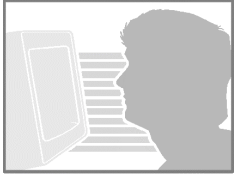


Usability Interface



A combined publication of the STC Usability and Quality SIGs

DocQment

Quality SIG Newsletter

October 2000
Volume 7, Issue 2



Society for Technical Communication

Inside this Issue

Usability and Quality: Partners in Achieving Effective Documentation	1
Are Organizations Doing Enough to Improve Customer Satisfaction?	3
How Technical Communicators Can Apply User-Centered Design	4
GUI Bloopers: How <i>Not</i> to Design Software	6
Results on a Study of Usability Testing	7
Quality, Usability, and the Ontological Argument for the Existence of God	8
SIGs: A Good Investment	9
Musing on Metrics: Why Measure Usability?	10
SIG News.	11

Usability and Quality: Partners in Achieving Effective Documentation

Ralph E. Robinson, Quality SIG Manager

Dick Miller, Usability SIG Assistant Manager

When we began discussing this joint issue of Usability Interface and DocQment, we quickly realized that no concise definitions for either usability or quality existed: in fact, there were many definitions for both terms.

A query to a usability professionals' mailing list for a definition produced many varied responses from which a brief working definition of usability can be distilled: *A usable product is one that is learnable, efficient, memorable, error-averse, and satisfying.* In other words, usability is a characteristic of a product or document. Defining quality is no easier: there are also a variety of answers. A brief working definition might be: *Quality documentation is documentation presented in an easy-to-use, straightforward manner with complete glossaries and indexes.*

What is quality documentation and what makes documentation usable?

The answer is not just a checklist of features to be included. There's much more to it than that.

Have you ever encountered a piece of documentation that was technically accurate, used language and grammar correctly, was easy-to-use, had a great glossary and index, but was written for a different audience than what it claimed to cater to? You know the type, the User Manual that was really a System Administrator's Guide! Did you find it very helpful? In your mind, was it a quality document?

Quality purists claim a quality document is one that complies with all the requirements set down by the customer for the documentation.

Continued on page 13.

Call for Articles



For publication in Usability Interface

Usability Interface is accepting original articles and case studies, anecdotes, cartoons, and book reviews for the following upcoming topics:

Usability of Online Help

Submission deadline: November 5, 2000

Using Style Guides for Usability

Submission deadline: February 3, 2001

Designing for the Web

Submission deadline: mid-May 2001

Hardware Usability

Submission deadline: early August 2001

Contact David Dick at david.dick@village.uunet.be for submission guidelines. •

For Publication in DocQment

DocQment is currently accepting articles on all facets of quality in technical communication. Review a book for “BookQueue,” show off your technical knowledge for “What's Hot,” or report on the daily grunt and grind of industry for “In the Trenches.” Don't know what to do with your collected quality information? Write an article for “Musing on Metrics.” Do you have something that doesn't fit into these ongoing topics? Share your successes with a feature article in DocQment. Contact Shelby Rosiak at rosiak@us.ibm.com for submission guidelines.

DocQment is currently looking for an *Acquisitions Manager*. Become more involved in your Quality SIG—become a member of the DocQment newsletter team. We are currently looking for an Acquisitions Manager. This position has the primary responsibility of soliciting and obtaining content for the newsletter. This individual solicits content from members of the Quality SIG, from other STC members, and from individuals and publications outside of the STC. Contact Shelby Rosiak at rosiak@us.ibm.com to volunteer for this or other newsletter positions. •

DocQment & Usability Interface

October 2000

Volume 7, Issue 2

DocQment and Usability Interface are quarterly publications of the STC Quality and Usability Special Interest Groups

Managers: Ralph E. Robinson (Quality SIG) and Whitney Quesenbery (Usability SIG)

Newsletter Editors: Shelby Rosiak and David Dick

Distribution Editor: Candace Cole
candace.cole@ncr.com

Masthead Design: Debra Gabriel

Assistant Editors: Judith Blostein, Susan Duncan, Vanessa Mosher, Aruna Panangipally, Alice Preston, Steven Weintraub.

Usability Interface Publication Policy

We encourage letters, articles, and other items for publication. Articles can consist of up to 1000 words. Acceptable formats: Pictures: jpg or gif. Text: Word, RTF or ASCII. Address correspondence to David Dick, Maastrichtlaan 19, Vilvoorde 1800, Belgium or to david.dick@village.uunet.be.

DocQment Publication Policy

We encourage letters, articles, and other items for publication. Articles can consist of up to 400 words. Acceptable formats: Pictures: jpg or gif. Text: Word, RTF or ASCII. Address correspondence to Shelby Rosiak atrosiak@us.ibm.com

Quality SIG Web Site: <http://stc.org/pics/quality/>

Quality SIG e-mail list: To subscribe, send an email to lyris@lists.stc.org and in the body of the message type: subscribe stcqsig-l <your name> (The character at the end of stcqsig-l is a lowercase 'l' and the <> characters are not part of your entry.) For example: subscribe stcqsig-l shelby rosiak

Usability SIG Web Site: <http://stc.org/pics/usability/>

Articles archive

<http://stc.org/pics/usability/newsletter/newsletter-archives.html>

Usability SIG e-mail list: To subscribe, send email to stcusesig_l-request@lists.stc.org
In the body of the mail message, type:
“subscribe stcusesig_l”

Reprints

Reprints from the newsletter are allowed if permission is obtained from the original author, credit is given to the author and the newsletter, and a copy of the reprint is sent to the newsletter editor.

