bounded problem/opportunity, while leveraging off of past team successes and lessons learned.*

The turbo-team approach is useful in any situation where a structured team or problem-solving process is required. It supports the problem-solving process by providing clear direction, quantifying effectiveness, and leveraging off teams that have already experienced success.

This approach evolved over the past few years at US&S as experience was gained in working with teams. Several teams contributed to this experience base, including the original test working group, a metrics working group, and a configuration management working group. The current turbo-team approach will continue to evolve as experience is gained and lessons learned are analyzed and documented. US&S believes it is important to recognize that although teams become successful only with experience, teams in an organization without experience do not have to start from scratch. One objective of this report is to provide access to the lessons learned from US&S; the approach documented here may be applied as a starting point to defining team processes in any organization.

The principles being applied in the turbo-team approach are described below:

- selective participation—getting the right people involved
- re-use—re-using past team approaches, systematic methods, artifacts, and lessons learned
- clear focus—having a clear scope, including a charter and plan
- quantification of costs and benefits—using measurement to set goals, plan, and track
- empowerment—giving the team power to make changes, not just suggest them

Selecting the Right Team Members

One scenario frequently experienced is having the wrong people on a team. Many teams experience the Pareto Principle—20% of the people do 80% of the work. Successful teams are those that are staffed with people actively working in the area of interest on their projects. People who are motivated make better team players because they see value in the team and are more willing to participate and invest effort. The team members will see the value in the team if they can apply the team products on their job. The team will recognize value in the members if each member contributes and brings something to the team. For this reason, it is best to avoid putting people on the team who are simply too busy to participate. No matter how interested or qualified people are, if they don't have sufficient time available to participate, they won't help the team. Productivity is important to improvement teams because these activities are limited to a few hours a week, often in addition to regular responsibilities.

Decision making is another team function that often causes friction, especially if, when the time comes to make a decision in a meeting, the wrong people are there. One way to solve this problem is to make sure that the right people are members of the team, and the right people are at the meetings where decisions are made. People selected as members and/or participants in meetings should be selected for the following reasons:

- Their inputs are required.
- They are affected by the decision and should participate in the decision-making process.
- They (as a team or as individuals on the team) have the authority to make a decision.
- They are empowered to provide the team with direction.

It is also important to consider the team make-up as the work of the team progresses. Often a particular team is assembled to address a problem, and after chartering and planning the team members realize that because of other commitments, or lack of interest, the team is not something they can support. Or, the team recognizes that they will need additional support from other areas of the organization. It is important to continuously re-evaluate the team membership, and keep positive contributors on the team.

Re-Use

The next principle of turbo-teams is also focused on productivity: re-use helps the team leverage off of what past teams have accomplished.

Process improvement work is analogous to software development. When an organization builds a system for a customer, they often start from scratch. Development of unprecedented systems usually means doing a prototype. Once experience is gained, the organization has some valuable tools, artifacts, and knowledge to re-use. They have the actual product, which may be re-used in its entirety or in parts. They also have developed some skills and techniques that they can apply to new customer requirements to avoid making the same mistakes. With each successive project, the organization gets a better idea of its capability, strengths, and weaknesses. These same ideas apply to process improvement teams. Software process problems are not often unique. Most organizations struggle with basic planning and tracking, requirements and configuration management issues.

Testing is a common area for organizations to try to improve. Consider the plethora of well-attended conferences that are given every year on these basic topics: testing, metrics, project planning, and tracking.[4] The improvement team members should recognize that the problem they are trying to solve has probably been addressed before, perhaps in their organization, and surely elsewhere in industry. The team should look for as much information

[4] For example, the SEI Software Engineering Symposium and SQE's Applications of Software Measurement.

as possible on potential solutions, or even full solutions or approaches, before attempting to create their own.

Even if the problem is unique, improvement team solutions or approaches to solutions are not. At US&S, several teams were initiated in areas of process improvement, project management, and customer satisfaction. These areas have had a diverse set of problems and situations that needed to be addressed. But each of these teams started out basically the same way: the first two to three meetings were spent forming the team, developing a charter and plan, and making sure the team had a direction in which to proceed. The charter and plan formats are similar for all of the teams. The process used to develop these charters and plans is also the same. Again, time was saved by quickly offering the newly formed team a method for helping define their problem and focusing it on a clear plan and direction. (See the charter in Appendix A.)

At US&S, team outputs are monitored and a library of artifacts is maintained for other teams to use. Charters and plans are one example of this. Another example is a survey of some common issues that project teams are faced with. Subjective surveys were developed to help quantify intangible things like feelings and perceptions. Example survey questions include the following:

- Are you comfortable with the schedule?
- Do you feel adequately informed about management decisions?
- Do the managers have the skills necessary to do their job?

These surveys were used on multiple project teams. The survey findings can be used during an improvement effort to measure progress towards a goal, or they can be used after the improvement effort has been implemented to measure the success or monitor the perceived improvement.

The more teams you charter, the better you get at chartering teams. With experience comes efficiency and better ways to solve problems. Recording everything the team does, even if it is a poorly-performing team, is important so that successes and lessons learned can be applied to future teams.

Clear Focus

As mentioned above, US&S teams concentrate their initial activities on developing a charter and plan for the team. This is based on two premises derived from lessons learned of past teams:

- Teams that excel share a sense of purpose.
- Teams that fail or flounder often attribute their troubles to an unclear purpose and ambiguous goals.

A well-defined charter

- clarifies what the team is expected and authorized to do
- provides a basis for setting goals and detailed planning
- focuses the energies and activities of the team members
- communicates the team's purpose to others

The example illustration below shows a typical flow of how a charter can take a list of issues, prioritize them, and focus them on some defined areas for improvement. Each team starts out with a set of issues that it must address. These issues may be the findings from a process appraisal, the results of a survey, or it may be a brainstormed list that the team generates as its first activity. The team then systematically prioritizes these issues into categories of which the Priority A list are the ones that they will work on first. The charter and plan then focus these issues on a well defined set of goals and objectives to be achieved. The illustration shows that this particular team had three main categories of issues to be addressed, namely reality, project crisis, and lifecycle. Reality issues were further categorized into issues related to roles, skills, communication, and planning on the project. These were the issues that the team addressed first. The figure also shows that the top two or three categories of issues were addressed, the full set of issues would be revisited. This is because time passes, circumstances change, and the other issues may no longer apply.

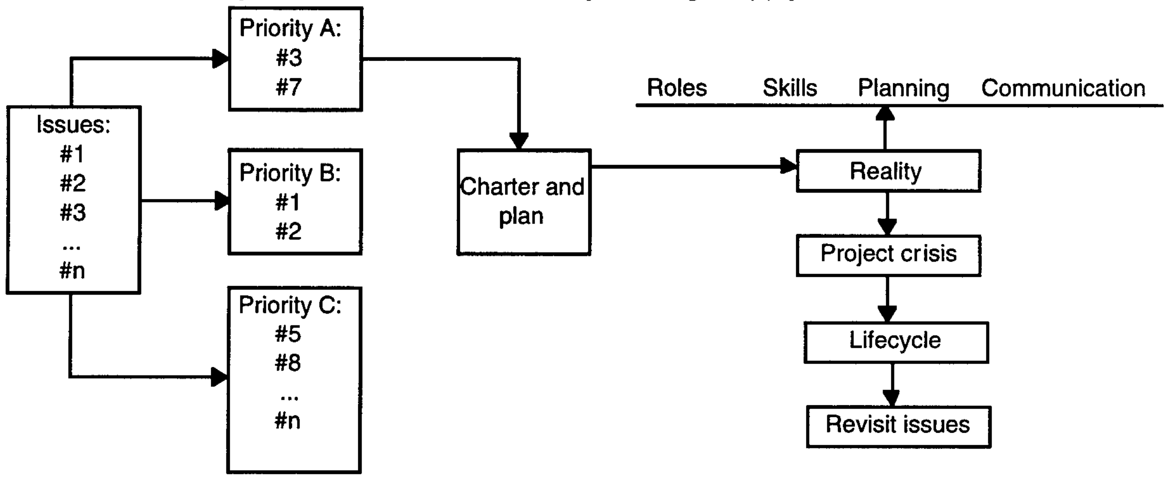


Figure 1. Focusing the Team

It should also be noted that teams may be chartered by management. Management can give a team a very specific goal statement as opposed to a large list of issues such as is illustrated in Figure 1. In this case, chartering may be easier, but just as important to get the team headed in the same direction. Management must be clear in setting these goals, and expectations must be well understood by both management and the team.

Quantification of Costs and Benefits

The clear focus and direction captured in the charter identifies what the team wants to accomplish, gathering the team members together around a set of issues. But this is not enough to ensure success. Teams often don't know how to define success. Turbo-teams quantify costs and benefits as described below to help determine what impacts are being made by the team, and what the value of the team is to the organization.

The costs of a team are straightforward. Hours spent operating the team and the dollars that the team will spend must be captured and tracked. Some examples are listed below.

Hours spent on the following:

- meetings, preparation, minutes
- preparing, reviewing deliverables
- evaluating tools, training
- installing the process

Funds spent on the following:

- tools for evaluation
- team tools (e.g., *The Memory Jogger* [Brassard 91])
- training
- outside consulting

Some other things to consider when staffing a team include

- opportunity cost - what other work could the team members be doing?
- transition costs - the cost of getting the organization to adopt a new technology.

The benefits of the team are not always as straightforward. Some teams are tasked to improve productivity, or reduce problem reports—those may be easy to quantify. But other teams are tasked to put a key process area (KPA) in place (such as requirements management in the software Capability Maturity Model [SW-CMM^{sm}]), or improve on an ad-hoc process; these results are more difficult to quantify. The team is faced with the problem of re-defining a nebulous, uncontrolled process and fitting it into the organization's process. Quantifying the initial state as well as the desired state is key so that the team can plan for and measure progress towards the goal.

Once a team has produced a product such as a process, technology or tool to be installed in the organization, that product must be transitioned into the organization. Transition costs are often not accounted for. Many times they are not understood until the organization has operated a few successful teams. It is necessary to know what it took to get a solution in place—for example, training, guidebooks, mentoring, etc. If your organization does not have

^{sm} CMM and Capability Maturity Model are service marks of Carnegie Mellon University.

examples to draw from, reports such as this one or benchmarking with a partner may provide helpful information.

Teams members are often technical people. Quantification helps the team understand the business ramifications as well as the technical ramifications of the team's work. It helps the team relate the technical goals to the business objectives. This is a key point in getting real management sponsorship.

A final reason for quantification is that the team has set quantifiable goals to achieve. Working toward well-defined goals helps to maintain focus for the group.

Empowerment

The final principle defined for turbo-teams is empowerment. Thus far, we've discussed several principles related to operationally planning and tracking the team, and determining the costs and benefits. However, the critical piece to making the team successful is to empower the team to make changes. The common scenario with teams is for them to meet weekly, build plans, document processes, and put together a suggested set of practices, or even define a process for the organization to use. Although this is often necessary, documenting a process does not effect change. The team needs to be empowered to facilitate the changes, not merely suggest them.

This can be more complicated than just assigning a team and saying, "You're in charge, make it happen." First of all, the team must be given clear and realistic goals. Telling a team that they must raise the maturity level of the organization from level 1 to level 3 in the SW-CMM in the next twelve months may be clear, but is not realistic. At the same time, telling the team that they must improve productivity may be realistic, but is not clear what kind of productivity needs to be improved, or by how much. An example of a clear, realistic goal is as follows:

> *Analyze the existing test processes on projects X, Y, and Z; identify and document the best practice from the combined set of projects.*

Once the team is given clear and realistic goals, they can build a charter and plan that focuses their work toward the end result. During the planning process, they must identify areas where they need support from other parts of the organization, including management. Here they are looking for commitment from the organization to provide the resources and time necessary to carry out this plan.

In summary, the team must be empowered to make decisions and to set directions. They need to be given

- authority to make decisions and act
- control over the work
- accountability for actions
- support for open communication between the team, sponsor, and stakeholders
- access to information

3.3 Team Operating Procedures

Team operating procedures are common in most organizations, and well documented in literature.[5] Basically, teams need to define

- meeting process (standard agendas, meeting minutes, action item and issue database)
- standard roles (leader, facilitator, recorder, librarian, timekeeper, etc.)
- decision making process
- conflict resolution process

For this effort, the team operating procedures were decided upon in the meeting detailed in the minutes included in Appendix D. US&S has operating procedures for working in teams that are documented in the SEPG 96 tutorial, "Turbo-Teaming Toward Improvement" [McAndrews 96]. For details refer to that material.

[5] For example, Scholtes, *The Team Handbook* [Scholtes 88]; Kayser, *Mining Group Gold* [Kayser 90].

4. Software Test Team Experiences and Artifacts

Based on the preceding description of turbo-team principles, it may appear that these teams are focused on quick, short-duration activities to solve a problem that is limited in scope. But the problems that most organizations face are typically quite complex. While the activities may be quick, they are rarely simple, and not always of short duration. The turbo-team approach can be adapted for larger technical area solutions such as installing a software test process.

The prototype *Process Change Guide* was used to apply the turbo-team approach to the testing process improvement. Using the prototype *Process Change Guide*, the test improvement effort was broken down into small, well-defined steps. The turbo-team principles were used to systematically address each step.

Figure 2 below illustrates a block diagram of the stages that were executed by the software test team. These stages are adapted from the prototype *Process Change Guide*. The primary adaptation includes the development of a solution and pilot testing very early on in the software test team effort. This was a necessary adaptation because the team needed to show progress very quickly while continuing to meet the testing needs of projects. The following sections detail each stage of this model.

