

DOCUMENTE

through the Di

BY MARILYN B. KLOSS
Boston Chapter

The “discovery process”? What does that mean to technical writers? At the company where I work, SolidWorks Corporation, it means that the writers learn about the SolidWorks software not by asking the engineers or writing to specifications (that is, working from definition documents), but rather by gathering information about the software’s functionality through various sources and exercising the software themselves. Because they are involved in the development process from beginning to end, they are equipped to write better documentation. The advantages of the discovery process over writing to specifications include the following:

- Better documentation because the writers are familiar with the software
- More complete coverage because the writers know the intent, use, and evolution of functions
- More accurate descriptions because the writers work with the latest software implementation
- Better communication between engineers and writers because the writers are considered active members of the development team

- More dedicated writers because they have a sense of product ownership

To develop accurate and complete documentation, the writers have to know what new functions are available or what existing functions have been altered. We have developed several mechanisms for learning about new projects and implementations. The writers do discuss functionality with product definition, design, and quality assurance engineers—but not as the primary means of learning.

Our Process

The best way to illustrate the discovery process is to show it in action at SolidWorks Corporation. As is typical in many small startup companies, in the early days of SolidWorks, engineers were too busy to spend time with writers, time was short for both engineers and writers, and the writing staff was small. In addition, there was some uncertainty about what the software would do and how it would perform. In this environment, the best way to determine the functionality of the software was to try it out. The process has become more structured over the years, but it is still evolving.

The first writer coined the term “dis-

covery process” when she started describing to job applicants the way she uncovered information about the software. Writers from large, established companies typically worked with specifications, mature software, and engineers who had time to talk to them. Sometimes these writers did not even use the software or hardware they wrote about. Using the term “discovery process” was a way of describing the very different approach these writers would face if they were hired at SolidWorks.

Although SolidWorks Corporation and the SolidWorks software have matured over the years, the discovery process has worked so well that it is still the primary means by which writers gather and verify information.

Documentation Deliverables

The SolidWorks software is a 3D CAD (computer-aided design) package in which users develop 3D solid models of parts and assemblies, turn the models into 2D drawings, and export the files in various formats, often to production machines. Around the core product are a number of Web and Internet initiatives and third-party products. The audience for the documentation is mechanical

NOTIFICATION

discovery Process

engineers, and the documentation deliverables consist of the following:

- *Getting Started* book (for new users)
- *What's New* book (for current users)
- *Online User's Guide* (context-sensitive reference manual in HTML format)
- Online tutorials (step-by-step procedures)
- Online design portfolio (discussions of approaches to design issues)
- Online help for various add-in products
- Release Notes, Read This First, and other small documents

The Documentation Team

With such a technical product, one might expect the writers also to be engineers. In fact, some writers do have engineering degrees, but while an engineering background is helpful, it is not required. Some writers have a technical support background, some have more experience with software applications such as graphics packages, application program interfaces, or HTML, and others are primarily writers with skills in editing, proofreading, indexing, grammar, and punctuation. All have experience with computers. With a team of seven writers, each can contribute different expertise to the effort. Peer reviews

take advantage of the writers' diverse skills. All writers learn to use the software so they can test its functionality at every step of the writing process.

New writers begin by working their way through the current *Getting Started* book, which is a series of examples introducing various functions of the software, and a set of supplementary online tutorials. Then they take a five-day training class given by the SolidWorks training department. All this training gives the writers a foundation for running the software on their own.

Sources of Information

The writers turn to various sources for clues and pointers to functionality that they must test by using the SolidWorks software. The following list of resources is more or less in order of importance and frequency of use.

- Descriptions of programming submissions
- Dialog box database
- Product definition project list
- Project database
- Project specifications
- Software performance report (SPR) database
- Quality assurance (QA) testing reports
- E-mails from engineers

Descriptions of Programming Submissions

The writers' primary source of information is a listing (named "Summary.txt") that contains descriptions of all submissions by engineers and writers. A new listing is generated automatically every time the software is rebuilt, which is usually every day or so. Engineers place programming files on the server in a specified folder that is gathered up during the build process. (Writers submit online help and document files.) The wizard that directs the submission process asks for a description of the change or addition, then includes the description in the Summary.txt file. The listing is routed to the engineers and writers by e-mail.

