

Douglas G. Thacker

Douglas Thacker is the Manager of Desktop Quality Assurance at Liberty Mutual Insurance Company in Portsmouth, New Hampshire. He has 4+ years experience in managing and administering QA and change control processes within the Desktop computing environment. In addition to his QA background, Doug has 10+ years experience in IT and Production Support at both Liberty Mutual and Bath Iron Works Corporation. This combination of experience gives him valuable insight into the world of the IT customer, and why it's so important to "get it right the first time" when deploying software. He earned a Bachelor of Science degree in Civil Engineering at the University of Maine, and resides in Maine with his wife and two children.

Going Beyond QA: Total Product Readiness

Douglas G. Thacker
Manager, Desktop Quality Assurance
Liberty Mutual Insurance Company
douglas.thacker@libertymutual.com

Abstract:

The successful release of software requires more than just testing to ensure the product functions properly; success is also defined by how prepared the product is for advertisement, delivery, installation, training, support, etc. In this paper, we'll discuss how testing can be expanded to cover all aspects of Total Product Readiness (TPR). You'll be able to verify that each of these vital components has been properly planned and prepared for during the product development cycle. In this paper we will discuss:

- How to apply the Total Product Readiness process
- The key areas to include in the process
- The benefits of Total Product Readiness to your organization

With Total Product Readiness you can better ensure the successful introduction, end user experience, and supportability of any product.

Background:

The primary focus of software quality professionals has typically been producing defect-free products through rigorous testing. Indeed, looking at the presentations at the StarEast (Software Testing, Analysis and Review) 2002 conference this spring, a majority of them emphasize testing (testing methods, testing tools, test automation, web testing, etc).

I have been involved with software quality for about four years at Liberty Mutual. The quality process we put in place there includes traditional testing, but also encompasses other elements to ensure that our end users have a positive overall experience with any product entering our production environment. As I read articles in software quality publications and attend events like the StarEast conference, I began to realize that our approach to total product readiness is not a common approach in the software quality community. This surprised me, because the concept of total product readiness is very straight forward, fairly easy to implement, and produces significant benefits to an organization. Therefore, when SQE put out a call for papers, asking for success stories to share with other software quality professionals, I felt that our successful total product readiness program was that story.

What is Total Product Readiness?

Consider this. If you develop software, thoroughly test it, and deploy it to your customers/end users, will they be satisfied if:

- they don't know how to install it?
- they don't know how to use it?
- the help desk doesn't know how to support it?

Will your company or I/S organization be happy if the software product:

- clogs your network with excessive traffic?
- violates security policies or standards?
- significantly increases the volume of email through your servers?

If your answer is 'no' to either of these questions, then perhaps it time to consider Total Product Readiness (TPR).

So, what is Total Product Readiness? I define it as:

The process of validating that a product is sufficiently prepared for advertisement, delivery, installation, training, support, etc. to ensure the successful introduction, end-user experience and supportability of the product.

The Total Product Readiness (TPR) process should be the final step prior to production release of software products. Therefore it is expected that normal product development life cycle is followed prior to entry into the TPR process (unit testing, integration testing, system testing, acceptance testing, etc.). It is also expected that all preparation for the product's advertisement, delivery, installation, training, support, etc. has been completed prior to entry into the TPR process. The TPR process is not meant to be a "working" process where product readiness is completed, but rather, a checkpoint to ensure that the preparation has been completed as part of up-stream processes.

There are no "right" ways of implementing Total Product Readiness. Each organization has it's own processes, it's own policies to uphold, and it's own values of what is important. What is important for one organization to include in a Total Product Readiness process may not be important to others. However, I believe there are some common components that should be included:

- Independent Test Lab
- Network
- Help Desk
- Communications/Training
- I/S Security
- Software Distribution Services
- Operations

Our implementation of Total Product Readiness at Liberty Mutual is designed for products to be implemented and used by internal users within the company. As such, we also incorporate components into our process for our deployment team, break/fix team, level 2 support team and executive support team. Total Product Readiness could also be easily be implemented in organizations providing software to external customers, and they may choose to incorporate other components to suit their processes.