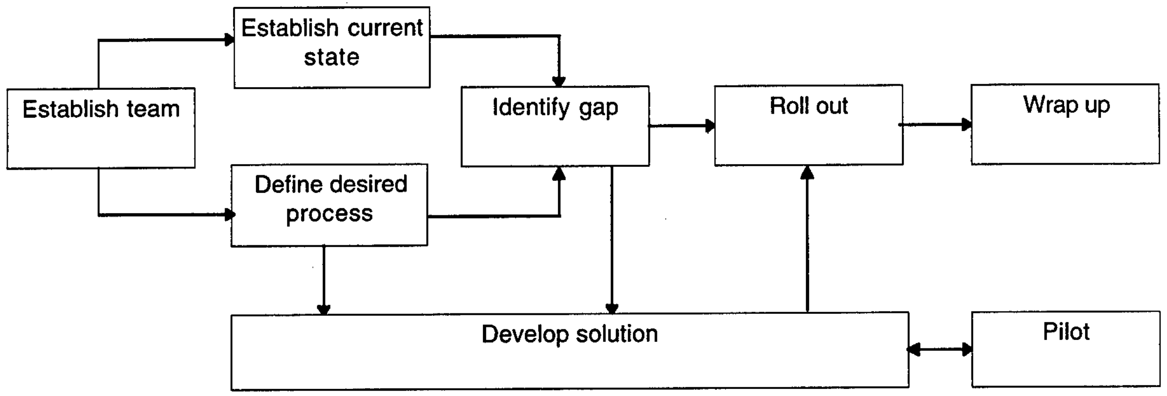


Figure 2. Block Diagram of US&S Turbo-Team Activities

This section describes each of the stages identified in Figure 2 as they were implemented by the software test team. Each section discusses a stage in terms of what was accomplished, when it was accomplished, what was produced, and lessons learned.

4.1 Activity 1: Establish Team

The purpose of this phase is to establish the team and set up working relationships for the tasks to be accomplished. Tasks in this phase include assembling the team and producing a plan that documents a direction for the team.

Summary

Several working sessions were conducted to set up the technical collaboration with the SEI and also to do some preliminary planning and preparation for the kick-off meeting.

There were two team meetings during this phase. The first meeting was a collaboration kick-off meeting with the SEI, the US&S SEPG, and the people who would become members of the software test team. At this meeting, participants reviewed some of the documentation of historical needs for testing, and the approach that would be taken using the prototype *Process Change Guide*. A draft charter and plan was presented at the first meeting.

The second meeting took place the following week with the software test team and an SEPG facilitator. This was the chartering and planning meeting. The purpose of this meeting was to review the draft charter and plan, and to define operating procedures. A team leader was chosen and meeting processes were defined. For example, the team agreed that decisions would be made by consensus. When consensus could not be achieved, decisions were to be made by majority rule. No decisions would be made without a quorum, which was defined as the team leader plus at least half of the other team members.

From this meeting, the charter and plan were finalized and distributed to the sponsors and the SEPG for review. The sponsors approved the charter, and the plan was accepted as working document. ("Working document" is an informal term at US&S referring to a document that is continuously revisited and updated to reflect the current situation.)

Artifacts

The team charter and plan were produced and approved. The charter was signed by all team members and sponsors, and the plan was accepted as a working document.

The charter is included in Appendix A; the plan is in Appendix B. A sample agenda and the minutes from the Operating Procedures meeting are in Appendices C and D. At US&S, the SEPG has standard team training that defines the operating procedures in a meeting, and this was used to define the operating procedures for this team.

Lessons Learned

Chartering and planning teams is a process that is well defined at US&S. This team also had the benefit of having a draft charter from a previous testing working group that was conducted in a prior year (described in Section 2.1). Many of the same issues were still applicable, and the team built on that previous charter rather than developing a new charter from scratch.

Having an SEPG facilitator for the team was very effective in getting it launched. This helped the team gain maximum leverage from the US&S standard team operating procedures, including the chartering and planning process.

The activities described in the prototype *Process Change Guide* proved to be valuable in creating a good list of activities to be incorporated into the plan. The prototype guide helped the team to look ahead to develop the plan.

The kick-off session with the SEI reinforced the team's purpose, and having an outside party involved provided evidence that management sponsorship was real. This showed the team that the organization was committed to the effort and that the methods that would be used were based on the experiences, successes, and lessons learned of other improvement teams.

One issue identified at this point was the lack of a formal infrastructure for process improvement.[6] For example, the lack of a management steering committee impacted the team in several ways. The most significant effect was that the software test team lead was forced to meet with the managers of each team individually to discuss the effort, get their support and buy-in, and negotiate for their time. Throughout the effort, status had to be reported to individual managers; it would have been more efficient to report progress to a steering committee.

Finally, the team wanted to see results fast. They did not want to be involved in a six-month "improvement effort" without seeing some real results. They wanted to see changes made on the projects. Team members agreed that they would move as quickly as possible through the prototype *Process Change Guide*, while paying attention to the specific activities, but trying to expedite implementation. This affected the pilot stage most directly, as discussed in Section 4.6.

4.2 Activity 2: Define Desired Process

The intention of the team during this phase was to define the desired state for the test process. This desired state was used as a target to be measured against. During this phase, the team set out to define the test process to be implemented within the organization. The desired process was compiled from US&S project best practices, industry standards, and findings from previous appraisal efforts at US&S.

[6] The necessity of a formal infrastructure for process improvement has been widely discussed. See, for example, *IDEAL^{SM}: A User's Guide for Software Process Improvement* [McFeeley 96]. IDEAL is a service mark of Carnegie Mellon University.

Summary

Creating the desired process consisted of defining the test process requirements and drafting a test process specification. The process requirements specification was a brief summary of the requirements that a software test process must meet. It represented the minimum set of requirements that must be implemented on a project. This specification was analogous to a CMM KPA, which contains requirements for process areas such as requirements management or configuration management. The test process specification was a more detailed, step-by-step guide to implementing the test process, including roles, artifacts, and activities that make up the test process.

The team worked to create test process requirements during one meeting. Again, draft materials were prepared and distributed for review before the team meeting. This phase was unusually brief because the team had information from the earlier testing working group that it was able to reuse. Requirements were drafted from a previous testing working group and a software process appraisal conducted in 1994. The draft process specification was produced based on best practices on two of US&S's larger projects.

The test process requirements specification was produced and documented to resemble a KPA. As noted above, the format was taken from the SW-CMM and included attributes defined for commitment to perform, ability to perform, activities, monitoring, implementation, and verifying implementation. In this requirements specification, requirements were included that had been identified in previous appraisals as well as those noted as deficiencies from US&S testing activities. The KPAs from the CMM at levels 2 and 3 were also reviewed to extract any requirements that relate to testing.

The test process specification was drafted based on the processes in place on two projects whose processes were very consistent. The process was documented by the SEPG because they were the only ones with process definition experience.

In a team meeting, a mini-tutorial was conducted on process definition so that the team all understood why and how the process was going to be documented. The format was to be kept consistent with the US&S Software Engineering Handbook.

Artifacts

The requirements specification and process specification are included in Appendices E and F respectively.

The desired testing process evolved significantly throughout this effort. The testing process was drafted initially as a target for pilot testing. It was also important to the acceptance of pilot efforts that the team have something available to implement on projects as each project had needs and milestones that had to be met.

Another artifact produced at this point was a list of issues related to installing processes in an organization, similar to the whole product concept.[7] The whole product "wheel" illustrates that documenting a process does not get it installed; there are a number of other factors associated with facilitating the installation of a process. Training on the process, tools, mentoring, and management briefings must be considered in addition to documenting the process. In preparing the whole product wheel, the team attempts to identify all of the things that are necessary to successfully install the newly-documented process. Appendix G contains the whole-product wheel items that were identified by the software test team. Putting this list together early on in the effort helped the team to recognize that there were many things besides defining a process that needed to be done to get the process installed.

Lessons Learned

One of the drawbacks to "turbo-teaming" in an effort like this is that the team will try to cut corners where possible to get things done quickly. One corner that was cut was not giving the team formal "process improvement" training or "process definition" training. Thus, when the SEPG drafted the test team process specification, it was not easily understood by the team at first. The team learned that both the need for and the use of process documentation is not intuitively obvious to software developers who have no experience with process definition models. This was raised as a risk.

Also, the whole-product wheel concept was difficult to understand initially. It represented ideas that didn't mean much to team members at that point in the turbo-teaming process. The team did not spend a lot of time trying to think too far ahead; they recognized the need for training and tools at this point, but also recognized they could address them later.

The fact that the team was not receiving much feedback from reviewers (sponsors, SEPG) on the distributed materials was both good and bad: while this enabled the team to move rather quickly through the prototype *Process Change Guide*, it also raised a flag that perhaps there was neither buy-in nor support for the effort.

4.3 Activity 3: Establish Current State

This phase was intended to document the current state of test processes on projects. Once the process requirements specification was produced, there was a target to measure against. The team looked at existing projects and defined their current state with respect to the desired state. There has been some debate in the industry on the topic of defining the desired and current states. While some people with process improvement experience feel it is appropriate to establish the current state first, others feel that it is important to define the desired state initially. Our experience supports the notion that the desired state should be defined before the current state is established: having a defined desired state made it easier

[7] Geoffrey Moore, *Crossing the Chasm* [Moore 91].

to describe the current state because it provided the team with a reference model and a common understanding of what the team was trying to accomplish.

Summary

During this phase the team completed a process baseline matrix and collected data used to establish performance measures. One meeting was conducted to discuss this data and achieve consensus on the process baseline.

The process requirements specification was used to help generate the process baseline matrix. Each requirement was listed on a spreadsheet; then, each team member evaluated his or her project against the requirements. Several team members had worked on more than one project, so they were asked to complete the template for all projects they were familiar with. Figure 3 illustrates part of this matrix. The complete matrix is included in Appendix H.

		Projects Assessed								
Requirement Section	**Description**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Org. Level**	**Resp. Party**
Commitment 1	A dedicated team is assigned to software testing and is staffed with qualified test engineers.	yes	yes	no	yes	inc	no	inc	no	Mgt

Figure 3. Sample Process Baseline Matrix

A row was entered in the spreadsheet for each requirement. The projects 1, 2, 3, 4, etc., represent projects that members of the software test team either worked on or were familiar with. Each team member completed the templates outside of team meetings; then the results were compiled in a matrix such as the one illustrated in Figure 3. In the next team meeting, each row was reviewed for consistency. These matrices plus group voting were used to reach consensus for the organization level. If all projects answered *yes*, then the organization level would be *yes*. If all projects answer *no*, then the level would be *no*. However, in most cases, there were some *yes*, some *no*, and some *inc* (incomplete). The team discussed discrepancies and arrived at a consensus for the organization. When the organizational level answer was *no*, the team decided who was responsible for remedying that problem: management, the software test team, or another group. This information was expanded upon in the identify gap and risk mitigation activities of Activity 4.

The other activity completed during this phase was the collection of historical data on the test process. Although there was considerable historical defect data available, it hadn't been collected consistently enough to allow a baseline to be established. This was recognized as part of the process that must be defined for future analysis of defect data. The primary data of interest was data that could be used for planning purposes and for forecasting future project testing needs.

Artifacts

The process baseline matrix is included in Appendix H.

Lessons Learned

The baseline matrix was interesting, but represented a narrow and subjective point of view. The data from the team members was inconsistent, even where the same project was being appraised by more than one person. However, consensus was reached at an organizational level and an organizational baseline was established. This meant that although the team might disagree on individual projects' testing processes, everyone agreed on the organizational level and was clear on what needed to be done.

As part of this phase, historical data was collected on testing activities across projects. This data was helpful because it was consistent for planning purposes. This provided the team with confidence that they could accurately predict what would be needed on future projects in the area of testing.

On the other hand, defect data was found to be inconsistent on past projects. The software test team recognized the need to come up with a better, more consistent way to track problems from testing and from the field.

4.4 Activity 4: Identify Gap and Analysis

In this phase the change team looked at the current state with respect to the desired state and identified gaps. At US&S this started with the completion of the process baseline matrix by identifying which parts of the process were not being implementing organization wide. In this phase, the software test team took a close look at each of these deficiencies, analyzed root causes, and established risk mitigation strategies.

Summary

This phase began shortly after completing the process baseline, which identified deficiencies in the software test process. The team then dedicated one meeting to a fishbone analysis of the deficiencies to determine the causes. However, once the root causes were identified, it was several months before agreement was reached that the causes were addressed adequately by the risk mitigation strategies. This was a difficult process because the team did not have much experience in documenting risks or determining mitigation strategies. Ultimately, the risk mitigation matrix was completed with assistance from the team's SEI coach. This phase was completed in pieces over a period of three months.

Each organizational deficiency from the process baseline matrix was put on the fishbone to analyze and determine root causes. Root causes were analyzed in terms of commitment, requirements, abilities, configuration management, measurement, verification, subcontractors, and general activities.

Another team meeting addressed risk statements, writing them in the format of "If <situation> occurs, then <result> will happen." These are listed in the risk mitigation matrix in Appendix H. Only after some subsequent meetings with the SEI coach did it become clear what the mitigation strategies would become. These risks were re-addressed in the roll-out phase once the team was established for the long term as a testing function in the organization.

Artifacts

Appendices I and J include partial copies of the fishbone diagram and the risk statement matrix.

Lessons Learned

The fishbone diagram provided the team with a clear, concise explanation of the process deficiencies, and proved more useful than the process baseline matrix. After the fishbone exercise, the team felt more confident that they understood the process and the problems now facing the organization; it was clear what the connections and interdependencies were. Also, the fishbone exercise proved to be a good team-building exercise. All members were very active in this exercise.

4.5 Activity 5: Develop Solution

The next phase in the prototype *Process Change Guide* is to develop a solution based on the gaps. At US&S, a slightly different approach was taken than what is described in the prototype *Process Change Guide*. Developing the solution became an evolving process that started when the current state was defined in Activity 2.

Summary

One meeting of concentrated effort was held to review the process specification to date and to discuss additions to the whole-product wheel.

Artifacts

The process specification and the whole-product wheel were updated. These are included in Appendix F and G, respectively.

Lessons Learned

Implementation materials for the long term, such as training, were not developed here. The team simply prepared process materials such as procedures, tempates, and a database to do pilots, and planned to support each active project.

The team did not have a formal process for capturing changes to the process. They were relying on the team members to keep the process up to date. The team felt that this might become a problem when the test team began supporting more projects. In retrospect, the team felt it should have defined a process for addressing process changes as part of the whole-product wheel.

4.6 Activity 6: Pilot Use and Evaluation

In this phase, the test team installed elements of the test process on a limited number of projects and evaluated their use. One of the things that contributed to the success of the software test team effort was the existing need for testing on projects, which facilitated the pilot process. There were several projects at different phases of the testing process to be worked with in piloting.

Summary

Pilot testing is often a source of resistance within organizations and especially in process improvement efforts. At US&S, pilot testing was perceived as an academic exercise, and as representing additional cost to the project. This is one perception that the test team clearly had to address; projects could not afford to "pilot test" the testing process at their own expense.

