Incorporating Usability Testing into the Documentation Process

CHRISTI-ANNE POSTAVA-DAVIGNON, CANDICE KAMACHI, CORY CLARKE, GREGORY KUSHMEREK, MARY BETH RETTGER, PETE MONCHAMP, AND RICH ELLIS

The MathWorks, Inc. produces over 65 software packages that scientists, engineers, and students use for technical computing. The documentation for these packages consists of thousands of pages, available in both online and printed format.

Cross-functional teams carry out software development at The MathWorks. These teams include software engineers, quality engineers, usability specialists, and technical writers. The writers are involved throughout the development process, contributing to the design and the implementation of the final product.

The documentation for a given product typically consists of tutorial-style “getting started” sections oriented toward new users, task-oriented “user guide” topics, and reference material. For the most part, writers single-source both online and printed documentation from FrameMaker files.

The MathWorks has shifted its emphasis for customer documentation from printed to online. Users view the documentation through a customized help browser that provides navigation with a table of contents, index, and full-text search.

The creation of the Documentation Usability Team

For a number of years, a few writers at The MathWorks had been performing documentation usability tests. However, most writers in the group did not test their documentation. Documentation usability testing was viewed as a nice thing to do if you had the time.

When a small team of writers began moving core documentation into online format, they encountered many issues with respect to the content, organization, and presentation of the information. Because most writers in the Documentation Group had little experience with this type of online documentation, they did not know how to resolve these issues. They discovered that usability testing provided the empirical evidence they needed to drive the decision process.

Writers involved in the initial project felt documentation testing was a valuable experience in itself. They found that observing testers using the products enriched their understanding of how customers use documentation to accomplish real tasks. In light of this experience, the writers found themselves reconsidering their previously held assumptions about what users read and in what order, what attracts attention and what does not, and what kind of information users want.

Inspired by the success of this approach, the writers wanted to spread the word to the entire Documentation Group. To accomplish this goal, they decided to create the Documentation Usability Team to help writers incorporate documentation testing into their writing process.

The Documentation Usability Team’s permanent members include a usability specialist and writers representing the company’s different product areas. In addition, writers testing documentation attend a number of meetings that focus on their testing. The team’s charter is to maximize return on documentation usability testing by

- Assisting Documentation Group members who are new to documentation testing
- Refining the techniques that the team uses for documentation testing
- Providing a forum to discuss findings and issues resulting from tests
- Publishing information to the rest of the Documentation Group based on observations made during tests and discussions
- Maintaining a database where test plans, findings, and issues are recorded

Manuscript received 11 June 2002; revised 10 July 2003; accepted 1 August 2003.
Incorporating Usability Testing into the Documentation Process

The ongoing goals of this team are to maintain an easy-to-follow procedure for writers performing usability tests, and to create a database of testing results that all writers can use as guidelines.

**Documentation usability testing**

The Documentation Usability Team uses a testing process that focuses on two major aspects of documentation: the content and the presentation of technical information. (A different team in the organization designs and tests the help browser.) The Documentation Usability Team uses documentation testing to determine how successfully customers can perform realistic tasks with the products, while relying on the documentation to provide direction.

The team conducts most of its testing using online documentation because customer feedback has shown that users refer primarily to online help. In addition, writing online documentation is an area in which most writers at The MathWorks have the least experience; therefore, testing in this environment provides the greatest return. The team has also applied its testing methodology effectively to other forms of documentation, such as printed manuals.

At The MathWorks, the Documentation Usability Team and product usability specialists work with writers to conduct documentation tests. The MathWorks has an established product usability testing process that the team built on to create its documentation testing methodology. The team employs discount methods and encourages team involvement to get the highest return from testing (Nielsen 1994; Butler and Ehrlich 1994).

**Common testing roadblocks**

While many writers accepted the benefits of including documentation testing in their writing process, some were reluctant to test their documentation. There were several reasons for this reluctance.

- **Some writers were not eager to have their work subjected to public scrutiny before it was final.** The team countered this argument by pointing out that usability testing is a tool intended to help writers be more successful. The earlier a problem is caught, the easier it is to fix. It is better to find problems before customers do.

- **Some writers insisted that testers must read the entire product documentation before being expected to complete a task.** The team believed that this was not a reasonable expectation because the technical support group at The MathWorks maintained that users rarely read much of the documentation before using the product.