A key factor to the success of any Total Product Readiness program, regardless of how it's implemented, is the support and commitment of the organization's senior management.

How is Total Product Readiness Implemented?

Software products that are entered into the Total Product Readiness process are reviewed by a team of representatives from each of the departments who have a stake in the product's implementation. Each team representative has a signoff on each product, and no product can be certified for production release without signoff from each team representative.

In the next section I will discuss the role of each team representative. But first I will describe how the TPR process works, the role of the product sponsor and the TPR Administrator. Please refer to Figure (1) for a process flow chart of a typical TPR implementation.

Product Sponsor

The product sponsor is the person introduces the software product into the TPR process, and is ultimately responsible for getting the product TPR certified and implemented into the production environment. The sponsor must be intimately familiar with the product and should be involved with the product throughout it's testing and implementation planning. The sponsor introduces their product to the TPR team, works proactively with each of the TPR team members to obtain their signoffs, answers questions and resolve issues that come up during the certification process. A product sponsor may sponsor more than one product at a time through the TPR process.

TPR Administration

At the center of the process is TPR Administration. The administrators keep the track of all activity within TPR. Their primary role is receiving the entries from the sponsors, processing the entries and distributing them to the TPR Team, facilitating communication between sponsors and team representatives, as well as any necessary review meetings, collecting and keeping track of team signoffs, and exiting products once all the signoffs have been obtained. TPR Administrators also find themselves tracking and expediting issues, educating sponsors on the process, and reporting various statuses to management.