Because of this, the software test team took an approach that pilot testing would be done on projects that had specific needs. Projects that were in the test planning phase would pilot test the test planning process; projects that were actively testing would pilot test the new procedures for executing tests and reporting test results. In other words, all phases of the testing process were pilot tested on multiple projects simultaneously.

The only drawback to this approach was that the test process was not applied from end-to-end on one project. However, this is not unlike the situation that will occur when the testing process is implemented across all projects. The projects will all be at different phases of development. The testing process must be flexible enough to work with projects that already have things in place, or customer constraints that would require tailoring of the testing process.

To execute the pilots, team members were assigned a different part of the process and asked to develop pilot plans. The team members were assigned to pilot test that part of the testing process that they were currently working on the project that they supported. Pilot plans were documented to clarify the scope and purpose of the pilots, and to communicate to the project managers 1) how the pilot efforts "support" their project effort, and 2) that the pilots do not represent additional work to them. Examples and a template for the pilot plans are included in Appendix K.

As a result of the pilots, each individual who executed the pilot updated the documented test process with lessons learned. Although there was not a formal review of pilot results with the project managers, their informal feedback was captured as a result of the work that the software test team performed for them. Primarily, their feedback related to the usefulness and adaptability of the artifacts that were produced, such as system problem reports, test reports, test time logs, and other related test metrics.

Artifacts

Appendix K contains a template for the pilot plans and three examples of pilot plans that cover different parts of the testing process.

Lessons Learned

Pilot testing is critical. Anyone who works on process improvement has seen defined processes "thrown over the wall." That approach, defining a process and then just assuming the engineers will follow the documentation without training, coaching, etc., typically fails. It is essential that processes documented on paper reflect and address the realities and needs that exist in a project environment.

The pilot efforts on this testing process were no exception. The software test team started small, pilot testing individual pieces of the testing process on different projects. From this, the process evolved considerably. If an attempt had been made to roll out the testing process on all projects at once, the way it was initially documented, there would have been confusion, resistance, and eventual abandonment.

As a result of the well-planned, focused pilot efforts, the process can now be revised to work with other projects systematically because there is documented knowledge of what works and what doesn't work. US&S projects are much more receptive to process change when they understand how it has benefited a prior project. This early success gave the software test team credibility and also momentum for rolling the test process out to other teams.

Another major benefit of pilot testing is process data. Data was added to the US&S historical testing metrics database and a better understanding of the test process was gained related to the following:

- Staff-hour and schedule data: How much time does testing/should testing take? How much calendar time should be allocated to testing?
- Defect data: How many defects are/should be found? What kind of defects are/aren't being found?

The software test team was also able to identify some areas for improvement in other parts of the software development process, such as lower levels of testing, as well as installation and field testing. These areas could be natural extensions of the testing process that we have defined.

4.7 Activity 7: Roll-Out

During this phase the improvement team transitions the process to the rest of the organization. However, in this case the members of the improvement team actually became the test team. This phase consisted of putting test plans in place on each project.

Summary

After the software test team reached a level of comfort with the testing process, they were in a position to systematically begin to put the testing process in place on other projects. At this point, there were some problems with employee turnover. Three members of the test team that were dedicated to testing on projects left the company (the original team had eight members). At the time the team suffered the loss of three staff, there were open job requisitions to staff the team to meet the needs of other projects.

US&S was able to recruit and fill these positions over a period of about two months. Four new members from outside the organizations were added to the team. This was a true test of the defined testing process as the team worked to integrate its new members.

As the test team was staffed, work plans were developed for other projects using the metrics data gained from the pilot efforts. Once the team was fully staffed, the plans were integrated into one plan to provide visibility across projects on where the needs existed, and where coverage was needed. At US&S, test resources and test efforts on projects tend to be cyclical. As projects are developing plans and writing requirements, there is a need for a dedicated resource to plan testing. As projects develop their software, test team resources are dedicated to writing test cases and procedures. Once development is complete, testing resources are applied to test the software in the lab, and in some cases in the field. Each of these unique activities requires different levels of support. On US&S projects, this ranges from a part-time level of support to document a test plan to six or seven full-time engineers needed to fully test a system in the lab. Having a dedicated team for testing enables US&S to apply resources more effectively.

An attempt was made to develop some test process training. Two weekly meetings were dedicated to designing instruction and generating a list of instructional needs. This list is included in Appendix L.

There was some benefit in generating this list because the test team then recognized what capabilities were needed by engineers to perform as testers. Although no resources have been dedicated to develop the training to date, some effort in the future will be dedicated to this task.

Artifacts

Work plans for projects were developed as well as an integrated resource plan that identifies the projects and needs for test resources. Needs were identified for training, but the training was never developed.

Lessons Learned

Having a well-documented testing process had several benefits at this point in the effort. With new team members being recruited from outside the organization, the documented test process enabled interviewers to clearly describe to potential candidates how testing would be done. Because of the documented process, it was possible to clearly discuss with them how they could support the team in the specific activities related to testing. And once hired, the new team members had a documented process to study and learn from.

In addition, as work proceeded with other projects, more detail became available on how project needs differ. Thus, additional experience was gained on tailoring the test process.

As a result of increased activities supporting projects, the software test team improvement meetings began to taper off, and eventually diminish. The meetings that were held after the roll-out phase were more like staff meetings, where project progress was reviewed. This represents a weakness in that the improvement focus may have been lost, at least in a structured sense, because there was no clear transition. However, there also were significant benefits gained from having the team meet and discuss and review project testing efforts. Valuable insight was gained into problems that were occurring on every project, and team members were better able to adapt and look for these problems before they surfaced. The software test team was also able to meet the needs of projects more effectively, especially in times of crisis, because of the cyclical nature of testing. When projects had periods of high testing activity, those needs were met by moving testing resources around. This was definitely easier to accomplish than in the past when testers were assigned to only one project.

4.8 Activity 8: Wrap-Up

In the prototype *Process Change Guide*, wrap-up represents an end to the effort. The US&S effort never really "wrapped up" because the people on the improvement team were the same people that eventually became the software test team.

Summary

In the case of the US&S testing improvement effort, the wrap-up is this report. This report documents successes and lessons learned, and offers suggestions to future improvement efforts.

Artifacts

This report is the only artifact from this phase. At a future date, the software test team will document the test process in the US&S Software Engineering Handbook.

Lessons Learned

No significant lessons learned resulted from our implementation of the wrap-up phase. Note, however, that it is important to look for the following during the execution of an improvement effort like this:

- successful progress or completion of phases (things are going well)
- unsuccessful completion of phases (things are not going well)
- not completing phases (things are not going—the effort is stalling)

In any case, the team needs to recognize signs that would indicate an end to the effort. Although some cases result in celebration (success!), other teams must recognize failure, and know when to quit expending resources. Even in the worst failures, lessons can be learned and applied to the next effort.

5. Key Issues

The installation of a software test process at US&S is complete and operational, so the effort was successful. Previously, each project team conducted its own testing. Currently, there is a central group of test engineers whose time is purchased by project teams to do system testing on many US&S projects. The goal was to have experienced test engineers doing testing in a methodical and uniform manner, and that has been accomplished.

With respect to the charter, the team was successful in meeting two objectives, partially meeting some, and not really addressing others. The major accomplishments were

- developing process documentation and example artifacts for the test process
- pilot testing and evolving the testing process
- getting software testing into practice on projects

Some areas that were not addressed include

- clarification of roles of management, QA, and testing within the organization (and the software release process in particular)
- training on the process (the team did not allow for time to adequately develop formal training)
- tools—the team did little to expand the toolset for testing

Progress, and lack of progress, was taken into account when the team chartered itself for the following year.

Using the prototype *Process Change Guide* gave the improvement effort a structure that is repeatable for future improvement efforts. However, the prototype *Process Change Guide* was just that—a prototype—and therefore not perfect. Nevertheless, the prototype *Process Change Guide* offered an excellent checklist of things to consider in each phase.

A key benefit is that US&S can now directly apply the defined process to lower-level testing activities. With the system-level test process now defined, it is simple to apply the process to integration and unit test activities. Having a defined process also puts the test team in a position to look at automated tools that could be incorporated in the process to increase productivity and quality.

Limitations include the lack of a process improvement infrastructure at US&S, which puts the test team at risk. Because the required infrastructure is not in place, there is no systematic way to determine which projects are in need of test team support—the team still relies on individual relationships to identify its client projects. A company-wide policy insisting on the use of test team support would greatly increase the effectiveness of the test team.

Also, future teams using the prototype *Process Change Guide* or its successors must realize that the resulting improvement will only be as good as the expertise on the team. The guide offers a general process applicable to any technology area similar to the Level 2 KPAs of the software CMM. This effort was a success because the team members understood what testing is and how to do it effectively and efficiently. For example, at US&S, there is currently a planning, control, and configuration team in place attempting to define a standard process; but unless the team is staffed with members who understand how to plan, control, and configure, the team is likely to fail in its efforts.

References

Brassard 91 Brassard, Michael. *The Memory Jogger: A Pocket Guide of Tools for Continuous Improvement.* Metheun, Ma.: GOAL/QPC, 1991.

Kayser 90 Kayser, Thomas A. *Mining Group Gold: How to Cash in on the Collaborative Brain Power of a Group.* El Segundo, Ca.: Serif Pub., 1990.

McAndrews 96 McAndrews, D.; McDonough, B.; & Mavya, A. "Turbo-Teaming Toward Improvement." *Broadening the Perspective for the Next Century: 1996 SEPG Conference.* Atlantic City, N.J., May 20-23, 1996. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1996.

McFeeley 96 McFeeley, Robert. *IDEAL: A User's Guide for Software Process Improvement* (CMU/SEI-96-HB-001). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1996.

Moore 91 Moore, Geoffrey A. *Crossing the Chasm: Marketing and Selling Technology Products to Mainstream Customers.* New York: HarperBusiness, 1991.

Scholtes 88 Scholtes, Peter R. *The Team Handbook: How to Use Teams to Improve Quality.* Madison, Wi.: Joiner Associates, Inc., 1988.

Appendix A: Software Test Team Charter

UNION SWITCH & SIGNAL

Charter for Operation

Software Test Team Charter

Scope

This charter identifies and describes the function of the Software Test Team for the Union Switch and Signal (US&S) Automation and Information Systems (A&IS) Business Unit.

Mission

To improve software validation testing practices by focusing on defining and installing a process that incorporates lessons learned and is rigorous and tailorable. The validation testing process will support effective and efficient validation testing throughout the entire product lifecycle.

Guiding Principles

Test Benefits - An effective testing process should be established for our own benefit to ensure the quality of the systems we deliver and the re-usability of test products among projects.

Early Defect Detection - The Software Test Team is focused on taking steps in the early phases of the lifecycle to ensure that testing goes smoother in the later stages of the lifecycle and defects are found as early as possible.

Quality - Software products are not to be released until rigorous testing processes have been executed, and established criteria have been met for software release.

Responsiveness - Technical expertise in the area of software testing will be provided to address the immediate needs of projects.

Objectives

Management: Define test management, QA, and Software Test Team roles and activities to promote adherence to the testing process stages and release criteria.

Planning: Define detailed test activities and roles during the planning stages of projects, including standard metrics and work breakdown structure elements.

Process Documentation/Engineering Steps: Develop documentation which specifically lists guidelines, examples, and steps required to perform all phases of the software test process including test case development, regression test, performance test, and managing and reporting problems.

Training: Provide training to build skills and expertise on the testing process for effective use within the US&S software development process.

Tools: Establish a toolset that supports the engineering steps for all levels of test planning, documenting, executing, and reporting, including test case management tools.

Usage: Get software testing into practice through needs analysis, pilot testing, and feedback from the users.

Deliverables

For a detailed list of deliverables, please refer to the Software Test Team Plan. The following are the types of deliverables the team will produce

- Team Charter
- Detailed Plan
- Agendas and Minutes of team meetings
- Documentation of a test process that could be used to update the SEH
- Marked-up process change guide to be submitted to the SEI
- Modified process change guide that can be used by other US&S improvement teams
- Testing Process Training
- Lessons Learned

Membership

The current test team members are:

Member	Project Represented	Signature
Don McAndrews	Team Lead/Tri-Met	
Janice Marchok	SEI	
Michelle Brown	MBTA	
Rajan Sharma	MBTA	
Dave Majernik	Boden/Hamersley	
Chris Leya	Boden	
Tom Regola	LA	
Craig Mamone	UP	

Authorization

The Software Test Team will be organized as a project within the A&IS Business Unit. The Test Team lead is Don McAndrews, who reports directly to the Engineering Manager of the SEPG. This charter is approved as written.