In the early days of SolidWorks, each writer analyzed the Summary.txt file individually, looking for clues and following up with investigations. Later, as there were more writers who were responsible for specific subject areas, one of the writers developed a spreadsheet that broke out information from the Summary.txt file into appropriate subject areas. Individual writers then investigated the items under their subject areas, entering notations, dates of completion, and other data to track progress of the functionality and the documentation.

Currently the Summary.txt information is in a database in Microsoft Access. One person examines the Summary.txt files and transfers information into the database. The Summary.txt database generates reports by subject, status, writer, and version.

The writers check the database periodically and test changes in the latest build of the software before writing about them. Often the information is cryptic enough to require investigation, sometimes by finding other documents and sometimes by consulting the engineers.

Dialog Box Database

All of the application's dialog boxes contain a Help button that brings up the appropriate help topic. At each new release of the software, one of the writers compares current dialog boxes with those in the previous release and enters the information into a database. The database lists the dialog boxes with their names and titles and indicates which are new and which are to be reviewed to see if they need updating. The writers check the dialog boxes in the software and indicate in the database when they have reviewed or revised related online help topics, and they also note any topics to be removed because the dialog boxes are obsolete.

Product Definition Project List

The product definition engineers listen to customers, sales and marketing personnel, and technical support and QA engineers. On the basis of information they receive from these discussions, they propose projects for adding new functionality to the software. For each new release, the manager of the product definition department produces a confidential list, for internal use only, of the project titles. The list includes projects completed, in process, or just in the planning stage.

The writers use the product definition project list as a top-level checklist for the *What's New* book and the online *What's New* topics. However, the writers must verify the functionality in the software before including items from the list in publications; sometimes projects are not completed in time or the scope of projects changes.

As the
writers
feel more
ownership of
the product,
they are ever
more committed
to making
it better.

Project Database

The research and development (R&D) department maintains a database of projects that includes brief descriptions, names of engineers, and start and completion dates.

Several types of reports are available. For the writers, a report sorted by subject area is useful for tracking project progress and testing in their areas of responsibility. When projects have been completed and are passing their tests, the writers can proceed confidently. If the writers have questions about particular projects as they test the software, this list can tell them which engineers to consult.

Project Specifications

Project specifications are descriptions of the proposed functionality written by product definition engineers. These descriptions are useful sources of engineering background, intent, clues to the user interface, and terminology.

Since these descriptions are usually written before the engineering work begins, the final functionality is not always as specified. On the other hand, some projects are never described, and

some descriptions are written after the project is under way. For some projects, the engineers do not know what is possible or practical until they have investigated. For all these reasons, the writers must verify the specifications by running the software and testing the functionality.

Software Performance Report (SPR) Database

The SPR database provides information on problems associated with a project. For the writers, this is usually secondary information that may confirm problems they've discovered while testing the software, or that indicates that the functionality is changing from the original intent. The SPR records often include sample models that can be useful for testing. Generating a report of all problems outstanding on a project can give the writer an idea of the project's progress. The official status of a project is also recorded in this database.

Quality Assurance Testing Reports

After the development engineer indicates that a project is completed, the QA team begins testing.

Testing is carried out at two levels: Unit Test and System Test. A project must pass Unit Test before System Test begins. Unit Test evaluates a project internally for its usability and the minimum functionality defined in the project specifications. System Test evaluates a project for customer usability; that is, production quality in the context of the complete software application. For System Test signoff, the project also must pass a QA peer review. QA informs engineers and managers, including the documentation manager, of the test results by e-mail and also updates the project database.

The writers follow the progress of the testing. Projects that fail their tests may be delayed or canceled, so writers wait until tests are positive before beginning their own checking and writing.

E-mails from Engineers

Occasionally engineers send e-mails to writers or to the whole R&D department (which includes the writers), warning of changes or explaining new functionality. These e-mails may tie into projects already being investigated, or they may

alert the writers to new projects or to actions not assigned formal project numbers. Writers can then follow up by investigating other sources and by testing the latest software build.

