User documentation as an integral part of a
software product

B. Vatovec
Hermes SoftLab, Litijska 51, 1000 Ljubljana, Slovenia
E-Mail: Bogo.Vatovec@hermes.si

Abstract

This article stresses the importance of user documentation as an aid that helps users become
familiar with a software product and helps users learn how to use it. The article describes the
methods used and the effort put into making the documentation an integral part of a product at
Hermes SoftLab. Only through this approach is it possible to ensure high quality
documentation that can compete with the quality provided by world leaders in software
development, to support the overall quality of software products, and to increase the quality
and usability of products even more.

The article outlines a method for improving documentation by using techniques borrowed from software development.

What is user documentation?

User documentation consists of any written or graphic material presented to end
users of a product in any form such as printed or online material. In general, user
documentation includes the following:

- instructions on how to use the product;
- online help, product messages, tutorials, and advice;
- educational material such as training course manuals;
- marketing material describing the product such as white papers and
  product sheets;
- sections of technical documentation intended for the end user as reference material.

User documentation does not include detailed technical documents such as product
specifications or high and low level design. Such documents are aimed at developers and not
end users.
The importance of user documentation

The primary purposes of user documentation are to assist the user in becoming familiar with the product, to help the user learn to use the product, and later to serve as a reference providing additional advanced information.

Unfortunately, many companies still regard user documentation as something not really essential but rather a "necessary evil" since a product appears more attractive and more professional if it has at least a few manuals packaged with it.

The main reason behind this attitude is that developers and most managers come from an engineering background and so unconsciously favour the technical development of the product over its documentation. This attitude results in last minute, ad hoc documentation, most often without a real vision or purpose other than to simply have something written since any documentation is better than none. In the computer industry, useless documentation has become proverbial.

However, those who believe that documentation is not important are mistaken.

Two decades ago when computers started to become widely available and used, most software was used by computer gurus who had been to some degree "initiated" into the secrets of computers. For ordinary people, however, this knowledge was beyond comprehension and in any case unattainable. Therefore, usability or how easy it was to learn to use a product was of secondary importance.

Nowadays, almost everyone uses computers or has at least some contact with them. More and more people want or must learn how to use computer software, not for its own sake but to help them perform their everyday tasks. They do not really want to learn about the intricacies of the software; they just want to use it.

Although much effort has been put into making software easy to learn and use, due to its growing complexity people instead often find it increasingly difficult to use. Much software nowadays has numerous features, but most people use only a few of them. For example, Word for Windows 6.0 has around 350 features, but about 90% of its users only need it for writing simple letters and reports using maybe up to ten features, most of which were available in simple, "prehistoric" word processors.

What do you do when you want to learn what your newly bought software can do? You browse through the manual and play around with the menus and online help.

What do you do when you want to do perform a specific task using the software but do not know how? If you can, you ask your favourite guru, but he or she is often too busy. Your next source of information is the documentation: the manuals or the online help.
If the documentation users receive with a product has been written by engineers at the last minute, it usually contains a lot of technical information not really needed to use the product and hardly anything helpful regarding the immediate problem. Users consequently become frustrated, afraid to use the wider functionality of the software because they can not understand it, and ultimately hostile toward using the product at all.

Most successful companies such as Hewlett-Packard, Microsoft, and IBM invest much effort and money into making their documentation useful and easy to use. Such companies conduct numerous studies into how people learn to use a product, how they build their understanding of a product, and how they try to accomplish their tasks with a product. Furthermore, by giving users good documentation and carefully designed products, they demonstrate that they care for their customers and are trying to make their lives easier.

**Who should write user documentation?**

The common belief that anyone who has the time or can write a decent sentence can write user documentation is based on underestimating the importance of documentation. The result is poor quality documentation that is difficult to read and understand and in which necessary information is hard to find.

As soon as one begins preparing documentation in a controlled manner and trying to achieve high quality and usable documentation, he or she faces huge problems such as planning and designing the content, acquiring information from team leaders and engineers, structuring and streamlining this information so that users will get what they need efficiently, tracking the development process, and so on.

To produce high quality documentation, experts who specialize in this area must be found, technical writers who have great communication skills, knowledge about structuring and presenting information, and a clear understanding of the user's needs.

**Improving the situation in a company**

Technical writers and managers in companies facing the problems mentioned above might want to keep the following points in mind to help them improve the situation in their companies regarding user documentation:

a) Development of user documentation must become an integral part of the software development process. This is illustrated in the following sections.

b) Company management, team leaders, and engineers must be encouraged to understand the importance of documentation as an integral part of the product itself. This often requires a great deal of informal conversation and persuasion.

c) The development of documentation requires specific knowledge. Documentation should therefore be written by professional technical writers.
with such knowledge. In many countries, such people are hard to find, requiring either investment in education or the hiring of people from abroad.

d) User documentation must be included in a project at an early stage and must be included in the project plan. This can only be done if project managers understand the importance of documentation. An experienced technical writer can help considerably in making a product easier to use.

e) The development of documentation must go in hand with the development of the software, as explained in the next sections. This ensures high quality and the cooperation of team members.

These points are interrelated; for example, when the people responsible for developing software understand the importance of documentation, they will understand that they need technical writers to create the documentation.

By incorporating the documentation in the development process, a better quality product is also achieved. Writing the documentation requires testing how a product works and considering things that engineers often do not think about, for example, whether letters and numbers can be used together in fields, consistent terminology, layout and style, and so on. Furthermore, difficulties encountered in describing how something works frequently expose potential problems that users may have in trying to use a product.