Sponsor: Robert M. Elder, Director, A&IS Date

Sponsor: Nadine M. Bounds, Engineering Manager, SEPG Date

Appendix B: Software Test Team Plan

		Planned			Actual				Earned Value		
Activity Name/Description	Resp.	Hours	Start	End	Hours	Start	End	% Com.	Plan	Actual (partial)	Actual (complete)
A1: Establish Team						2/5/96	2/26/96	100%			
Negotiate with managers of representatives	DRM, JMM	6			4	2/5/96	2/9/96	100%	1.68%	1.68%	1.68%
Prepare for kickoff meeting	DRM, JMM, PJF	12			3	2/5/96	2/9/96	100%	3.35%	3.35%	3.35%
Conduct kick-off meeting	Team, SEPG, PJF	22			24	2/9/96	2/9/96	100%	6.15%	6.15%	6.15%
Draft Charter and Plan	DRM, JMM	12			8	2/5/96	2/15/96	100%	3.35%	3.35%	3.35%
Prepare for Operating Procedures/Charter/Plan Mtg	DRM	1			2	2/12/96	2/12/96	100%	0.28%	0.28%	0.28%
Conduct Operating Procedures/Charter/Plan Mtg	Team, JM	16			18	2/16/96	2/16/96	100%	4.47%	4.47%	4.47%
Review Charter and Plan, Sponsor Approval	SEPG, RME, NMB, team mngrs	5			6	2/27/96	2/27/96	100%	1.40%	1.40%	1.40%
Lesson Learned, Kick-off to A2	DRM, JMM	4			2	2/26/96	2/26/96	100%	1.12%	1.12%	1.12%
A2: Define Desired Process			2/19/96	3/3/96		2/19/96	4/2/96	100%			
Compile Test Process Requirements	DRM. JMM	4	2/19/96	2/19/96	4	2/19/96	2/19/96	100%	1.12%	1.12%	1.12%
Draft Test Process Specification	DRM, JMM	8	2/19/96	2/20/96	8	2/20/96	2/20/96	100%	2.23%	2.23%	2.23%
Meet to review Test Process Specification	Team	18	2/23/96	2/23/96	14	2/23/96	2/23/96	100%	5.03%	5.03%	5.03%
Complete Test Process Specificaiton	DRM, JMM	4	2/23/96	2/25/96	4	2/24/96	3/5/96	100%	1.12%	1.12%	1.12%
Review and Sponsor Approval	SEPG, NMB, RME	8	2/28/96	3/2/96	2	3/5/96	3/12/96	100%	2.23%	2.23%	2.23%
Lessons Learned and Kick-off to A3	DRM, JMM	2	3/3/96	3/3/96	2	4/2/96	4/2/96	100%	0.56%	0.56%	0.56%
A3: Establish Current State			3/6/96	3/23/96		3/5/96	4/2/96	100%			

		Planned			Actual				Earned Value		
Activity Name/Description	Resp.	Hours	Start	End	Hours	Start	End	% Com.	Plan	Actual (partial)	Actual (complete)
Build a Test Process Specification Matrix	DRM, JMM	4	3/6/96	3/6/96	4	3/5/96	3/5/96	100%	1.12%	1.12%	1.12%
Complete Matrix for each project	Team	16	3/7/96	3/21/96	4	3/6/96	3/8/96	100%	4.47%	4.47%	4.47%
Define and Collect Baseline Measures for Testing Team	Team	8	3/7/96	3/21/96	4	3/13/96	3/19/96	100%	2.23%	2.23%	2.23%
Lessons Learned, Kick-off to A4	DRM, JMM	2	3/22/96	3/23/96	2	4/2/96	4/2/96	100%	0.56%	0.56%	0.56%
A4: Identify Gap			3/23/96	4/3/96		3/13/96	5/30/96				
Fishbone deltas, Why-Why symptoms	Team	16	3/23/96	3/25/96	16	4/22/96	4/26/96	100%	4.47%	4.47%	4.47%
Identify changės, time and scheudule, risks	Team	4	3/23/96	3/25/96	3	5/3/96	5/30/96	100%	1.12%	1.12%	1.12%
Prioritize	Team	4	3/23/96	3/25/96	2	5/3/96	5/30/96	100%	1.12%	1.12%	1.12%
Obtain management review and direction	DRM, JMM, NMB, RME	4	3/26/96	4/2/96	2	5/3/96	5/30/96	100%	1.12%	1.12%	1.12%
Lessons Learned and Kick-off to A5	DRM, JMM	2	4/3/96	4/3/96	2	5/3/96	5/30/96	100%	0.56%	0.56%	0.56%
A5: Develop Solution			4/3/96	4/15/96		4/15/96	6/6/96				
Refine "desired process"	Team	16	4/3/96	4/5/96	3	4/15/96	4/18/96	100%	4.47%	4.47%	4.47%
Prepare implementation plan	DRM, JMM	8	4/5/96	4/12/96	6	4/15/96	4/18/96	100%	2.23%	2.23%	2.23%
Develop implementation materials	DRM, JMM	8	4/5/96	4/12/96	12	5/30/96	6/6/96	100%	2.23%	2.23%	2.23%
Lessons Learned and Kick-off to A6	DRM, JMM	2	4/12/96	4/15/96	2	5/30/96	6/6/96	100%	0.56%	0.56%	0.56%
A6: Pilot Use and Evaluate			4/15/96	5/31/96		4/15/96	6/6/96				
Finalize pilot plan with PE/PM		4	4/15/96	4/22/96	2	4/15/96	4/18/96	100%	1.12%	1.12%	1.12%
Implement			4/15/96	5/31/96		4/15/96	6/6/96	100%	0.00%	0.00%	0.00%
Evaluate		32	5/22/96	5/29/96	24	4/15/96	6/6/96	100%	8.94%	8.94%	8.94%

		Planned			Actual				Earned Value		
Activity Name/Description	Resp.	Hours	Start	End	Hours	Start	End	% Com.	Plan	Actual (partial)	Actual (complete)
Lessons Learned and Kick-off to A7		4	5/30/96	5/31/96	3	6/6/96	6/6/96	100%	1.12%	1.12%	1.12%
A7: Roll Out			6/3/96	6/21/96							
Plan for institutionalization	Team	32	6/3/96	6/17/96	16	6/13/96	6/13/96	100%	8.94%	8.94%	8.94%
Implement, monitor, improve									0.00%	0.00%	0.00%
Lessons Learned and Kick-off to A8	DRM, JMM	2	6/21/96	6/21/96	2	6/13/96	6/13/96	100%	0.56%	0.56%	0.56%
A8: Wrap Up/Termination			6/24/96	6/28/96		6/20/96	10/30/96				
Lessons Learned	Team	20	6/24/96	6/25/96	12	6/20/96	6/20/96	100%	5.59%	5.59%	5.59%
Productize artifacts	DRM, JMM	32	6/26/96	6/27/96	85	6/20/96	10/30/96	100%	8.94%	8.94%	8.94%
Feedback on PCG (summarize)	DRM, JMM	8	6/27/96	6/28/96	2	6/20/96	10/30/96	100%	2.23%	2.23%	2.23%
Potential general PCG for use on next improvement area	DRM	8	6/28/96	6/28/96	2	6/20/96	10/30/96	100%	2.23%	2.23%	2.23%
									100.00%	100.00%	100.00%

Appendix C: Test Team Operating Procedures/Charter/Planning Meeting Agenda

UNION SWITCH & SIGNAL

InterOffice Memo

To:	Test Team
CC:	Test Team Sponsor
From:	DRM
Date:	February 9, 1996
Subject:	Meeting Agenda

Please attend the Software Test Team Operating Procedures/Charter/Planning meeting in Conference room **P2A, 9:00 - 11:00 am, Friday, 2/16.**

Purpose

In this meeting, we will charter our team, define the procedures we will operate by, and plan how we will accomplish our goal cf establishing a software test team.

Agenda Items

- Introduction--summarize where we are (team leader) 10 min
- Team Operating Procedures (SEPG coach) 60 min
- Review and Revise Charter (all) 20 min
- Review and Revise Plan (all) 20 min
- Discuss next steps (team leader) 10 min

Pre-work

Please review the draft charter and plan before coming to the meeting.

Appendix D: Test Team Operating Procedures/Charter/Planning Meeting Minutes

UNION SWITCH & SIGNAL

InterOffice Memo

To:	Test Team
cc:	Test Team Sponsor
From:	
Date:	February 19, 1995
Subject:	Minutes of 16 February 1996 Test Team Meeing

Attendees DRM, JLM, JMM, MLB, RS, DJM, CMM, CML, TVR

Review agenda

The agenda was set as follows:

- Introduction - summarize where we are (DRM)
- Team operating procedures (JM)
- Review and revise charter (All)
- Review and revise plan
- Discuss next steps

Continued on next page

Introduction

Charge numbers have been established for the Test Team as follows:

For TOTAL:

11-9624	Test Group Activities

For POSTINGS:

11-9624-010	Establish Team
11-9624-020	Define Desired Process
11-9624-030	Establish Current State
11-9624-040	Identify Gap
11-9624-050	Develop Solution
11-9624-060	Pilot Use and Evaluate
11-9624-070	Roll-out
11-9624-080	Evaluate
11-9624-090	Miscellaneous

Today's meeting will conclude the Establish Team phase.

Team Operating Procedures

Team roles will be as follows:

Leader/Facilitator: permanant, DRM
Recorder: permanent, JMM/DJM
Librarian: permanent: JMM/DJM
Scribe: rotating
Timekeeper: rotating

Meetings will be held weekly: typically from 9 - 11 am on Fridays. DRM will book the room, get food, and prepare the agenda. It was decided that

- Decision making will be done via consensus.
- Conflicts will be resolved by majority rule.
- A quorum will be defined as Don or Janice plus four other team members. No meeting will be held nor decisions made unless a quorum is in attendance.

Review and Revise Charter

Charter was reviewed. All recommended revisions have been made to the attached version.

Continued on next page

Review and Revise Plan

The Plan was reviewed through section 1.5. All recommended revisions have been made to the attached version. Review will be completed at the next meeting, scheduled for Friday, 23 February, 9 - 11 am.

Discuss Next Steps

Next steps:

- Update Charter (done--see attached)
- Have RME and NMB review and sign charter
- Finish updating Plan

Meeting Evaluation

Strengths:

- went well
- finished charter
- met objectives
- pace was good
- independent facilitator helpful
- good participation

Weaknesses:

- none reported

Appendix E: Software Test Team Requirements Specification

Software Test Team Requirements Specification

The following requirements are to be used to help define the testing process for the organization. These requirements are described in categories as follows:

Definition - What the team is.

Commitment - Describes the organizational commitment to having and using a test team.

Ability - Describes the staffing, funding, training, and tools necessary to give the team the ability to perform testing activities.

Activities - Describes the activities that the software test team should perform on each project.

Measurement and Analysis - Describes the instrumentation of the testing process so that it can be analyzed, managed, and decisions can be made on testing-related issues.

Verification - Describes the management and oversight roles that must be in place to ensure that the testing on projects adheres to the testing process as defined.

Next steps are as follows:

- Examine current testing on projects (MBTA, BN, UP, LA) with respect to the requirements - this will give us a baseline from which we will improve upon.
- Define the organization's testing process to be followed in future (on Hamersley, Tri-Met, Boden, beyond).

Definition of Team

The Software Test Team is the collection of individuals (both managers and technical staff) who have responsibility for planning and performing the independent (with support from the developers) system testing of the software to determine whether the software product satisfies its requirements and quality criteria. The need for independence of system and acceptance testing is based on technical considerations. This independence ensures that the testers are not inappropriately influenced by the design and implementation decisions made by the software developers or maintainers. Although the software test team is chartered to perform system level testing, there is an overlap with software integration testing that the team will also support to some extent.

Organizational Commitment to Software Test Team

A dedicated team is assigned to software testing and is staffed with qualified test engineers.

Managers establish quality and testing criteria with the test team and use testing data to make decisions.

Quality and testing criteria are developed and reviewed with the customer and the end-users.

Test readiness criteria are established and used on projects to determine when the lifecycle can proceed (e.g., when to proceed to the next phase, or when to ship software).

Resources for testing the software are assigned early enough to provide for adequate test preparation.

Resources are also available from the developers to work with the testers to ensure a thorough test suite.

Ability to Perform

On each project, staffing and funding are allocated to software testing in accordance with project needs to provide training, tools, and perform the testing activities.

Training is available for software testing techniques including verification methods, test planning, use of tools.

Tools to support testing are available, e.g., test management, test generators, test drivers, symbolic debuggers, test coverage analyzers, capture playback.

Procedures exist to be adapted onto each project so that the organizational test process is consistently implemented.

Activities

General Testing:

System testing is planned and performed to ensure that the software satisfies the software requirements and quality criteria.

Acceptance testing is planned and performed to demonstrate to the customer and end-users that the software satisfies the allocated requirements when required.

System and Acceptance testing are documented in a project test plan which is reviewed with and approved by the customer, end-users, and managers.

The test plan covers the overall testing and verification approach, responsibilities of the developing organization, subcontractors, customer, and end-users, test equipment, facilities, and support requirements, acceptance criteria.

Test cases and procedures are planned and prepared independent of but with support from the software developers.

Test work products undergo peer review, including within the software test team.

Test cases and procedures are documented and reviewed with and approved by the appropriate individuals (e.g., customer, end-users, managers, depending on the project) before testing begins.

Testing of the software is performed against baseline software and the baseline documentation of the allocated requirements and the software requirements.

Problems identified during testing are documented and tracked to closure.

Test work products are re-used across projects.

Adequate regression testing is performed when software and/or environment changes to ensure that changes have not caused unintended effects on the baseline.

Requirements Management:

Requirements allocated to software are reviewed by the STT to determine whether they are testable.

Consistency is maintained across software work products (including testing) via traceability matrices.

When requirements change, test work products are updated accordingly.

Software Configuration Management:

Test work products are identified as configuration items by the project.

Test work products are maintained in accordance with documented configuration management procedures/plans.

The test environment, including software, is controlled by configuration management procedures.

It is clearly understood what software is being tested, including new functionality, fixes, etc.

Subcontractor Management of Testing:

Acceptance test work products are primarily provided by the subcontractor.

Acceptance testing is performed on subcontracted software as part of their delivery.

Acceptance testing is done in accordance with a documented procedure.

Acceptance procedures and acceptance criteria for each software product are defined, reviewed, and approved by both the prime contractor and the subcontractor prior to test.

The results of the acceptance test are documented.

Measurement and Analysis

Data are collected on the software testing process:

- estimated and actual size (number of procedures), effort (hours), cost (tools, training, etc.).
- quality (defects detected before and after release, yield, types of defects, escapes, etc.).
- productivity data and re-use.

Verification of Implementation

PE/PM conduct periodic and event driven reviews of testing activities.

Managers (middle, and/or senior) conduct periodic and event driven reviews of testing activities.

SQA group reviews and/or audits testing activities and product.

Software Test Team reviews and audits their own work.

Appendix F: Software Test Team Process Specification

Validation Test Process

This process specification is broken down into the following tables:

- Table 1. Validation Testing Activities (Identifies the high-level activities that are detailed in the engineering steps)
- Table 2. Validation Testing Products (Identifies the products that are produced, transformed, or consumed by the test activities)
- Table 3. Validation Testing Roles (Identifies the roles to be performed in the test process)
- Tables 4-9. Validation Test Engineering Steps (Details the steps of validation testing across the stages of the lifecycle)

A high-level graphic is included below (Figure 1).