- **Some writers thought usability testing would take too much time.** Conducting a series of well-executed usability tests does require a fair amount of work. In addition to the time required for preparation, each test in the series lasts for 1–2 hours, followed by a debriefing session. Also, the results of the test often compel writers to modify their documentation and perhaps change approaches they use throughout the product documentation. Writers are encouraged, however, by the testimonials of writers who have completed testing projects and have found them to be considerably helpful. These testimonials confirm the very high return on the effort invested in documentation testing.

- **Some writers were intimidated by writing the test plan and tasks.** For first-time testers, one of the most difficult parts of the process is creating the test tasks. Writers often assume that usability tasks need to be especially clever, profound, or somehow all-encompassing. To ease these fears, the team helps writers create test tasks.

The Documentation Usability Team found that once writers successfully completed their first test sessions, their fears and misconceptions were significantly allayed. The team established a process to help writers over the initial learning curve, created an internal Documentation usability testing guide, and developed a process that steps writers through planning and performing a test.

**PLANNING THE TEST**

When the Documentation Usability Team was formed, the members included both experienced documentation testers and writers with no documentation testing experience. As the inexperienced members conducted their first tests, the team carefully noted the difficulties they encountered, the steps in the process that confused them, and their reactions to successes and failures. Based on this information, the team developed a series of steps for writers to follow. These steps are discussed in more detail in the Documentation usability testing guide.

**Steps for writers to follow**

When writers sign up to perform their first documentation usability test, the Documentation Usability Team guides them through the process using a series of steps. This general process does not stray far from what is recommended in the literature (Rubin 1994; Dumas and Redish 1999).

**Step 1: Getting started.** Writers get a feel for documentation testing.

- **Read the Documentation usability testing guide**—The guide answers many questions typically asked by writers when they begin testing.

- **Observe a documentation test**—The team encourages writers to attend a documentation usability test.
Step 2: Formulating test objectives. Writers formulate their test objectives and plan the test.

- Meet with the Documentation Usability Team for the first time—The team helps writers outline their test plans and formulate questions they want to answer about their documentation. Writers receive the Writer’s Checklist (Appendix A) and the Test Plan Template (Appendix B), which they then complete. The checklist and template help ensure that writers are prepared for testing. Each section in the Documentation usability testing guide has an associated section in the Writer’s Checklist.

- Work with a usability specialist—After the first meeting, writers work with their usability specialists to specify tester profiles, determine what tasks are appropriate, discuss how to run the tests, and schedule the test sessions.

- Complete the test plan—Writers may revise their test plans based on discussions with their usability specialists.

Step 3: Writing test tasks. Writers create and revise their test tasks.

- Write tasks and create testing materials—Writers work with their usability specialists to develop scenarios and create tasks.

- Meet with the Documentation Usability Team for the second time—When the test plan and tasks are ready, writers and the team review the tasks. Team members test the tasks to debug any problems and ensure that the usability test will run smoothly. Writers discuss the roles they will play in the test sessions.

- Revise tasks and prepare for the test—Writers may revise their tasks as a result of the second meeting. The team helps writers rehearse their tasks, check that the computer in the usability lab is set up properly for the tests, and create a “teacher’s edition” of their tasks with solutions and behaviors that observers should watch for.

Step 4: Conducting the test. Writers conduct their tests with the help of their product usability specialists, and everyone discusses the results.

- Conduct the test—A test is usually conducted with the writer whose documentation is being tested and the usability specialist in the lab with the tester. Interested parties and at least one member of the Documentation Usability Team observe the test from the observation room. The observation room contains remote monitoring equipment that enables observers to hear the conversation and see both the computer screen and the tester’s facial expressions.

The usability specialist manages the test session. This responsibility includes explaining to the tester how the test will be conducted, discussing the test objectives, and describing how the tester should interact with the software and documentation during the test. Recall that the goal of these tests is to determine whether testers can perform tasks successfully while relying on the documentation provided.

- Debrief the participants—Immediately after conducting each test, the writer, usability specialist, and observers meet to debrief. During this meeting, everyone discusses and categorizes the observations.

Step 5: Documenting the results. Writers discuss significant findings with the team and document results.

- Meet with the Documentation Usability Team for the third time—The team discusses the issues categorized in the debriefing meeting as being of general interest to documentation writers. The team draws conclusions with respect to the observations and further categorizes the findings either as potential guidelines for writers or perhaps as data points worth noting.