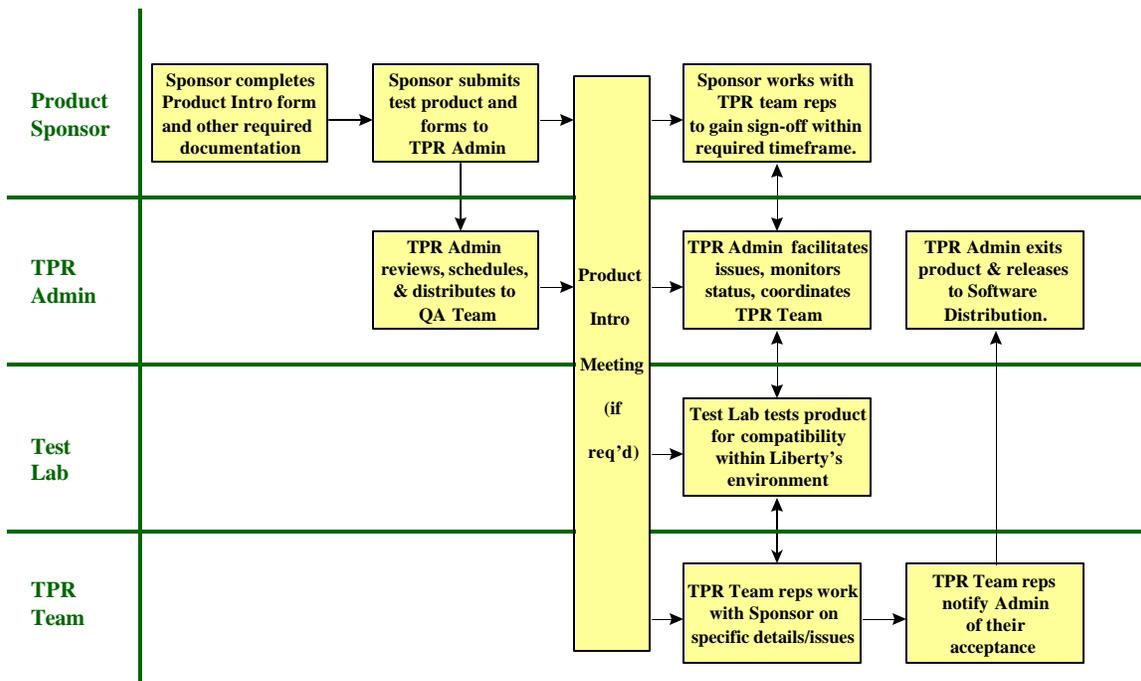


Figure (1): Total Product Readiness Process Overview

The TPR Process

As previously stated, it is expected that all the requirements for TPR certification have been adequately prepared prior to entry into the TPR process, so that each TPR representative can review their requirements and provide signoff in a timely manner. To start the TPR process, the product sponsor fills out a Product Introduction form and submits it to the TPR administrator. The Product Introduction form contains information about the product, such as the product name, version, description of the product or change,

who the user population of the product is, etc. It also includes sponsor contact information such as name, department, phone number, manager's name, alternate contact name, etc. The form then provides key information about the product which will allow each team representative to better understand how the product impacts their area. Based on this information, the team representative may be able to signoff immediately, or conversely, indicate that they are "N/A" on that particular product because it does not impact their area at all. In all likelihood, however, the team representative will need to follow-up with the sponsor to ask specific questions about the product and how it meets their requirements.

The TPR administrator receives the Product Introduction form, reviews it for completeness, and enters the product into a log or database. He/She then schedules a planned certification date when the TPR process will be complete and the product available to implement into production. The planned certification date will be determined by a standard service level agreement (SLA) that is put in place between the TPR team and the product sponsors. This SLA may vary from organization to organization, but we've found the following to work well for us:

New products (not previously TPR certified)	10 business days
Upgraded or fixed products that have been previously TPR certified	5 business days
Emergency fixes to resolve outages or other high exposure issues	24 hours

The TPR administrator distributes the Product Introduction form to the TPR team for their review process, indicating when the product is scheduled to be certified. The product sponsor then works with the team representatives to gain their signoffs. The product sponsor delivers the support scripts to the help desk representative, the software package and test scripts to the test lab, and any other necessary documentation as required/requested by the TPR Team. If needed, a meeting between the sponsor and the TPR team is scheduled to review the product as a group. This could be done for all products, or just those that are new or highly complex.

As each team completes their review of the product and the various requirements, they work out issues with the sponsor and any other impacted TPR group. When the review process is over, the team representative sends an email note to the TPR administrator, copying the product sponsor, indicating the whether or not they sign off on the product. The four types of notes they can send are:

- **N/A:** Introduction of this product has no impact at all to the area represented by that team member.
- **Signoff:** All requirements have been satisfied for the area represented by that team member.
- **Conditional Signoff:** The team member will signoff on the product, but attach certain conditions to the signoff. Conditions could limit how, where or when the product can be used, or require that certain outstanding issues be resolved within a specified timeframe.
- **Failure:** When issues arise that prevent a team representative from signing off, they can fail the product. A failure by any TPR team representative will result in the immediate failure of the product from the TPR process, allowing the sponsor to take the product back to the development team to continue working the issue until resolved.

The TPR administrator collects all the signoffs and makes sure the product stays on track for its planned certification date. Once all the signoffs have been received, the TPR administrator issues an email note to the product sponsor and the TPR team indicating that the product has been certified and exited from the TPR process. The product is now ready for production implementation.

Makeup of a Total Product Readiness Team

The makeup of a TPR team will vary depending on the makeup and needs of the organization, but will always have a product sponsor, a TPR Administrator (as previously described), and a team of reviewers representing their areas of responsibility within the organization. In this section I will describe the function

of these roles: what they are looking for in order to sign off, and what benefit their review provides to the product, the end user /customer, and/or the organization.