Putting It Together

The writers must assimilate all the information from the sources described above and synthesize it into coherent descriptions of new and changed functionality for the *What's New* book (including examples) and topics in the *Online User's Guide* for each new release.

The writers check the Summary.txt database regularly, take note of new items, and test the descriptions against the current software build. They enter comments on their experiences and on reports of test results and e-mails from engineers in the database, and they investigate SPRs that are mentioned in Summary.txt.

The writers don't start writing for the upcoming release until the vice president of R&D states that engineering projects are frozen. At that point, no new functionality is to be introduced; however, it is inevitable that, as the software proceeds through alpha and beta testing and problems are fixed, some changes in user interface or functionality will occur. The writers must be flexible enough to adjust to any changes right up until the books go to press. After the books are printed, any changes are reflected in the Release Notes and in updated online help topics.

The first document generated at each new release is the *Getting Started* book. This book teaches the fundamentals and various important aspects of the software, but it does not introduce the latest functionality, so it can be printed while the software is being stabilized. Changes in the *Getting Started* book involve updating software user interface information, adding information on functionality from the previous *What's New* book, and occasionally modifying the book format.

As the writers begin writing the *What's New* book, they check the product definition list to be sure that they include all new functionality. They check the project database for the status of the projects. They consult the project specifications for terminology or functionality issues.

They test everything in the software one more time before submitting writing for review. Chapters are circulated to other writers for peer reviews, and then to engineers for technical reviews.

After the *What's New* book has gone to press, attention turns to the *Online User's Guide*. Online What's New topics for each subject area contain brief descriptions of new functionality with links to topics that have complete descriptions and procedures. The writers write or revise topics relating to new functionality, and finally they check all other help topics to be sure that they reflect current user interface and functionality as the new version of the software is released.

Keeping online help current is an ongoing process. Even after new releases, service packs are made available to customers every few weeks. Any changes in functionality are checked out by the writers and described in Release Notes and updated help topics.

Evolution of the Process

This description of the discovery process is a snapshot of its current state. The process continues to evolve. For example, at first the Summary.txt and SPR database were the sources of information. Over time, other information, such as project data and e-mails from engineers, has been made available to the writers. Handling of the Summary.txt information has evolved from individual perusals to database management. These changes have resulted from requests from the writers for more information and more involvement during the development process. With more information, the writers are able to test the software more intelligently and produce better documentation more quickly.


The deliverables have evolved as well. Originally, the printed materials included a large reference manual, while online help was minimal. Working with the software, the writers realized that online help was more useful than a printed reference manual, partly because access is easier and partly because the material can be more complete and accurate. Topics can be updated right up to the software release, and they can continue to be updated for service packs. In 2000, all

the information in the reference manual was transferred to online help topics, which became the *Online User's Guide*. Online tutorials and a design portfolio were added in 2000. In 2001, the help facility was converted from *WinHelp* into HTML format.

Plans for later this year include moving step-by-step examples from the *Getting Started* book to the online tutorials. The documentation group initiated this change to streamline the printed book process and enhance the online tutorials. The *Getting Started* book will be reshaped to provide more design information and to point users to corresponding online tutorials. The writers will reshape the step-by-step examples into tutorials by re-creating the models in the software and taking screen shots to illustrate the steps.

Recent initiatives from the writers include offering to review new messages, menu items, and other language in the software and asking to take part in planning meetings and demonstrations. Both these initiatives are in the beginning stages, but we expect them to deliver dividends in improved software, as well as improved documentation, as the writers work even more closely with the software and with the engineers. As the writers feel more ownership of the product, they are ever more committed to making it better.

Conclusion

Our writers become thoroughly familiar with the software they write about through the discovery process. As we have experienced, it takes time and effort for writers to be accepted as part of the development team, to be conversant with the software, and to produce the best documentation possible. The process may be more difficult than writing from specifications, but it is certainly more satisfying and results in better documentation. We encourage writers at other software companies to consider initiating a similar process. 

Marilyn Kloss is a technical writer at Solid-Works Corporation in Concord, Massachusetts, and a member of the Boston Chapter STC. She can be reached at mkloss@solidworks.com.