- Record information in the database—Team members work with writers to enter information into the team database. Writers take screen captures of documentation sections before any changes are made. When they then revise sections of their documentation based on test findings, they also capture images of the updated sections so that they can include the “before” and “after” views in the database. These are particularly useful to other writers and help to convince them of the benefits of documentation usability testing.

Creating the test plan
The utility and effectiveness of test plans is well documented in the literature (Rubin 1994; Dumas and Redish 1999). The test plan (see Appendix B) describes the objectives of the documentation usability test. It includes a description of the product, a statement of the test purpose, a profile of appropriate testers, questions the writer wants the test to answer about the documentation, and a description of the testing environment needed (for example, special hardware or software versions). The test plan serves several purposes. It helps the writers formulate the questions they would like their tests
Starting testers with a simple task relieves some of the anxiety that is inevitable when working in a laboratory setting.

The documentation explains how to do each of the steps listed, but the tester must first find the appropriate sections, understand them, and then apply the correct techniques to the data provided.

TESTING METHODOLOGY

The goal underlying testing with the Documentation Usability Team is to reconstruct a typical user experience with the product and documentation being tested. Of course, a usability lab is not quite the same environment as a tester’s office. A certain amount of pressure is placed on a tester by the fact that a number of people are observing the test. To calm a tester’s fears, the usability specialist emphasizes that it is the documentation that is being tested, not the tester’s abilities.

The team’s testing methodology differs from the norm in that testers are actively encouraged to use the documentation whenever they have a problem. For example, testers sometimes state that they would ask a colleague what to do when they cannot figure out how to solve a problem on their own. In such a case, the usability specialist encourages the tester to use the documentation to find the answer. In addition, the team begins documentation tests by asking the tester to focus on the documentation and how it helps in completing the tasks. Testers are told that for this kind of test, observers are less interested in their comments on the product than on how the documentation helps them to use the product.

During the test, the usability specialist minimizes the amount of time testers are allowed to search unsuccessfully for the right page or section in the documentation. While this approach might not give observers an accurate gauge of when and how users actually navigate through the documentation in the real world, it does enable them to observe and analyze testers’ experiences with the documentation when they do use it.

The team has consciously made the decision not to test the navigation of the documentation in these tests but rather the content. The team has established a protocol for prompting testers to find the areas of interest in the documentation, and to help them focus on critiquing the documentation rather than the product. Much of the methodology is based on the work done by Chinoporos and Butler (1997).
Sometimes the team does not use the usability lab if the testing requires special hardware or if the lab is not available. Testing can be performed in the writer's office with a small group of observers. Sometimes testers feel more comfortable in this type of situation because there is no one-way mirror concealing an unknown group of observers. It is important to note that you do not need a usability lab to successfully conduct documentation testing, as long as you can make the tester feel comfortable being observed (Scanlon 2002).

Roles of test participants

The writer and usability specialist for the product are present in the lab during each test, with interested parties observing from behind a one-way mirror in an adjacent room. Although testers are always informed that others are observing, keeping the number of people in the testing room to a minimum reduces the ambient noise and distractions.

The usability specialist runs the test and is responsible for most of the interaction with the tester. Therefore, it is necessary for the writer and the usability specialist to be able to communicate in subtle ways. For example, the writer might notice that the tester keeps missing key information on a particular page, and might want the usability specialist to tell the tester that the answer is in fact on the current page. In other cases, the usability specialist might want to confirm with the writer that the tester has gone astray.

As writers watch their work being tested, they might be tempted to engage in conversation with the tester, or to give clarifying information. However, writers should avoid clarifying points in the documentation or discussing various aspects of the product implementation. The team encourages the writers to hold comments until the end of their tests, thereby giving testers more time to talk and to work out the test tasks on their own.

Testers often want to engage writers in discussions about how the product works. To keep the tester focused during the tests, the team maintains a database that records the following information for each test session:

- **Product-related issues**—Although the goal of these tests is not to test the product, the usability specialist notes any product issues that arise for discussion with the product development team.
- **Documentation-specific issues**—The writer collects findings that are specific to the documentation being tested.
- **Help browser issues**—Although the goal of these tests is not to test the help browser, the usability specialist notes any issues related to the functioning of the help browser that arise and discusses them with Help Browser Team.
- **Technical communication issues**—The Documentation Usability Team discusses and documents issues that are related to the art of technical communication in general. See the “Typical database entry” section below for an example of such an issue.
- **Testing methodology issues**—The Documentation Usability Team also collects issues related to its testing methodology. The team uses these observations to improve the way it conducts tests.