Independent Test Lab

The test lab performs several functions in the TPR process, providing a final quality check of the product to: validate the product installs properly and according to instructions, ensure the product works as intended, and make sure the product is compatible with the hardware its installed on and other software that has previously been installed.

The product sponsor must provide a test plan describing how to install the application (including any pre-requisite software required), and basic instructions on how to launch the application and perform basic functionality. This is necessary since the testers in the independent lab may not be familiar with the application and thus acting more like the intended end user of the product. The tester will follow the test plan loosely, deviating as they deem necessary to get a good feel for the product. The entire product will not be tested, but instead, a sampling to see if there are any problems. If problems arise, the tester will test further to find the extent of the problem or other problems. They will then work with the sponsor to resolve the problems, or in severe cases, fail the product so the sponsor can go back to the development team to fix the problems.

The test lab facilities should include a variety of devices that represent what the end user are using in the production environment. This is especially important if this type of testing was not performed prior to entry into the TPR process. Devices used in testing should be loaded with all the software that the end user will typically have on their computer, and these applications should be tested to ensure that installation of the sponsor's product does not corrupt or interfere with these other applications. Wherever possible, the test environment should be isolated from the production environment to minimize risk to production users as testing is performed, problems identified and resolved, etc. At the same time, the test lab environment should be set up to mimic the production environment as closely as possible to yield the most accurate test results.

Upon completion of testing, the test lab will issue an email note to the sponsor, the TPR administrator and the rest of the TPR team with the results of testing. The note will either indicate that the product passed or failed testing, and indicate what tests were performed. If the product passed, but certain issues or unexpected results were discovered, these will be added as notes to the test note. For example:

“Product X works fine on all devices containing Internet Explorer version 5.5 or higher, but produces an error on devices with older versions of IE. The product sponsor is aware of this, and indicates that the intended user group of the product all has IE 5.5 or higher. Support scripting has been updated to reflect this issue.”

Since the test note is also sent to the rest of the TPR team, they can act accordingly (the Help Desk can make sure the condition is scripted, training and communications can incorporate the condition into their materials, etc.).

Each problem or issue that can be corrected prior to production implementation reduces the number of calls by your end users to the help desk, as well as reduces the amount of downtime they experience. Issues discovered during testing that do not need to be corrected, but are communicated properly to the other TPR team representatives will help these groups be better prepared to prevent end user from having difficulties (through training, communications, etc.) or reduce the amount of downtime the end user experiences (through better scripting for the help desk or the operations groups). All of which result in savings for your organization, and a better experience for your end users

Network

The function of the network group is to perform network traces while the product is running to measure the amount of network traffic the application creates. The purpose of this testing is to see how this product will impact the network and other users on the network. If the product exceeds certain pre-defined guidelines for network utilization, then the software would need to be reworked to reduce the amount of network traffic; otherwise the application could be restricted or rejected. An example of a guideline for network utilization is:

“No application can consume more than 60% of the network for more than 5 seconds. In addition, no application, server or client can have a delay of more than 60 seconds that results in client inactivity”

Since the product is already set up to run in the test lab, it is natural for network testing to be performed in the lab during or immediately after testing is performed.

If the organization has different types or speeds of network connections in its production environment, the test lab should be set up to simulate these various network connection speeds. It is best to gauge the product's performance on the slowest of these links, since it can be assumed that performance would be better on faster links. By measuring the product's network traffic on a simulated network connection in the lab, you can get a pretty good idea of how it will perform in production. Keep in mind, however, that in the production environment there will be other network traffic in addition to the application you are testing, as well as the likelihood that more than one person will be running the app at the same time. These factors must be taken into consideration in the Network signoff of the application.

The benefit of performing this type of network testing and analysis is to minimize the impact to the end user. If a user is working with an application that takes more than 60 seconds to respond, the user is likely to think the product isn't working. If an application creates a lot of network traffic, this could slow response time for other users working on other products. If the network traffic gets too high, users could think the network is down or that their other applications aren't working and call the help desk. Depending on how the user describes the problem, the helpdesk could spend valuable time trying to resolve the wrong problem. Pinpointing the cause of these types of problems is difficult, time consuming, and expensive. Performing network tracing and applying appropriate network utilization standards can go a long way towards preventing this.