Improving the development process

The basis of improvement lies in making the transition from ad hoc development to a planned and controlled approach using the project life cycle. The terminology and the process itself is borrowed from software development where it has been verified and improved through years of application. Some of the benefits of using the same project life cycle are the following:

a) The process has been proven to work through years of software development and is flexible enough to use in similar processes such as documentation development.

b) Easy integration of documentation into the development of the software since they both use same phases with approximately the same results and stages of development.

c) Software developers are familiar with and understand the software project life cycle; therefore, it is easy for them to understand the documentation life cycle.

d) Similarity makes it possible to use similar tracking tools and methods that have been used by the company to track and manage software development.

The transition to a well managed and organized process is not necessarily simple since it depends on existing company policies and structure. In organizations that already have well defined processes for other parts of product development and use them in everyday work, it is usually painless enough to
establish a new process for documentation development. Furthermore, in these organizations the definition of such a process would most likely be required in any case. On the other hand, in quickly growing companies that do not yet have well defined processes, the transition can become fairly painful. A well defined process for documentation development can only work if all the other processes are well defined and, even more importantly, are followed and used by all developers.

The development process of documentation is, as already mentioned, essentially very similar to the development process of the software itself, with some specific differences. The following typical phases appear in the life cycle of documentation development:

- Information planning (10%)
- Content specification (20%)
- Implementation (50-60%)
- Production (19%)
- Evaluation (1%)

**Description of the development phases**

This section briefly describes each of the development phases occurring in documentation development. For a more detailed description of phases, activities, and deliverables, please refer to the reference materials listed at the end of the article.

**1. Information planning (10%)**

**Activities:**
- develop a user profile in order to conduct a high level task analysis
- understand the plans for the development of the product
- establish usability goals
- participate in product developers' planning sessions and conduct a Phase 1 review

**Deliverables:**

- Project Plan:
  - estimates of scope, hours, required resources
  - preliminary milestone schedule
  - plan for editing, translating, production
  - acquire resources such as writers, editors, and so forth
  - acquire the tools needed

- Information Plan:
  - describe users
  - describe high level tasks
Software Quality Engineering

- describe the product
- describe the usability goals
- outline various design implications
- create a user/task matrix
- create a media plan

2. Content Specification (20%)

Revise the Project Plan:
- reestimate the project scope, hours, required resources
- revise milestone schedule as needed
- acquire additional resources as needed
- acquire additional tools as needed.

Activities:
- expand the user profile
- conduct a detailed task analysis
- learn details about the product and the interface
- write progress report status
- review the Content Specification as needed
- plan and conduct Phase 2 Reviews

Deliverables:
Content Specification:
- revise the user description
- describe the detailed tasks
- explain the organization of the publication
- create a detailed, annotated outline (Table of Contents)
- estimate the number of pages and graphics needed

3. Implementation (50%)

Activities:
- track hours, percentage of completed work, changes in milestones, changes in the product
- estimate the effects and changes
- revise schedule
- find additional time and resources as needed
- create prototype sections to conduct draft reviews and make changes
- coordinate draft reviews
- language editing

Deliverables:
- First draft (pre-Alpha or Alpha phase)
4. Production (19%)

Activities:
- work with translators
- monitor production activities
- prepare printing copy
- conduct final checks of camera-ready copy
- arrange for distribution
- monitor progress and solve usual problems

5. Evaluation (1%)

Activities:
- conduct the project wrap-up meeting
- obtain customer feedback
- evaluate the success of the life cycle
- plan for process improvements
- conduct team and individual evaluations
- establish project archive
- participate in the project evaluations

Deliverables:
- project wrap-up report to project archive

Quality inspection of documentation

A common problem that developers of software and its documentation face is how to check the quality of their product. One popular definition of quality states that the customer is the only reliable source of information about quality. As it is often impossible to get in touch with customers until a product is released, this implies that it is also impossible to check the quality before the final release of a product. At this stage, however, it is nearly impossible to make any changes without incurring extreme costs. It is essential therefore that we have a way to check quality earlier, that is, during development. Although a process with deliverables and milestones allows easier inspection of quality during development, it is impossible to ensure quality without regular reviews and inspections.

The following list itemizes the quality inspections needed for each phase:

**Phase 1:**
- conduct early tests of the design
- conduct tests of previous and competitive products and publications
Phase 2:
conduct early tests of the proposed structure of the documentation, terminology, text, and graphic designs

Phase 3:
conduct tests of early prototypes of instructional and reference texts;
conduct tests of drafts
test indexes

Documentation as a part of the ISO 9000 standard

At Hermes SoftLab we develop software products and accompanying user documentation for companies such as Hewlett-Packard, SoftLab, and Bull France. To guarantee the level of quality demanded by our client companies, we introduced working instructions as guidelines for technical writers developing documentation. These guidelines provided assistance in the development process, deliverables, page layout, and style usage.

When Hermes SoftLab began incorporating the ISO 9000 standard, these working instructions successfully matched the descriptions of processes required by the ISO 9000 standard and were used as such.

Conclusion

Incorporating the development of user documentation and user assistance as a component part of a software project's life cycle has been and continues to be an on-going process at Hermes SoftLab.

The results are evident: Company management and development engineers alike have a better understanding of the importance of documentation and of the development process itself. At Hermes SoftLab we not only develop software products but also the complete technical and user documentation necessary for its use. Hewlett-Packard and other companies have used our documentation as their own without modification.

Although we can claim that our documentation is at a high quality level, we are working hard to improve its quality even more by following and using the newest knowledge and technology.

Key Words

user documentation, software documentation, online help, usability, life cycle

References