**TEST RESULTS DATABASE**

One of the most important duties of the Documentation Usability Team is to relay test findings to the other members of the Documentation Group. The team must also synthesize the data collected across tests to find common threads, issues, and solutions to issues. To do these things, the team maintains a database that records the following information for each test session:

- Test plan
- Tasks used in the test
- Technical communication and testing methodology issues

The team implements the database using a collection of HTML files that can be indexed and searched. Anyone in the company can access the database using a Web browser.

**Results**

The test plan and the tasks show the context in which the team obtained the results. The results documented in the database are the observations deemed to be of general interest to technical communicators. These results are further categorized as issues, data points, or success stories.

An **issue** is a problem with some aspect of the documentation that the team believes it knows how to improve on. Issues usually result in corrective action by the writer, the result of which is usually included in the issue description. When sufficient evidence exists, issues can become the catalysts for guidelines established for the Documentation Group.

A **data point** is an observation worth recording, but the team does not feel it understands the situation enough
to draw a conclusion or to take corrective action. If additional related data points are collected, a data point can be elevated to an issue.

A success story is an observation of an interaction with the documentation that leads users to the knowledge they are seeking in a reasonably painless and rapid way. Success stories might also lead to the establishment of guidelines.

In addition to assigning observations to specific categories, the team gives each item a status so it can track any actions that need to be taken:

- Needs resolution
- Under consideration
- Guideline proposed
- Guideline adopted
- No action required

Periodically, the team shares testing results with the rest of the Documentation Group.

---

**TABLE 1: TYPICAL DATABASE ENTRY—LINKS IN GETTING STARTED MATERIAL**

<table>
<thead>
<tr>
<th>Product and document</th>
<th>Creating graphical user interfaces (GUIDE), “Getting started” chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief description of issue or finding</td>
<td>In the “Getting started” example, testers followed links that took them out of the example and did not realize they had left the example.</td>
</tr>
<tr>
<td>Category (issue, data point, success story)</td>
<td>Issue</td>
</tr>
<tr>
<td>Usability specialist and writer testing doc</td>
<td>Mary Beth Rettger and Rich Ellis</td>
</tr>
</tbody>
</table>
| Discussion and conclusions (include link to screen shot of page in question or how to reproduce it) | Tester was working on getting started example. Clicked on link out of curiosity, which brought him out of the getting started example. ♦ He did not know that he had left the example. ♦ He did not need to follow the link. ♦ He did not return to the example.  

We have seen people get distracted by links before. They click on links because they think they need to or they think it is part of the flow of the example. |
| Guideline | In “Getting started” material, add links to information only if it is absolutely necessary that the user have this information at that time, and if you cannot include it in the body of the example. For example, in these cases links make sense:

♦ Single-sourced material—a link to supported data acquisition boards, which is maintained at a specific location.
♦ Long code examples or lists—the GUIDE “Getting started” example links to the application M-file section, which shows where to add the initialization code. |
| Link to tasks and test plan | GUIDE Task One |
| Related bug tracking numbers | n/a |
| Report author | Rich Ellis |
Typical database entry
Table 1 illustrates a typical database entry for an issue. In this case, adding unnecessary links from a tutorial to reference material often led the tester astray. The team observed that many testers click any link that looks remotely interesting, and they may never return to the tutorial. The format for the database entry is similar to others in the literature (Dumas and Redish 1999; Pernice and Butler 1995).

CONCLUSION
Many writers at The MathWorks who perform usability testing on their documentation go on to incorporate documentation testing into their writing process. They feel that they have improved considerably as writers, as well as having improved specific aspects of their documentation. In fact, many writers do not want to release documentation without subjecting it to documentation testing any more than the company would release software without quality testing.

The Documentation Usability Team will likely begin to shift its focus from ushering writers into the process of documentation testing to providing a constant flow of usability information on the MathWorks documentation. The use of real-world tasks ensures that observations made during testing reflect issues with many aspects of our products. TC

ACKNOWLEDGMENTS
The MathWorks Documentation Usability Team would like to thank the editorial staff for their assistance in preparing this article for publication.