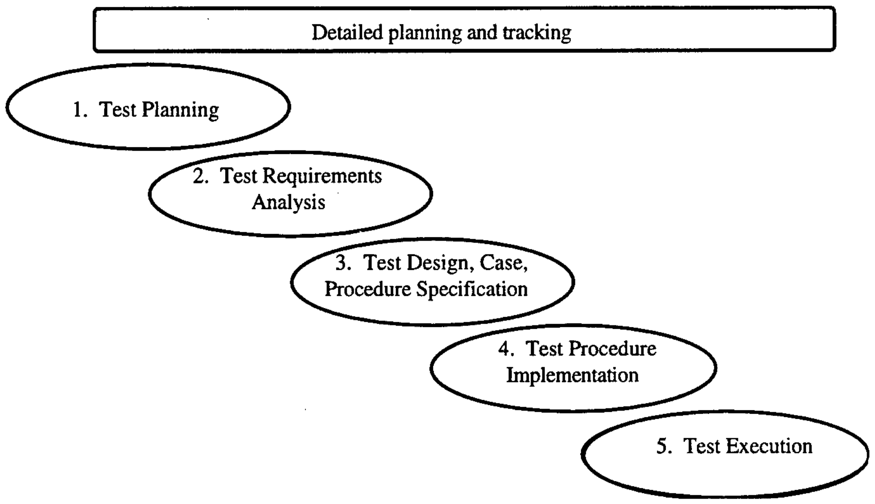

Detailed planning and tracking
1. Test Planning
2. Test Requirements Analysis
3. Test Design, Case, Procedure Specification
4. Test Procedure Implementation
5. Test Execution

Figure 1. Validation Test Process Diagram

Table 1. Validation Testing Activities

Id Code	Name and Description
TEST-PLAN	Test Planning Done during the Concept/Planning/Requirements stages of a project, this activity determines the philosophy of test and generates a high-level plan (SVTP) of the test activities.
TEST-REQT	Test Requirements Analysis Activity to review software requirements for testability, and map them to test method, test designs, test cases, and test procedures to ensure that there are validation test procedures to validate all testable software requirements.
TEST-SPEC	Test Design, Case, and Procedure Specification Defining the designs, cases, and procedures, including setup conditions, inputs, outputs, pass/fail criteria, etc.
TEST-PROC	Test Procedure Implementation Writing the step-by-step procedures detailed to customer needs to be executed.
TEST-EXEC	Test Execution Running the procedures to find defects in the software and verify proper operation.
TEST-PTO	Detailed Planning, Tracking and Oversight Activities to generate detailed workplans for all test activities, and track on a regular basis, including management oversight.

Table 2. Validation Testing Products

Id Code	Name and Description
TEST-SVTP	Software Validation Test Plan Planning document that outlines the test philosophy on a project, including unique terminology and defining levels of testing applicable to a customer requirements. This plan also establishes high-level plans for testing, including resources, schedule, tools and training required, and an initial pass at outlining the test design specifications.
CUST-SPEC	Customer Specification The documented customer requirements for the project.
TEST-RTVM	Requirements Test Verification Matrix This matrix maps the test designs, cases, and procedures to the project requirements to ensure that all testable requirements are validated.
TEST-TD	Test Design Specification Work product that breaks the testable requirements into several categories of tests, e.g., CTC, track warrants, train sheets, bulletins.
TEST-TC	Test Case Specifications Work product that breaks the test design specification down into functions that need to be tested, e.g., signal clear, dispatcher transfer, log on/off.
TEST-TP	Test Procedure Specification Work product that specifies all of the individual scenarios that a function must be exercised in order to verify proper operation and also to try to break.
TEST-PROC	Test Procedures Instructions that detail the step-by-steps to exercise all of the scenarios specified in the test procedure specification.
TEST-REPT	Test Reports Product that describes the results of test execution activities.
TEST-SPR	Problem Reports Documented event that results from an unexpected outcome of a test activity.
DEV-DOC	Development Documentation Documentation developed throughout the software process may be used for testing purposes, as a source of requirements and/or insight into what needs to be tested. These documents include the User Manual, Software Requirements Specification, Software Design Document, and any other informal documentation that is produced, e.g., meeting minutes, telecon minutes, etc.
TEST-CHKLST	Test Checklists
TEST-LL	Test Lessons Learned
TEST-MET	Test Activity Metrics Data and information that quantifies the testing activities and their results, e.g., hours per activity, start and end dates, defects.

Table 3. Validation Testing Roles

Id Code	Name and Description
TEST-LEAD	Test Lead Person responsible for overseeing test activities across projects, including preparing and reporting detailed planning and tracking metrics.
TEST-ENG	Test Engineers Those individuals responsible for testing software products.
RQTMGR	Requirements Manager or RTM Administrator Point of contact for the requirements activities; maintains the RTM.
MGT	Management Levels of management above the PROJMGT level that participate in the sponsorship and review of the requirements activities; e.g., Directors and above, Managers of Projects, Engineering Managers.
PROJMGT	Project Management Project Managers, Project Engineers, Supervisors with one (typically) project or functional area reporting to them. These individuals oversee the requirements management activities.
DEV-ENG	Engineers Those practitioners assigned to design and implement the systems, including hardware and software, on a project.
QA	Quality Assurance Those individuals assigned to ensure process adherence.
CM	Configuration Management Those individuals assigned to ensure product assurance.
CUST	Customer The contracting entity that produced the original need for the project (either new development or maintenance).
QUAD	Quality Access Database Automated tool that is used to enter, store, and retrieve information regarding SPRs (change requests and problem reports).
RTM	Requirements Traceability Matrix A relational database or spreadsheet used to store the project requirements and the allocations/traceability to other software development and software testing work products.
SYSENG	System Engineer Responsible for setting up lab environments, including test environment and supports systems/performance testing.
STT	Software Test Team Functional group responsible for validation test activities on projects, including cross project reviews of test work products.

Table 4. Test Planning

Id Code	Description
TEST-PLAN	Test Planning (during Concept/Planning/Requirements stage).
TEST-PLAN- Entry Criteria	From: Project startup Inputs: Project requirements specification (System Spec, RFP, contract, etc.) that details the functional requirements and the project schedule.
TEST-PLAN-1	PROJMGT and TEST-LEAD gathers and reviews all testing requirements related to the test process, including: • Scheduled dates of phases and stages • Contract Deliverable Line Items (CDRL's) related to testing and/or used as inputs to the testing activities • Dates specified in the project schedule for testing activities and/or deliverables • Levels of test specified by the customer, e.g., FAT, SAT, Trial Run, Final Commissioning • Initial estimate or forecast of the hours allocated to the testing activities • Quality criteria specified by the customer in the CUST-SPEC
TEST-PLAN-2	TEST-LEAD develops a test strategy that outlines all of the phases, stages, types, and levels of testing that will occur on the project.
TEST-PLAN-3	TEST-LEAD completes the sections (2.1-2.3) in the SVTP that documents the test strategy, including a flow diagram that illustrates the levels of testing from Unit/Integration, through System/Acceptance testing, through the various levels of testing as defined by the CUST-SPEC (e.g., FAT, SAT, Commissioning, etc.).
TEST-PLAN-4	TEST-LEAD identifies the testing work products that will be produced, by name and delivery date.
TEST-PLAN-5	TEST-LEAD completes the SVTP section on work products (2.4).
TEST-PLAN-6	PROJMGT and TEST-LEAD identify the source of functional requirements on the project, i.e., the CUST-SPEC. This may be the RFP, a Functional Spec, a System Spec, the Contract, or some other form of documentation. Typically, the test work products will trace back to the same requirements that are in the RTM, so whatever document was used to create the RTM is the document that testing will trace to.
TEST-PLAN-7	TEST-LEAD reviews the CUST-SPEC functional requirements in order to determine a logical set of Test Designs (TEST-TDs). *Note: At this point it is useful to determine a re-use strategy. If the project being planned is based off of and/or similar to an existing project, the TEST-LEAD should review the existing test work products in order to determine what might be re-used.*
TEST-PLAN-8	TEST-LEAD documents the Test Design Specification outline in the Features to Be Tested (and features not to be tested) section (2.5) of the SVTP.
TEST-PLAN-9	TEST-LEAD identifies all testing activities to be performed on the project, including preparation for testing, test execution, and post-test activities.
TEST-PLAN-10	TEST-LEAD documents test activities in the SVTP (section 2.6).
TEST-PLAN-11	TEST-LEAD maps all testing activities onto the project schedule and determines start and end dates as well as risks and contingencies.

TEST-PLAN-12	TEST-LEAD completes the schedule section of the SVTP (2.7). Risks and contingencies may be listed separately in section 2.12.
TEST-PLAN-13	TEST-LEAD and the PROJMGT define the staffing required for testing and support of testing, based on available resources.
TEST-PLAN-14	TEST-LEAD completes the Staffing section (2.8) of the SVTP.
TEST-PLAN-15	TEST-LEAD documents the test deliverables in the SVTP (section 2.9).
TEST-PLAN-16	TEST-LEAD and SYSENG define the environment for testing, including hardware that will be available and software tools and resources necessary.
TEST-PLAN-17	TEST-LEAD documents testing environment(s) in section 2.10 of the SVTP.
TEST-PLAN-18	TEST-LEAD, CM, and SYSENG define test control procedures as part of the configuration management of the project.
TEST-PLAN-19	TEST-LEAD completes the test control procedures section of the SVTP (2.11).
TEST-PLAN-20	TEST-LEAD completes the SVTP, including the risks and contingencies (2.11), suspension criteria and resumption requirements (2.13), sample forms, acronym list, according to the SVTP template.
TEST-PLAN-VAL	FTR of the SVTP with the following participants: • TEST-LEAD (author) • STT (moderator, also other members of the team may review the document) • RQTMGR (for consistency with the use of the RTM and test work products) • PROJMGT (to ensure consistency with project strategy, and that all schedulable items are accounted for) • ENG (to ensure that the functionality is adequately represented) • SYSENG (to ensure that the environmental needs are known and documented) • QA (to ensure the process is being followed) • CM (to identify the configurable items and review test control procedures)
TEST-PLAN-METRIC	SIZE: None EFFORT: Actual versus planned number of hours to execute the TEST-PLAN activities as described above, including re-work. SCHEDULE: Actual versus planned start and end dates for the TEST-PLAN activities; actual versus planned release date of the SVTP, including re-releases. QUALITY: Number of issues/defects in the SVTP via FTR's, tracked to closure.
TEST-PLAN-Exit Criteria	To: TEST-REQT activity Outputs: SVTP approved (FTR closed) - Customer approval is necessary for completion of this activity, but once the FTR is closed out, TEST-REQT may proceed. However, there is a risk in proceeding into other test activities without customer approval.

Table 5. Test Requirements Analysis

Id Code	Description
TEST-REQT	Test Requirements Analysis (during Requirements stage).
TEST-REQT- Entry Criteria	From: TEST-PLAN activity Inputs: RTM, SVTP (for initial TEST-TD outline)
TEST-REQT-1	TEST-LEAD and TEST-ENG review each requirement in the RTM and allocate each to a particular test method.
TEST-REQT-2	TEST-LEAD and TEST-ENG review all testable requirements and allocate them to a particular TEST-TD.
TEST-REQT-3	Any TEST-TDs that were not identified in the SVTP need to be included in the RTM. TEST updates the RTM to include the identified TEST-TDs.
TEST-REQT-4	TEST-ENG produces a report of the requirements allocation to test method for review (initial RTVM). *Note: This step may be combined with TEST-REQT-5.*
TEST-REQT-5	TEST-ENG produces a report of the testable requirements allocation to the TEST-TDs (complete TEST-RTVM).
TEST-REQT-6	As problems or issues with the requirements come up, TEST-LEAD and/or TEST-ENG document them and discuss them with the PROJMGT. If necessary, SPRs are generated based on defective requirements.
TEST-REQT-VAL	FTR of the TEST-RTVM with the following participants: • TEST-LEAD • TEST-ENG (author) • STT (moderator, also other members of the team may review the document) • RQTMGR (for consistency with the use of the RTM and test work products) • PROJMGT (to ensure consistency with project strategy, and that all schedulable items are accounted for) • DEV-ENG (to ensure that the functionality is adequately represented) • SYSENG (to ensure that the environmental needs are known and documented) • QA (to ensure the process is being followed) • CM (to identify the configurable items and review test control procedures) • CUST (to ensure they are aware of testable/untestable requirements)
TEST-REQT-METRIC	SIZE: # of requirements: total, allocated to each TEST-TD, allocated to each test method. EFFORT: Actual versus planned number of hours to execute the TEST-REQT activities as described above, including re-work. SCHEDULE: Actual versus planned start and end dates for the TEST-REQT activities; actual versus planned release date of the TEST-RTVM, including re-releases. QUALITY: Number of issues/defects in the TEST-RTVM via FTR's (or informal walkthroughs), tracked to closure.
TEST-REQT-Exit Criteria	To: TEST-SPEC activity Outputs: Approved TEST-RTVM

Table 6. Test Design, Case, and Procedure Specification

Id Code	Description
TEST-SPEC	Test Design, Case, and Procedure Specification (during Preliminary/Detailed Design stage)
TEST-SPEC- Entry Criteria	From: TEST-REQT activity Inputs: TEST-RTVM, SVTP (includes TD, TC, and TP formats)
TEST-SPEC-1	TEST-ENG reviews CUST-SPEC, RTM, and DEV-DOC to ensure that the outline of TEST-TDs is appropriate. Consider if traceability will be difficult. Select the most comprehensive document that will be used and maintained for software development (probably the SRS and/or UM). Test cases and procedures should follow this outline, with clear traceability to the RTM requirements.
TEST-SPEC-2	TEST-ENG produces TEST-TDs according to the template in the TEST-SVTP.
TEST-SPEC-3	TEST-ENG reviews the requirements allocated to each TEST-TD from the TEST-RTVM and outlines a logical set of test cases.
TEST-SPEC-4	TEST-ENG produces the TEST-TCs according to the template in the TEST-SVTP.
TEST-SPEC-5	TEST-ENG updates the TEST-RTVM with test case allocation and traceability information.
TEST-SPEC-6	TEST-ENG reviews the requirements allocated to each TEST-TC from the updated TEST-RTVM and outlines a logical set of test procedures.
TEST-SPEC-7	TEST-ENG produces the TEST-TPs according to the template in the TEST-SVTP.
TEST-SPEC-8	TEST-ENG updates the TEST-RTVM with test procedure allocation and traceability information.
TEST-SPEC-9	TEST-ENG reviews the requirements allocated to each TEST-TP from the updated TEST-RTVM and also gathers any additional DEV-DOC related to each procedure. (This step will continue throughout the remainder of the project, in addition to the next step).
TEST-SPEC-10	TEST-ENG specifies all relevant test scenarios to exercise the software based on the updated TEST-RTVM requirements allocated to each procedure and any other DEV-DOC that is appropriate. Throughout the project, as new information is gathered about a particular function, TEST-ENG collects this information in the form of test requirements in the TEST-RTVM. (These last two steps are sketchy, somehow we need to keep a flexible test procedure specification that we can continuously add new things to test or look out for)
TEST-SPEC-11	TEST-ENG prepares all TEST-TDs, TCs, and TPs, and an updated TEST-RTVM for distribution and review.
TEST-SPEC-VAL	FTR of the TEST-TDs, TCs, TPs, and TEST-RTVM with the following participants: TEST-LEAD TEST-ENG (author) STT (moderator, also other members of the team may review the document) RQTMGR (for consistency with the use of the RTM and test work products) PROJMGT (to ensure consistency with project strategy, and that all schedulable items are accounted for) DEV-ENG (to ensure that the functionality is adequately represented) SYSENG (to ensure that the environmental needs are known and documented) QA (to ensure the process is being followed) CM (to identify the configurable items and review test control procedures) CUST (to ensure they are aware of testable/untestable requirements)