STC's Mission

The mission of the Society for Technical Communication is to improve the quality and effectiveness of technical communication for audiences worldwide.

Postmaster Send address changes to Society for Technical Communication, 901 N. Stuart Street, Suite 904
Arlington, Virginia, USA, 22203-1822

Are Organizations Doing Enough to Improve Customer Satisfaction?



Editorial Response

No

By David Dick, Editor, Usability Interface

Time-to-market pressure can diminish product testing time and quality. The results are product recalls, shoddy merchandise, and apologies by CEOs about *poor quality*. The consequence is the loss of consumer confidence. Don't these companies realise that there's no compromise on quality? I'm sure that these companies are ISO 9000 certified or have a Total Quality Management (TQM) program, so what is the problem? Perhaps the problem is not with ISO 9000 or TQM but with the way it is used. For example:

- Documenting procedures, as ISO 9000 requires, does not improve quality. It does require that staff be trained on how to accomplish their tasks, strategies for evaluating quality and improving product design, and an environment that streamlines processes and stresses best practice. How often do you see it happen?
- Collecting metrics, as TQM requires, does not improve customer satisfaction unless the organization uses the data to improve the product and service. If used properly, TQM will result in improving product design, customer satisfaction, and reduce costs without compromising quality. How often do you see it happen?

What's the solution? You must become an educated consumer and user. Don't buy a product that is not durable, safe, reliable, easy to use, easy to maintain, and easy to learn. If you don't feel you got value for your money, return the product and ask for an exchange or refund.

Customer service should be helpful and friendly. If you're not satisfied, write a letter to the President or CEO of the company with your grievances.

Be a volunteer usability tester: if you are aware of poor product design, speak up and let your concerns be known. •

Yes

By Shelby Rosiak, Editor, DocQment

Customer satisfaction is a major focus for organizations. Quality tools such as ISO 9000 certification and Total Quality Management programs are an excellent way for companies to collect quantitative information on processes and customers. But these tools are not enough to ensure customer satisfaction. For this reason, companies are reaching out to consumers in unprecedented ways to gain direct customer feedback. Take some examples that I've encountered recently:

- While checking me out of a hotel, the desk employee asked if I'd filled out a comment card and encouraged me to provide my input.
- I noticed a toll-free number on a candy bar wrapper with the following words: "[Our company] values your questions or comments. Satisfaction guaranteed or we will replace this product."
- After I completed an online purchase, a pop-up screen invited me to rate my buying experience at a third-party web site.
- After renting a car, I received a call from a customer service representative in the corporate office to see how my renting experience had been.
- Our on-site cafeteria has an Intranet site with a web-based comment form.

But perhaps the most surprising incident was one I had with my telephone company. I called the telephone company at 3:00 in the afternoon to report static over my home phone line. The telephone company promptly committed to having my phone fixed by 9:00 that night, or I would receive a credit on my next bill.

Continued on page 13.



How Technical Communicators Can Apply User-Centered Design

Lori Fisher, STC Fellow, Silicon Valley Chapter

The user-centered design process applies to designing a piece of technical communication as well as designing a product. Placing the user at the center of the design and development process for information ensures that a usable piece of communication will be delivered to the customer. Technical communicators can apply each of the user-centered design (UCD) tasks to their own writing process and information development cycle.

Technical communicators can apply UCD tasks to their own writing process and information development cycle.

Knowing the Users

As technical communicators begin planning a piece of communication such as a manual or tutorial, they can use the UCD process to assist them in gathering information about their audience.

The requirements-gathering process for a product should also include requirements for information—how do customers want to get information about this product? Through online help? Through wizards or tutorials? Through written manuals? What kind of information have they used in the past that proved most useful and why? How much do customers already know about this subject? What is their typical level of experience with this kind of product?

All of these questions are appropriate for focus sessions, surveys, or other requirements-gathering activities that may take place as part of the overall UCD process. If the overall UCD process does not include questions on information, technical communicators need to do this research themselves, for example through one-on-one interviews with typical customers, focus sessions, or surveys.

Task Analysis

Closely related to understanding a user is understanding the tasks that they perform. Task analysis is key to the UCD process and invariably results in very valuable information for technical communicators. As technical communicators design their information deliverables, they should consider which tasks are done most often by the users, which are considered the most important or critical tasks, which are the most difficult, and so on. The output of the task analysis activity provides a good crosscheck for the design of information, since every task needs to be covered in some way by the technical communication deliverables. Technical communicators can participate in the UCD task analysis activities to ensure that they understand how a user expects to complete certain tasks, or other details that may impact the information design.