REFERENCES


APPENDIX A: WRITER’S CHECKLIST

Step 1: Getting started
☐ Read the Documentation usability testing guide.
☐ Observe a documentation test.
☐ Set up your first meeting with the team.

Step 2: Formulating test objectives
☐ Meet with the team for the first time.
☐ Meet with your usability specialist.
☐ Complete the test plan.

Step 3: Writing test tasks
☐ Write tasks.
☐ Write a scenario.
☐ Write answers to tasks.
☐ Meet with the team for the second time.
☐ Rewrite tasks.

Step 4: Conducting the test
☐ Meet with your usability specialist a second time.
☐ Make copies of the test.
☐ Set up test environment and room.
☐ Conduct the test.

Step 5: Documenting the results
☐ Meet with the team for the third time.
☐ Record information in the database.
APPENDIX B: TEST PLAN TEMPLATE

What is (your product)?
A brief description of the product for observers and those reading your results who aren’t familiar with the product; text from the introductory sections in your documentation may suffice.

Test purpose
A paragraph about why you are testing your documentation, which usually includes the general topics you plan to test.

Tester profile
A list of what you’re assuming the tester knows and doesn’t know. This section should include both ideal and actual tester profiles.

Questions to answer
A list of questions you want to answer by testing your documentation. Ask your test observers to read this list of questions so they know what to focus on during the test.

Testing environment
A checklist of things you need to prepare or set up for the test. You and your usability specialist can refer to this checklist when setting up the tests. Things to consider listing include:

- Software
- Hardware
- MAT-files, Simulink model files, and so forth that are necessary to complete the test tasks
- Copies of the test tasks for the tester and observers
- Copies of the test plan for observers

CANDICE KAMACHI graduated from MIT in 2000 with a BS in mathematics. She then began working as a technical writer at The MathWorks, where she joined the documentation usability team and STC. She is now working on a start-up hobby and educational robotics company, Pololu Robotics and Electronics. Contact information: candice@pololu.com

CHRISTI-ANNE POSTAVA-DAVIGNON is a technical writer at The MathWorks. She writes documentation for fixed-point products in the DSP/communications and control design automation areas, and currently leads the documentation usability team. She holds an MS in physics from Michigan State University, and a BS in physics and a BA in philosophy from the University of Richmond. She is a member of the Boston Chapter of STC. Contact information: cpostava@mathworks.com

CORY CLARKE received her BS and MS degrees in mechanical engineering at Boston University. While in school, she interned at C. Raymond Hunt Associates, a naval architecture firm in Boston. After graduation, she began work as an application support engineer at The MathWorks. After nine months, she transferred into the documentation group to pursue her interest in technical writing. For the past two years, she has worked as a technical writer in the areas of control design automation and DSP/communications. She is a member of the Society for Technical Communication. Contact information: cclarke@mathworks.com

GREGORY KUSHMEREK is a senior technical writer at The MathWorks, documenting their internal integration tools and processes. He holds a graduate certificate in technical writing from Northeastern University, which he obtained after earning his BA in history from the University of Colorado, Boulder. He volunteers with non-profit and government agencies, helping them reach their audience in the electronic world. Contact information: gkushmerek@mathworks.com

MARY BETH RETTGER is director of the usability group at The MathWorks. Previously, she ran the usability group at Lotus Development. She is a past president and board member of the Usability Professionals’ Association and is a frequent contributor and reviewer for UPA and SIGCHI conferences. She holds an MBA from Northeastern University, and a BA in psychology from Brown University. Contact information: mrettger@mathworks.com

PETE MONCHAMP is the documentation manager for the writers in the control design automation group at The MathWorks. He is a senior member of the Society for Technical Communication and an online competition judge for the Boston chapter. Before a career change to technical communication, he worked as a research scientist in environmental analytical chemistry. Contact information: pmonchamp@mathworks.com

RICH ELLIS is a principal technical writer at The MathWorks. He has more than two decades of experience as a technical communicator and software developer working for companies such as GTE, Apollo Computer, Hewlett-Packard, and O’Reilly & Associates. He currently leads the Integral Information Architecture for Contextual User Assistance project at The MathWorks. Contact information: rellis@mathworks.com