TEST-SPEC-METRIC	SIZE: # of requirements: total, allocated to each TEST-TD, TC, TP, allocated to each test method. EFFORT: Actual versus planned number of hours to execute the TEST-SPEC activities as described above, including re-work. SCHEDULE: Actual versus planned start and end dates for the TEST-SPEC activities; actual versus planned release date of the TEST-TDs, TCs, TPs, including re-releases. QUALITY: Number of issues/defects in the TEST-TDs, TCs, TPs via FTR's (or informal walkthroughs), tracked to closure.
TEST-SPEC-Exit Criteria	To: TEST-PROC activity Outputs: Approved TEST-TDs, TCs, TPs, updated TEST-RTVM

Table 7. Test Procedure Implementation

Id Code	Description
TEST-PROC	Test Procedure Implementation (during Implementation stage)
TEST-PROC-Entry Criteria	From: TEST-SPEC activity Inputs: Approved TEST-TDs, TCs, TPs, updated TEST-RTVM
TEST-PROC-1	TEST-ENG review the TEST-RTVM and TEST-TDs, TCs, and TPs to prepare detailed workplans for preparing, reviewing, updating, and releasing the test procedures.
TEST-PROC-2	TEST-LEAD rolls up detailed workplans from TEST-ENG, to be tracked as part of TEST-PTO.
TEST-PROC-3	TEST-ENG writes detailed step-by-step procedures according to the depth that the customer requires. The procedures should clearly show how each scenario from the TEST-TP specs are being covered.
TEST-PROC-4	TEST-ENG updates detailed workplans based on progress of TEST-PROC activities.
TEST-PROC-5	TEST-LEAD updates high-level workplans based information from TEST-ENG.
TEST-PROC-6	TEST-LEAD prepares weekly reports of TEST-PROC progress and distributes to PROJMGT and STT.
TEST-PROC-7	TEST-ENG conducts FTRs of TEST-PROCs as described under TEST-PROC-VAL.
TEST-PROC-8	TEST-ENG updates TEST-PROCs based on FTRs.
TEST-PROC-9	TEST-ENG updates the TEST-RTVM based on TEST-PROC activities.
TEST-PROC-10	TEST-ENG prints out final executable procedures marked as "Preliminary" and "Subject to Change" for release to customer.
TEST-PROC-11	TEST-LEAD sends TEST-PROCs and TEST-RTVM to customer.
TEST-PROC-VAL	FTR of the TEST-PROCs, and TEST-RTVM with the following participants: • TEST-LEAD • TEST-ENG (author) • STT (moderator, also other members of the team may review the document) • RQTMGR (for consistency with the use of the RTM and test work products) • PROJMGT (to ensure consistency with project strategy, and that all schedulable items are accounted for) • DEV-ENG (to ensure that the functionality is adequately represented) • SYSENG (to ensure that the environmental needs are known and documented) • QA (to ensure the process is being followed) • CM (to identify the configurable items and review test control procedures) • CUST (to ensure they are aware of testable/untestable requirements) - *OPTIONAL*

TEST-PROC-METRIC	SIZE: # of requirements: total, allocated to each TEST-TD, TC, TP, allocated to each test method; actual versus planned # of procedures identified, prepared, reviewed. EFFORT: Actual versus planned number of hours to execute the TEST-PROC activities as described above, including re-work. SCHEDULE: Actual versus planned start and end dates for the TEST-PROC activities; actual versus planned release date of the TEST-PROCs, including re-releases. QUALITY: Number of issues/defects in the TEST-PROCs via FTR's (or informal walkthroughs), tracked to closure.
TEST-PROC-Exit Criteria	To: TEST-EXEC activity Outputs: Approved TEST-PROCs, updated TEST-RTVM

Table 8. Test Execution

Id Code	Description
TEST-EXEC	Test Execution (during Integration/Validation test through Operation and Maintenance stages).
TEST-EXEC- Entry Criteria	From: TEST-PROC activity Inputs: TEST-PROCs Test events, including: • Dry-runs • Customer demonstrations • Factory acceptance testing, Site Acceptance Testing • Regression testing • Verification of fixes
TEST-EXEC-1	Before a test, TEST-LEAD and TEST-ENG prepare a checklist of procedures to be executed for an upcoming test event.
TEST-EXEC-2	TEST-ENG ensures that all procedures are reviewed and up-to-date, ready for test.
TEST-EXEC-3	TEST-ENG works with SYSENG and DEV-ENG to coordinate the baseline and lab setup for the test event. Must ensure all know what is supposed to be in baseline.
TEST-EXEC-4	TEST-ENG and any support from the STT and/or DEV-ENG execute procedures at least two weeks prior to a test event to find problems in the software and/or test documentation.
TEST-EXEC-5	TEST-ENG updates procedures a final time as a result of test execution dry-runs.
TEST-EXEC-6	TEST-ENG sits with CUST and/or QA to perform the test event.
TEST-EXEC-7	SPRs are generated from test events, including dry-runs.
TEST-EXEC-8	TEST-ENG prepares TEST-REPTs in the format identified in the TEST-SVTP.
TEST-EXEC-VAL	Dry-runs with the DEV-ENGs, STT, QA. Test Readiness Reviews with PROJMGT. FTR of the TEST-REPTs, and TEST-RTVM with the following participants: • TEST-LEAD • TEST-ENG (author) • STT (moderator, also other members of the team may review the document) • RQTMGR (for consistency with the use of the RTM and test work products) • PROJMGT (to ensure consistency with project strategy, and that all schedulable items are accounted for) • DEV-ENG (to ensure that the functionality is adequately represented) • SYSENG (to ensure that the environmental needs are known and documented) • QA (to ensure the process is being followed) • CM (to identify the configurable items and review test control procedures) • CUST (to ensure they are aware of testable/untestable requirements)

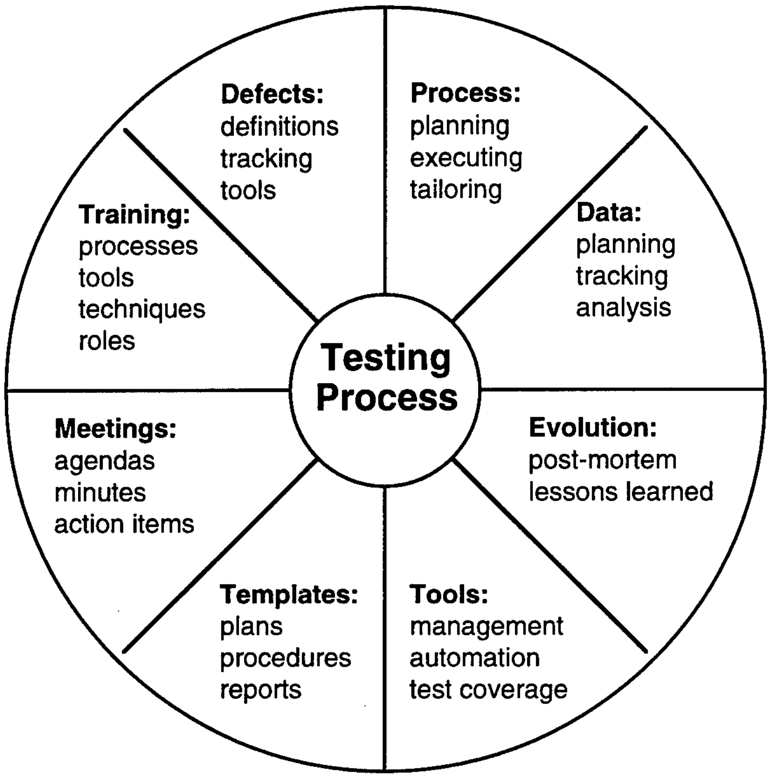
TEST-EXEC-METRIC	SIZE: # of procedures executed, passed; # of SPRs. EFFORT: Actual versus planned number of hours to execute the TEST-EXEC activities as described above, including re-work. SCHEDULE: Actual versus planned start and end dates for TEST-EXEC activities; actual versus planned release date of the TEST-REPTs, including re-releases. QUALITY: Number of issues/defects in the TEST-REPTs via FTR's (or informal walkthroughs), tracked to closure; # of SPRs sliced and diced many different ways, tracked to closure.
TEST-EXEC-Exit Criteria	To: END Outputs: Approved TEST-PROCs, updated TEST-RTVM

Table 9. Detailed Planning, Tracking, and Oversight

Id Code	Description
TEST-PTO	Detailed Planning, Tracking, and Oversight (during all stages)
TEST-PTO- Entry Criteria	From: All test activities Inputs: Planned data from SVTP; Actual data from various test activities.
TEST-PTO-1	TEST-LEAD prepare high level workplans that defines the scope of work and level of effort on a project, including planned hours and dates.
TEST-PTO-2	TEST-ENG prepares detailed workplans based of the high level workplans that details down to the procedure level what activities take place.
TEST-PTO-3	TEST tracks actual progress against the detailed workplans weekly.
TEST-PTO-4	TEST-LEAD rolls up detailed workplans and tracks high level workplans weekly.
TEST-PTO-5	During periods of testing, TEST generates problem reports and provides summary reports.
TEST-PTO-6	TEST-LEAD tracks test problems to closure.
TEST-PTO-7	STT reviews workplans bi-weekly.
TEST-PTO-8	Postmortem
TEST-PTO-VAL	PROJMGT reviews data weekly. QA reviews data periodically to ensure it is being tracked. MGT reviews reports quarterly.
TEST-PTO-METRIC	SIZE: Based on the number of requirements and historical data, TEST-LEAD estimates how many test procedures need to be developed for this project, and some idea of how much re-use can be realized. EFFORT: Actual versus estimated hours to plan, track, and oversee testing activities. SCHEDULE: Release dates of reports QUALITY: None
TEST-PTO-Exit Criteria	To: All activities Outputs: Test Metrics reports

Appendix G: "Whole-Product Wheel"

Considering the testing process being defined as the core technology, the group brainstormed a list of all of the components that will be needed to support adoption of the process within the organization. The picture below illustrates some of the important considerations that need to be addressed in support of the testing process. Although not all of these areas could be addressed immediately, it does provide the team with a better understanding of how the test process fits within the organization's software processes.


Testing Process
Defects: definitions tracking tools
Process: planning executing tailoring
Data: planning tracking analysis
Evolution: post-mortem lessons learned
Tools: management automation test coverage
Templates: plans procedures reports
Meetings: agendas minutes action items
Training: processes tools techniques roles

Appendix H: Process Baseline Matrix

Reqt Section	Description	A	B	C	D	E	F	G	H	I	Yes	No	Inc	n/a	blank		Org Lvl	Respon-sibility	Risk
Commitment 1	A dedicated team is assigned to software testing and is staffed with qualified test engineers.	yes	yes	no	yes	inc	no	inc	inc	inc	3	2	4	0	2	5	no	Mgt.	Perception from STT is that it is a working group; but others feel there is an independent, dedicated STT.
Commitment 2	Managers establish quality and testing criteria with the test team and use testing data to make decisions.	no	inc	inc	yes	inc	no	no	no	inc	1	4	4	0	2	3	no	Mgt.	During crunch time we tend to overlook testing and/or rate of SPR generation.
Commitment 3	Quality and testing criteria are developed and reviewed with the customer and the end-users.	inc	inc	inc	no	yes	no	yes	no	yes	3	3	3	0	2	4.5	no	Mgt/STT	Same as Commitment 2
Commitment 4	Test readiness criteria are established and used on projects to determine when the lifecycle can proceed (e.g., when to proceed to the next phase, or when to ship software).	inc	no	inc	yes	no	no	no	no	no	1	6	2	0	2	2	no	Mgt/STT	Need entry and exit criteria to define boundaries between unit / integration / validation test.
Commitment 5	Resources for testing the software are assigned early enough to provide for adequate test preparation.	yes	inc	inc	yes	inc	no	inc	yes	inc	3	1	5	0	2	5.5	yes	Mgt.	

Commitment 6	Resources are also available from the developers to work with the testers to ensure a thorough test suite.	no	inc	inc	yes	inc	inc	inc	yes	no	2	2	5	0	2	4.5	yes	Mgt.	
Ability 1	On each project, staffing and funding are allocated to software testing in accordance with project needs to provide training, tools, and perform the testing activities.	inc	no	no	yes	inc	no	inc	yes	inc	2	3	4	0	2	4	no	Mgt.	Need to base test resources on size of project, and retain test resources during crunch times, not re-assign to code.
Ability 2	Training is available for software testing techniques including verification methods, test planning, use of tools.	inc	inc	no	n/a	inc	no	no	yes	inc	1	3	4	1	2	3	no	STT	Will define and detail needs as we develop testing process.
Ability 3	Tools to support testing are available, e.g., test management, test generators, test drivers, symbolic debuggers, test coverage analyzers, capture playback.	inc	inc	no	no	inc	no	inc	yes	no	1	4	4	0	2	3	inc	STT	Not a current priority.
Ability 4	Procedures exist to be adapted onto each project so that the organizational test process is consistently implemented.	no	inc	inc	yes	no	inc	inc	no	no	1	4	4	0	2	3	no	STT	STT is working on this.