Help Desk

The Help Desk is the front line of support for your users, and in the best position to provide help when your users encounter problems. Even if your application is bug free, you should never underestimate your users' ability to create new problems on their own. Therefore, during the development and testing of your application it is very important to determine every conceivable problem your users could encounter. For instance:

- What happens if they enter the wrong password?
- How do they reset their password if they forget it?
- What happens if they enter invalid data, or do some operation out of sequence?
- What happens when printing doesn't work?
- What happens if they are disconnected from the network when they use the application, or the network goes down while using the application?
- How do users re-install the application if they accidentally deleted it?

If the help desk is not prepared to receive these questions, they will not know how to answer them. The user will be more frustrated because it will take longer to get an answer and get back to work. Yes, the user will blame the “Helpless Desk” for not being any help, but they will also blame the software for having the problem. Therefore, before the Help Desk signs off on any application in TPR, they need to be satisfied that the scripts they have are sufficient to help these end users.

The product sponsor needs to develop a list of problems that the user could encounter, including all known error messages within the application. Then they need to develop each problem into a support script that the help desk can follow. The elements of a good help desk script are:

- **Problem Description:** Briefly describe the symptoms the user would see. Phrase it the way the user would. If there are more than one way for a user to describe a problem, list each description.
- **Problem Cause:** What is responsible for the user's problem? "User Error" may be a tempting thing to list as a cause, but perhaps the real cause is "unclear error message" or "insufficient user training". If the cause is something within the application code or error handling routines, briefly describe it. The end user will be curious as to what's causing the problem, and will be more trusting of the solution if they understand the cause. (They may also be more likely not to repeat the same problem again!)
- **Problem Resolution:** List the steps to resolve the problem, making sure the language you use is not too technical. Resolution steps should be numbered for clarity, and in outline hierarchy if necessary. The resolution should also include a way to verify that the resolution actually works. (validation step)
- **Escalation Path:** If the resolution does not work, or the user has a problem that is not included in the scripts, the help desk technician needs to know where to escalate the problem to for further help. This may be directing the user to call someplace else, or for the help desk to call on the user's behalf.

Keep in mind that the help desk is not an expert of your application, so the more information you provide, the better they will be able to service your customers. Also keep in mind that many help desks require some amount of lead time before they will actually use new scripts so that they have time to share them with all their technicians and learn them.

Communications/Training

This group would make sure that adequate communication to the end user is ready prior to the application's implementation, and that training is available as necessary before signing off on the application. The communication serves multiple purposes:

- Notifies the users of the application's availability
- Provides a description of the application and lets the user understand why they need it
- Provides detailed instructions on how to install the application
- Tells the user how to obtain documentation or training for the application
- Tells the user where to go for technical support

If training is necessary for an application, this group would validate the type of training (instructor led, videotape, computer-based tutorial (CBT), web-based tutorial (WBT), documentation, etc.), when and where the training is offered (before the application's implementation, at time of delivery, or ongoing as needed), and who developed the training curriculum.

Without the review of communications and training, the quality and timeliness of both would go unchecked, potentially causing end-user confusion, complications in implementation, and ultimately higher support costs.

I/S Security

This is an obvious, but often overlooked aspect of application development/implementation: Does the application conform to your organization's policies for I/S Security? There are many areas of security concern for organizations, from application security, handling of sensitive data, database security and internet security. Questions the security group may ask include:

- Is access to the application restricted? If so, how? If not, should it be?
 - What is the ID/Password process?
 - How does the application authenticate the user? (internal to app, query to server, etc.)
 - How are passwords changed or reset? (automated, Help Desk, other?)
- Does the application store data on a server or in a central database? Is the data secure?
- Is the application or data accessible via the internet or your organization's intranet? Have proper safeguards been put in place to protect these assets?
- Is there any sensitive data involved, and how is it protected during use or transmission?