Act/General 1	System testing is planned and performed to ensure that the software satisfies the software requirements and quality criteria.	yes	inc	inc	yes	no	no	yes	inc	no	3	3	3	0	2	4.5	yes	STT	
Act/General 2	Acceptance testing is planned and performed to demonstrate to the customer and end-users that the software satisfies the allocated requirements when required.	no	yes	inc	yes	inc	no	yes	inc	inc	3	2	4	0	2	5	yes	STT	
Act/General 3	System and Acceptance testing are documented in a project test plan which is reviewed with and approved by the customer, end-users, and managers.	no	yes	inc	no	inc	no	yes	yes	inc	3	3	3	0	2	4.5	yes	STT/Mgt	
Act/General 4	The test plan covers the overall testing and verify approach, resp of the dev organization, subcont, customer, and end-users, test eq, facilities, and support reqts, acc criteria.	inc	inc	no	yes	yes	no	yes	inc	yes	4	2	3	0	2	5.5	yes	STT	
Act/General 5	Test cases and procedures are planned and prepared independent of but with support from the software developers.	yes	yes	no	inc	yes	no	yes	yes	inc	5	2	2	0	2	6	yes	STT	

Act/General 6	Test work products undergo peer review, including within the software test team.	inc	inc	no	yes	inc	no	inc	inc	no	1	3	5	0	2	3.5	no	STT	Need a dedicated test team
Act/General 7	Test cases and procedures are documented and reviewed with and approved by the appropriate individuals (e.g., customer, end-users, managers) before testing begins.	yes	no	inc	yes	inc	no	inc	inc	no	2	3	4	0	2	4	inc	STT	
Act/General 8	Testing of the software is performed against baseline software and the baseline documentation of the allocated requirements and the software requirements.	yes	yes	inc	yes	inc	no	inc	inc	no	3	2	4	0	2	5	yes	STT	
Act/General 9	Problems identified during testing are documented and tracked to closure.	yes	inc	inc	yes	yes	inc	yes	yes	no	5	1	3	0	2	6.5	yes	STT	
Act/General 10	Test work products are re-used across projects.	no	no	no	yes	inc	no	inc	no	no	1	6	2	0	2	2	no	STT	Have not historically been designed for re-use; STT is working on this.

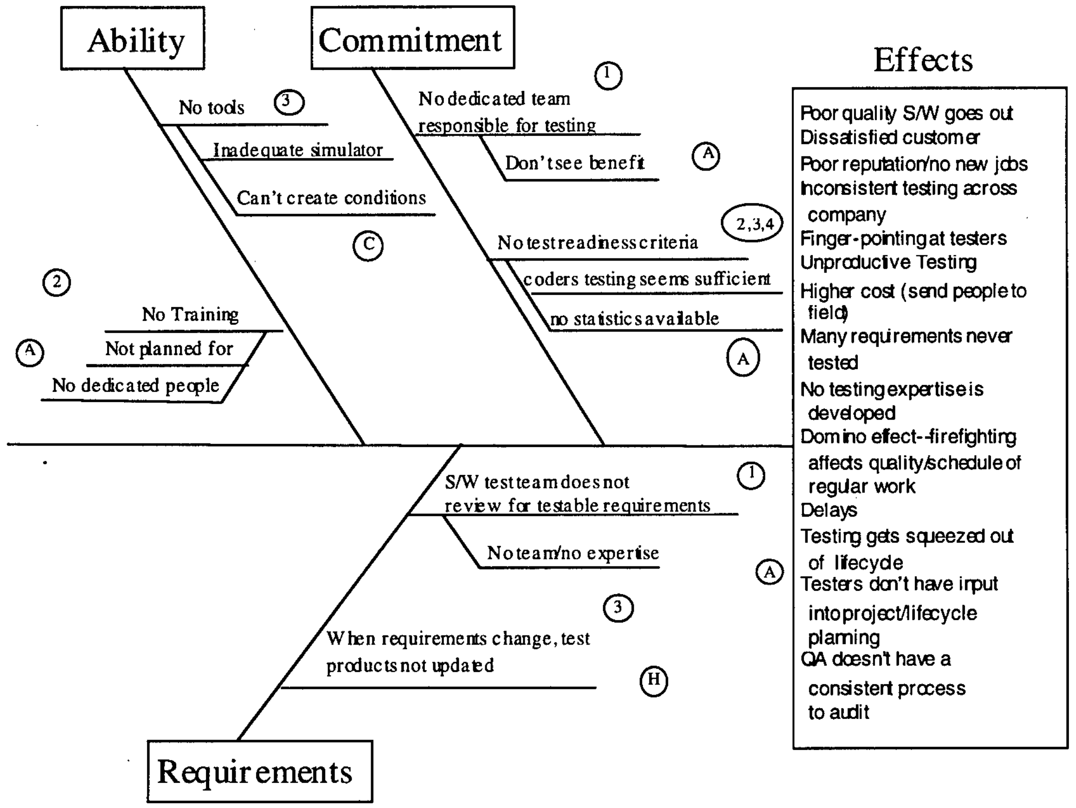
Act/General 11	Adequate regression testing is performed when software and/or environment changes to ensure that changes have not caused unintended effects on the baseline.	no	inc	no	inc	inc	no	inc	inc	no	0	4	5	0	2	2.5	no	STT/Mgt	No clear regression process; not often time to perform regression testing (blasts go out too quickly).
Act/Reqt Mgt 1	Requirements allocated to software are reviewed by the STT to determine whether they are testable.	no	n/a	no	inc	no	no	yes	inc	inc	1	4	3	1	2	2.5	no	Mgt/STT	Being built into testing process. RTM's have historically been created and not kept up or fully utilized.
Act/Reqt Mgt 2	Consistency is maintained across software work products (including testing) via traceability matrices.	inc	no	n/a	yes	yes	n/a	inc	yes	inc	3	1	3	2	2	4.5	yes	STT	
Act/Reqt Mgt 3	When requirements change, test work products are updated accordingly.	inc	inc	no	yes	inc	no	inc	inc	inc	1	2	6	0	2	4	no	STT	How do we get updates?
Config Mgt 1	Test work products are identified as configuration items by the project.	inc	no	no	yes	inc	no	yes	yes	no	3	4	2	0	2	4	yes		
Config Mgt 2	Test work products are maintained in accordance with documented configuration management procedures/plans.	inc	no	no	yes	no	no	yes	yes	no	3	5	1	0	2	3.5	no	STT	Need a config mgt function to work with.

Config Mgt 3	The test environment, including software, is controlled by configuration management procedures.	inc	yes	no	yes	yes	no	no	no	no	3	5	1	0	2	3.5	inc	STT/ CM/Mgt	Need a config mgt function to work with.
Config Mgt 4	It is clearly understood what software is being tested, including new functionality, software fixes, etc.	no	inc	yes	yes	yes	inc	no	no	no	3	4	2	0	2	4	no	STT/ CM/Mgt	Need a config mgt function to work with.
Act/Subcont 1	Acceptance test work products are primarily provided by the subcontractor.	no		no	n/a	n/a	no	n/a	n/a	n/a	0	3	0	5	3	0	no	Mgt.	Not initially under STT charter.
Act/Subcont 2	Acceptance testing is performed on subcontracted software as part of their delivery.	no		inc	n/a	n/a	no		n/a	n/a	0	2	1	4	4	0.5	no	Mgt.	Not initially under STT charter.
Act/Subcont 3	Acceptance testing is done in accordance with a documented procedure.	inc		no	n/a	n/a	no	inc	n/a	n/a	0	2	2	4	3	1	no	Mgt.	Not initially under STT charter.
Act/Subcont 4	Acceptance procedures and acceptance criteria for each software product are defined, reviewed, and approved by both the prime contractor and the subcontractor prior to test.	no		n/a	n/a	n/a	n/a	n/a	n/a	n/a	0	1	0	7	3	0	no	Mgt.	Not initially under STT charter.
Act/Subcont 5	The results of the acceptance test are documented.	yes		no	n/a	n/a	no	yes	n/a	n/a	2	2	0	4	3	2	no	Mgt.	Not initially under STT charter.

Measure 1	Data are collected on estimated and actual size (number of procedures), effort (hours), cost (tools, training, etc.).	inc	inc	no	inc	inc	inc	yes	yes	n/a	2	1	5	1	2	4.5	yes	STT	
Measure 2	Data are collected on quality (defects detected before and after release, yield, types of defects, escapes, etc.).	no	inc	no	yes	inc	inc	yes	yes	no	3	3	3	0	2	4.5	yes	STT	
Measure 3	Data are collected on productivity data and re-use.	no	inc	no	inc	no	no	yes	inc	no	1	5	3	0	2	2.5	no	STT	Need to define measures.
Verification 1	PE/PM conduct periodic and event driven reviews of testing activities.	no	no	no	yes	no	no	no	no	no	1	8	0	0	2	1	no	Mgt.	Testing needs to be reviewed as part of standard project reviews
Verification 2	Managers (middle, and/or senior) conduct periodic and event driven reviews of testing activities.	no	no	no	no	no	no	no	no	no	0	9	0	0	2	0	no	Mgt.	Testing needs to be reviewed as part of standard project reviews
Verification 3	SQA group reviews and/or audits testing activities and product.	yes	yes	inc	yes	yes	no	yes	yes	no	6	2	1	0	2	6.5	yes	QA	
Verification 4	Software Test Team reviews and audits their own work.	no	n/a	n/a	inc	no	n/a	yes	yes	no	2	3	1	3	2	2.5	no	STT	Haven't really had a fully staffed STT.

Appendix I: Fishbone with Notes and Actions

Below is a summary of the fishbone diagram and notes and actions to follow up. This represents about 1/8 of the total fishbone generated in a meeting.


Ability
Commitment
Effects
No tools
3
No dedicated team responsible for testing
1
Inadequate simulator
Don't see benefit
A
Can't create conditions
2,3,4
C
No test readiness criteria
2
coders testing seems sufficient
No Training
no statistics available
A
Not planned for
A
No dedicated people
Poor quality S/W goes out
Dissatisfied customer
Poor reputation/no new jobs
Inconsistent testing across company
Finger-pointing at testers
Unproductive Testing
Higher cost (send people to field)
Many requirements never tested
No testing expertise is developed
Domino effect--firefighting affects quality/schedule of regular work
Delays
Testing gets squeezed out of lifecycle
Testers don't have input into project/lifecycle planning
QA doesn't have a consistent process to audit
S/W test team does not review for testable requirements
1
No team/no expertise
A
3
When requirements change, test products not updated
H
Requirements

Notes and Actions

(A) Need a project that sticks to testing and collect statistics. Collect

- costs to and not to test
- problems found, not found

(C) Combined project solution

- figure out once and resuse
- standard library of test procedures

(H) Need to establish test role in change control procedures/process

Appendix J: Risk Statement Matrix

ID	Risk	Level	Impact	Prob-ability	Classifi-cation	Prior-ity	Assgn. To
	The condition followed by the consequence.	*Quality* *Cost* *Schedule*	*High* *Medium* *Low* *None*	*High* *Medium* *Low* *None*	*Policy* *Procedure* *Process* *Training* *Tools* *Culture* *Data*	*Critical* *High* *Medium* *Low* *None*	*Mngmt* *STT* *CM* *QA* *DV*
A-1	If management abandons testing on project, then we won't know if it would have worked or not (benefit).	Quality Cost Schedule	High	High	Culture	Medium	Mngmt
A-2	If we don't collect appropriate data then we won't understand the benefit.	Quality Cost Schedule	High	Low	Data	Medium	STT
A-3	If we don't dedicate individuals to team then expertise will not develop.	Quality Cost Schedule	High	Medium	Training	Medium	Mngmt
A-4	If we don't define consistent process and transition it, then projects will waste resources.	Quality Cost Schedule	High	Low	Procedure	Medium	STT
A-5	If we define processes but projects don't use then no benefit will be derived from effort.	Quality Cost Schedule	High	Low	Culture	Medium	Mngmt
B-1	If management doesn't allocate time to testing, more testing will have to be done in the field (at a higher cost).	Quality Cost	High	Medium	Policy	High	Mngmt
B-2	If management allocates time but not resources then testing is incomplete.	Quality Cost	High	Medium	Policy	High	Mngmt
B-3	If the equipment allocated for testing is inadequate, testing is incomplete.	Quality Cost	High	Medium	Policy	High	Mngmt
B-4	If neither time nor resources are scheduled, testing is incomplete.	Quality Cost	High	Medium	Policy	High	Mngmt
B-5	If tests fail and we ship anyway, then we lose credibility with the customer and in the industry.	Quality	High	High	Policy	Critical	Mngmt

B-6	If management doesn't know about defects, then they cannot make informed decisions.	Quality	High	Low	Procedure	High	STT
B-7	If exit criteria is not defined, we won't know when testing is complete.	Quality Cost Schedule	Med	Med	Policy	High	Mngmt/ STT
C-1	If we are not able to simulate conditions in field, then • software is not completely developed • performance issues are not addressed • total functionality is not covered • interface/stress not tested • we're not able to duplicate problems in the field • we're not able to do computability testing of old and new code	Cost Quality	High	Med	Tools	Med	STT/DV
D-1	If no procedures exist, then there is no consistency.	Quality	High	Low	Procedure	High	STT
E-1	If the process does not include reviews with management, then • allocation for scheduling is unknown • uninformed ship decisions are made.	Quality Schedule	Med	Low	Process	Med/ Low	STT/ Mngmt
F-1	If configuration management test procedures are not in place, time is wasted using an improper tool.	Cost	Low	Low	Tools	Low	STT
G-1	If software under test is not controlled, then time is wasted testing untested software.	Quality	High	High	Process	High	STT/CM

H-1	If the developers don't keep the testers informed of changes, then • testers will develop incorrect procedures • defects will be missed • real code will not be tested	Quality	Med	Med	Process	Med	Mngmnt

Appendix K: Pilot Plans

Pilot Plan Template with Instructions

Plan Component	Description
Pilot Objective:	<Define the objective of this pilot effort.>
Process/Phase:	<Identify the process area and phase being pilot tested.>
Project:	<Identify the project(s) that the pilot effort will take place.>
Sponsor:	<Identify names of sponsors, primarily the project PE's or PM's, that should review and approve this plan, as well as take part in the implementation/evaluation of the pilot effort.>
Description:	<Provide a 3-5 sentence description of the pilot effort.>
Milestones:	<Define the major milestones for the pilot effort, paying specific attention to project milestones that need to be met.>
Deliverables:	<Define the project deliverables that will result from this pilot as well as any artifacts from the pilot to be used for process improvement purposes, e.g., lessons learned.>
Resources:	<Identify the staffing and effort estimates needed for this pilot, including a comparison to existing project staffing and effort estimates, i.e., is the pilot effort above and beyond or is it in place of initial project commitments.>
Standards/Policy Waivers:	<Identify any deviations from standard procedures that the pilot effort requires. These need to be documented, reviewed and approved by QA and/or appropriate management.>
Training Necessary:	<Describe any training that is necessary for the improvement team, the pilot project, the sponsor, or anybody who needs to take part in either the implementation and/or evaluation of the pilot effort.>
Initial Process Description:	<Describe the process being pilot tested by referring to an attached process specification.>
Feedback Capture Method:	<Describe how feedback will be captured, e.g., in the form of SPR's, lessons learned or post-mortem meetings, etc.>
Proposed Solution:	<Describe the outcome of the pilot effort, and how it will be documented, e.g., red-lined initial process, draft SEH chapter, etc.>
Steps to be Taken:	<Describe sequence of steps to be implemented and tracked by referring to attached workplan.>
Measures:	<Describe size, effort, schedule, and quality measures to be estimated and tracked (plan versus actual.>
Evaluation Criteria:	<Identify how the pilot will be evaluated in terms of success or failure, and how this will be incorporated into further pilot efforts or standard procedure.>
Miscellaneous:	<List additional notes, constraints, criteria, etc. that relates to the pilot effort.

Test Planning Pilot Plan

Plan Component	Description	
Pilot Objective:	Assess the adequacy of the draft Test Planning process specification. Develop test plans for the Tri-Met project.	
Process/Phase:	Software Validation Test - *Test Planning*	
Project:	Tri-Met	
Sponsor:	Amy Peterson, Elaine Gunnel, Larry Harskowitch - Need not approve plan as it does not represent additional work on the project beyond what has already been negotiated as support.	
Description:	The Tri-Met project is a large, long-term transit job that involves a typical scenario of , multi-subcontractor, multi-releases of software, and multi-staged validation testing. This scenario, similar to MBTA, represents one end of the spectrum that our validation test process must satisfy, i.e., the most extensive test planning that we must accommodate. This is in contrast to a smaller job where we may not be integrating subs, nor have multi-releases of software.	
Milestones:	Solicit subcontractor information	4/1-4/15
	Subcontractor meeting	4/15-4/19
	Prepare draft Comprehensive Test Plan (CTP)	4/22-4/26
	Send out for final solicitation of subs info	4/29-5/3
	Prepare final CTP	5/6-5/10
	Discuss with Tom Szur (QA) re: SVTP part of CTP	5/13-5/14
	US&S Tri-Met project review of CTP and update	5/13-5/16
	Deliver CTP to Tri-Met	5/20
	STT FTR of CTP	5/24
	Update Test Planning Process Specification	5/31
Deliverables:	Comprehensive Test Plan Test Plan template Updated Planning process specification	
Resources:	Don McAndrews, prepare CTP and conduct FTR	100 hours
	Support at subcontractor meeting (team)	5 hrs X 10
	H/W, S/W, Subs (internal and external) provide information to plan	?
	QA review	4 hrs
	Project review (FTR)	16 hrs total
	STT review	2 hrs X 8
Standards/Policy Waivers:	This activity deviates from the format of the SVTP currently being used on other projects in the past. However, the content is the same. If appropriate, we will be developing a planning template for SVTP's that will be more like the Tri-Met CTP, the major difference being that only "planning" information is included. The "process" information that used to be in SVTP's will be put into the chapter on validation testing in the *Software Engineering Handbook*. This will be discussed with Tom Szurszewski (QA) to gain his acceptance.	
Training Necessary:	None to execute pilot. Training needs will be developed as a result of this pilot.	
Initial Process Description:	See Validation Test Process Specification.	

Feedback Capture Method:	Lessons learned will be discussed with Sponsors and in an STT meeting. Ultimately, the STT will conduct an FTR of the CTP and the final process description.
Proposed Solution:	Outputs of this pilot will be used as draft inputs to the *Software Engineering Handbook*.
Steps to be Taken:	See milestones
Measures:	Hours to develop test plans Hours to meet with H/W, S/W, subs Hours for reviews Hours expended by H/W, S/W, subs
Evaluation Criteria:	Was the CTP produced as planned? (obj) Was it approved? (obj) Was the process adequate? (subj)
Miscellaneous:	

Test Procedure Development Pilot Plan

Plan Component	Description	
Pilot Objective:	Assess the adequacy of the Test Procedure Implementation process. Develop test procedures Test Procedures for the Hamersly Iron project.	
Process/Phase:	Test Procedure Implementation	
Project:	Hamersly Iron	
Sponsor:	Wes McQuiston. Need not approve plan as it does not represent additional work on the project beyond what has already been negotiated as support.	
Description:	The Hamersly Iron is a modest railroad job located in Australia services mining trains.	
Milestones:	Develop Test Design Specs	3/6-3/9
	Develop Test Case Specs	3/12 -3/14
	Develop Test Procedure Specs	3/15-3/20
	Re-use strategy	3/7-3/20
	Develop Test Procedures	3/20-5/15
Deliverables:	Written and approved test procedures. Updated Test Procedure Implementation process.	
Resources:	Dave Majernik, write test procedures	250 hrs
	Support from developing engineers	5 hrs
	Project review	16 hrs
Standards/Policy Waivers:	None	
Training Necessary:	None to execute pilot. Training needs will be developed as a result of this pilot.	

Initial Process Description:	See Test Design, Case, and Procedure specifications.
Feedback Capture Method:	Lessons learned will be discussed with Sponsors and in an STT meeting. Ultimately the STT will conduct and DTR on the test implementation process.
Proposed Solution:	Outputs of this pilot will be used as draft inputs the the *Software Engineering Handbook.*
Steps to be Taken:	See milestones.
Measures:	Hours to develop test procedures. Hours for reviews.
Evaluation Criteria:	Were the test procedures produced as planned? Were test procedures approved? Was the process adequate?
Miscellaneous:	There is a constraint that some of the functionality for which tests are being developed have not been fully defined and/or developed.

Test Execution Pilot Plan

Plan Component	**Description**
Pilot Objective:	Assess the adequacy of the draft Test Execution process specification. Prepare and execute test plans for the MBTA project.
Process/Phase:	Test Execution (Table 8. of the Validation Test Process document).
Project:	The Massachusetts Bay Transit Authority (MBTA).
Sponsor:	Lori J. Karr - Project Manager - Need not approve pilot plan since it does not represent additional work on the project beyond what has already been scheduled.
Description:	This pilot effort will include updating checklists and test procedure preparation steps before test execution. The preparation will help formal test execution with the customer go smoother. The only SPRs created during formal test execution should be code or design related, not related to incorrect test procedures.

Milestones:	Updated Build 5 Preparation Report — 06/14/96 Completed Build 5 test plan checklists — 06/14/96 Build 5 Release: Integration for A. — 04/29/96 to 05/03/96 Integration for B. — 06/03/96 to 06/07/96 Integration for C. — 06/10/96 to 06/14/96 Update Test Execution Process Specification — 06/14/96
Deliverables:	1. Updated (red-lined) version of the Test Execution for the Validation Test Process for its use in the SEH. 2. Test Execution checklist for Before Execution and After Execution. 3. Testing Progress Reports for each part (A, B, C, etc.). 4. Software Problems Reports (SPRs) for each part (A, B, C, etc.). 5. Risk prioritized list of test procedures. (Critical, Major importance, Low, and 'nice-to-haves.'
Resources:	The staffing includes the following: 1. Project Test Engineer(s) 2. Software Test Team (STT) 3. Project Manager (PM) 4. Development Engineers involved in Release 5. 5. Configuration Manager (CM) 6. Customer (DCCo)
Standards/Policy Waivers:	None
Training Necessary:	None.
Initial Process Description:	See Validation Test Execution Specification.
Feedback Capture Method:	1. SPRs 2. Marked Test Procedures 3. Post-mortem meetings 4. Lessons Learned
Proposed Solution:	After executing this pilot, we will be able to use the red-lined version of the Test Execution section of the Validation Test Process to update the SEH. We will also have an updated version of the Checklist(s) used during the Test Exection phase.

Steps to be Taken:	See attached workplan.
Measures:	The following will be tracked during this pilot: 1. Test procedure reviews. 2. Test procedure updates. 3. Test procedure dry-runs. 4. Test procedure execution readiness. 5. Test procedure total preparation times. 6. Test procedure execution times. 7. Number of SPRs per test procedure.
Evaluation Criteria:	1. Build 5 testing completes successfully. 2. Customer (DCCo) is satisfied with the new test execution process. 3. Development engineers and Management feel the new test execution process is an improvement. 4. Build 5 high risk test procedures are executed thoroughly.
Miscellaneous:	1. This pilot requires the full support of the development engineers and management. 2. This pilot requires more than one test engineer.... this may have to include support from one or more members of the STT.

Appendix L: Annotated Outline for Test Process Training

Introduction

This outline identifies the training modules to be developed for software validation test process training. The process training described in this outline is for individuals who will be testing in support of projects, as members of the Software Test Team. Other training modules for managers, developers, customers, etc. can be developed from subsets of these materials.

This annotated outline contains the product plan and design information from the product planning meeting held 6/26, and an annotated outline that identifies materials to be developed and by whom.

Course Overview

This training is designed to be taught as a one-day workshop or two 1/2 day workshops. The first half day deals with the overall process and planning and tracking testing, while the second half day deals with more technical aspects of actually performing the testing and working in the project environments.

Each module is designed with the assumption that the Training Center or the Conference Center is used so that access to on-line demonstrations is possible. If these rooms are not available, the training needs to be tailored so that content can be taught in a classroom, then on-line demos can be conducted in the lab or at someone's desk.

Continued on next page

Other Requirements from Product Plan Meeting

Audience:

- Software professional with 3 to 5 yrs experience and some test experience
- No US&S domain expertise required
- PC literate/knows Microsoft Access
- +/OOP background

Roles:

Test Engineer, member of the STT

- integration/system/acceptance testing
- participating in design/code reviews
- maintaining RTM

Knowledge and Experience:

3 - 5 years software experience, vocabulary - new

Attitude & Motivation: N/A

Relevance - Each outline item was categorized according to the level of detail needed: Awareness, Understanding; Proficiency. See meeting minutes from 6/19 meeting for details.

Usage: Apply on a project.

Design Considerations

Development and Delivery Constraints:

- immediate to complete process and bring new employees up to speed
- target delivery date - August 15, 1996
- apply resources, STT members 2-4 hours per week

Media and Materials:

- slides
- computer demonstrations
- sample documents
- process documents
- video clips

Learning and Evaluation Techniques: use standard evaluation form

Module outline: see below

Continued on next page

Annotated Outline, 1st Half-Day

Objective: *For the students to understand the need for testing, the standard process for testing, and how to develop procedures on a project.*

Introduction

Course goals and objectives, review agenda.

Testing Overview

- Organizational units who Test and Relationships (Central Office, Wayside, Vehicle, other) - explain the various product areas, including hardware, and how testing activities do/don't inter-relate.
- Lifecycle Testing - describe the lifecycle at US&S and the various activities that occur on a project throughout the lifecycle related to testing (including design and code reviews).
- Defined Process - why we need it, overview of the phases and stages, procedures, tools, products and templates, agents (relationships with testers, developers, QA, managers, etc.), etc.
- High-level scheduling of testing - how testing fits in with project schedule.

Requirements

- What is a testable requirement?
- RTM Overview - how projects create them, what they use them for.
- Allocating Requirements - test designs, cases, procedure identification, RTVM (requirements Test Verification Matrix).

Writing Procedures

- Creating the procedure - format, requirements, checklist.
- Reviewing Procedures - getting feedback from developers, STT reviews, checklist.
- Planning and tracking test procedure development - detailed workplans.

Problem Reporting and Management

- Why track SPR's?
- SPR Process - describe the activities related to the lifecycle of an SPR, including the roles of people involved, the QUAD tool, reports, and data to record.

Testing Data and Analysis

- Quality Criteria - what is the data for? When do we ship?
- Planning and tracking metrics - parallel with development tracking.
- Analyzing defects - what can we learn from the data? Analyze escapes?

Continued on next page

Annotated Outline, 2nd Half-Day

Objective: *To familiarize the students with the activities related to executing test in the lab and the various domains represented at US&S.*

Introduction

- Overall review of US&S Testing - review of day 1 introduction - what testing goes on at US&S.
- Typical US&S System overview - description of the hardware from workstation to vehicle - examples, block diagrams, walk through labs?
- Typical scenarios for test set up versus field - what we normally have available in lab - what we use in field.
- Hardware testing overview/demo
- Database Checkout overview/demo
- Other testing overview/demo

Domain Knowledge

- Demonstrate a typical system - maybe two or three systems. Include basic CTC, train sheets, logging on/off, etc.
- Getting on a system
- Start-up/shutdown scripts, where can you test (lab, workstation).
- Terminology - what are the common terms used - what are project specific terms?

Tools for Testing

- How do we monitor the system? - Describe the Debug task, MSS instrumentation, tailing output files, etc.
- Pixie and Profile type tools - overview and demo?
- Code coverage tools - overview and potential use?
- Capture/Playback tools - overview and potential of Xrunner, PreviewX.

Configuration Management

- How do we know what we are testing?
- Overview of configuration management practices.

Test Reporting

Tracking and reporting test events.

Wrap - Up

- Where we are headed
- Improving the process

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (LEAVE BLANK)	2. REPORT DATE April 1997	3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE A Turbo-Team Approach to Establishing a Software Test Process at Union Switch & Signal		5. FUNDING NUMBERS C — F19628-95-C-0003
6. AUTHOR(S) Don McAndrews, Janice Marchok Ryan, Priscilla Fowler		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-97-SR-002
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/AXS 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES		
12.A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12.B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

Process improvement teams are often created and tasked to make complex changes in an organization, but are not provided with the tools necessary for success. This report describes what one team did to successfully install a software testing process in an organization. The principles of a turbo-team approach are introduced and a defined process for working in teams to introduce a new software technology is adapted from the prototype *Process Change Guide* developed by the Software Engineering Institute (SEI). Artifacts of this effort are included and described as examples for other teams to reference. Successes and lessons learned are described.

14. SUBJECT TERMS process improvement, technology transition, turbo-teams			15. NUMBER OF PAGES 90 pp.
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102