Without an independent review by I/S Security, many security exposures would be missed, creating frantic users and many calls to the help desk.

Software Distribution Services

Although this is usually the last step in the product development/implementation process, and occurs after the product has completed its TPR certification, it's important to make sure the delivery of your application is adequately planned and ready. The group responsible for software delivery needs to make sure the following is set up prior to signing off on the application:

- Has the correct user group of the application been identified, so that the right people receive the software?
- What is the most efficient method of delivering the software to the target audience?
- When will the software be delivered, and will it meet the schedule needs of the business?
- Will the delivery be coordinated with the end user communication?

Without a review of this criteria quality and timeliness of delivery of software to the end users would be at risk. A delay in the implementation of software to the appropriate users could result in loss of productivity for those groups, costing your organization money.

Operations

The operations group in your organization may be centralized or distributed by function, so they may sign off individually or as a group. These groups are responsible for the day to day operations of your infrastructure, so they should review each product to make sure they properly integrate into the existing production systems. Examples of this are:

- Does the product interface with the organization's email system? If so, how, and what volume of messaging is expected?
- Does the product reside on a server or have a server component? If so, is there space on the server, and are the proper permissions set up?
- Is there a web component to the product? If so, is there space on the web server, and does the application comply with the organization's web policies?
- Does the product or any of its data need to be backed up? If so, where is it being backed up from, and how much space is needed?

A review by these operations groups will ensure that the services the end user depends on will continue without disruption.

Procurement

It is recommended that organizations put 3rd party products (hardware and/or software) through the TPR certification process as well as software developed in-house. The procurement team would make sure that:

- the product sponsor can supply a Proof of License for the product,

- the Proof of License is for the number of licenses purchased, and
- the number of licenses purchased matches the number of intended users.

The procurement team can also ensure that a contract is in place with the 3rd party vendor, if necessary.

Conclusions

Software development and testing tends to focus exclusively on the quality of the software product itself, and the processes and methods used to produce the software with a minimum of defects or bugs. This is definitely important for the success of any software product, but if success is defined as the successful introduction and end user experience of the product, then several other factors must be considered and carefully planned:

- **Advertisement:** How will the product be communicated to the end user so that they know what to do with it?
- **Delivery:** What is the best method of getting the product to the end user, and how do you get it to the right user at the right time?
- **Installation:** How will the user know how to install the product, and will the installation go smoothly?
- **Training:** How will the users learn how to use the product?
- **Support:** Where will the users go to resolve problems or answer questions they have? Will the help desk be able to help them, and if not, where else can they go?

Total Product Readiness is a process where these other factors are reviewed by the groups impacted by these types of issues before the product is introduced into the production environment. All I/S groups needs to be fully ready and aligned when any new product is implemented. The benefit to the organization is the knowledge that when a product is implemented, it has been thoroughly tested in the environment, it doesn't degrade network performance, appropriate scripting is in place for the help desk, the application follows I/S Security rules, and it goes to the end users it needs to.

About the Author

Douglas Thacker is the Manager of Desktop Quality Assurance at Liberty Mutual Insurance Company in Portsmouth, New Hampshire. He has 4+ years experience in managing and administering QA and change control processes within the Desktop computing environment. In addition to his QA background, Doug has 10+ years experience in IT and Production Support at both Liberty Mutual and Bath Iron Works Corporation. This combination of experience gives him valuable insight into the world of the IT customer, and why it's so important to "get it right the first time" when deploying software. He earned a Bachelor of Science degree in Civil Engineering at the University of Maine, and resides in Maine with his wife and two children.

Disclaimer

The opinions expressed in this paper are the author's, and not those of Liberty Mutual.