Setting Usability Goals

Just as the product itself will have usability goals, the information should also be measured against usability targets. The audience analysis and task analysis should have provided some clues about what factors influence this particular user set's perception of usability: not needing to look in a book? finding information in five minutes or less? installing in less than one hour? Technical communicators should consider setting goals such as time to complete a step in a tutorial, time to find the right section on a particular topic, or 100% success when using the index to find a topic. Looking at the competition can provide valuable insights into the level of achievement of these goals that would be required to surpass the competition in usability. UCD sessions where customers are asked to discuss the strengths and weaknesses of other products and other information can also help to set appropriate targets.

Continued on page 5.

How Technical Communicators can Apply User-Centered Design (continued)

Prototyping

To obtain early feedback on the usability of information, technical communicators need to do as much testing with actual users as possible using early mock-ups of drafts, paper prototypes of online help systems, or storyboards of tutorials. All of these early methods are considered low-fidelity prototyping. They can be very valuable in determining early reactions to your information design before too much resource is invested in the actual writing and coding of the deliverables. It's always distressing to find out later, for example, that users want their reference information in alphabetical format or their installation information in three steps instead of 13.

Such early reviews of designs are often called *design walkthroughs*. As more product and information becomes available, the prototypes become more complete and sophisticated, or high-fidelity, and prototype evaluations can begin to take on the more rigorous characteristics of usability testing.

Usability Testing

Usability testing of information is often most effective when done in conjunction with overall product usability testing, using scenarios that measure both product usability and user perception of information usability. However, if your product team does not plan to use scenarios in their usability tests that include using the information to complete the tasks, technical communicators can design usability scenarios to test some parts of the information design independent of the product. For example, a relatively simple usability test can measure the time it takes to find specific topics using an index or using an online search tool for an online help system. Other usability tests can ask users to read a certain product-task scenario and then determine where in the information (online, printed, tutorial) they would expect to find the information necessary to complete the task.

Retesting and Iterative Design

Once an initial usability test of the information has been completed, the process begins again! Technical communicators can use the information from initial testing to redesign information as necessary, complete various deliverables, and test again. Beta test programs associated with early product delivery can be an effective mechanism for trying out early iterations of the information deliverables.

Skills Needed by Technical Communicators to be Successful with UCD

Many of the skills required for the UCD activities described above are skills that, traditionally, technical communicators already possess. These include: audience analysis, task analysis and task-oriented writing, designing survey questions and customer scenarios for information use, and clear communication with customers and team members.

However, full participation in the user-centered design process requires some additional skills that technical communicators may need to develop:

- Market analysis
- Competitive analysis
- Observing users
- Interviewing users
- Interpreting data from user feedback sessions
- Creating prototypes and running prototype feedback sessions
- Capturing the design and the rationale for design decisions
- Communicating the design to others
- Making design trade-offs based on schedules, resources, cost, and the priorities of other designers.

As Technical Communicators develop more of these additional skills, they will become more effective in the overall UCD process, and will be able to expand the value they bring to the product development cycle. •



GUI Bloopers: How *Not* to Design Software

By Jeff Johnson

Why are so many software products, electronic appliances, and online services difficult to learn and frustrating to use?

Here's one reason: most are designed and developed with little or no guidance from people professionally trained in making products and services usable. As a user-interface consultant, I am often called in to review or test software and suggest improvements. The problems I find are usually the result of the software having been designed by people who may be professionals at software engineering and

programming, but are amateurs at user-interface design.

Consequently, the software contains design errors that make it difficult to learn and use. Many of the errors are extremely common and can easily be avoided.

GUI Bloopers are extremely common and can easily be avoided.

I began compiling and categorizing common user-interface design errors (bloopers) to teach developers at my client companies how to avoid them. My blooper collection quickly became large enough that I decided to publish it as a book. It has recently been published and it's called *GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers*. Think of it as a design-guidelines book in reverse: instead of starting with the design rules for making software usable, it describes common design errors that make software hard to learn and use, and describes how to prevent them.

Of course, programmers and inexperienced designers aren't the only ones whose bloopers harm the usability of products and services. Many software development managers commit bloopers that cause their organizations to produce difficult-to-use

products and services. Management bloopers are in many ways more important than GUI design errors, because they effect more software and are harder to diagnose and correct. I have identified seven kinds of bloopers: GUI-Component, Layout and Appearance, Textual, Interaction, Web, Responsiveness, and Management.

1. **GUI-Component bloopers** are erroneous decisions about how to use on-screen controls. One very common GUI-component blooper is confusing checkboxes with radio buttons. Checkboxes are for independent on/off settings, while radio buttons are for choosing one value from several possible values.
2. **Layout and Appearance bloopers** are errors in arranging and presenting information and controls. One such blooper is displaying windows in odd areas of the computer screen. For example, one company had a software product that sometimes displayed new windows off screen, where users could not see them.
3. **Textual bloopers** are poorly written text in software. The most common textual blooper is describing errors in terms only programmers understand. For example, one web-based scheduling application displays the following message if you try to schedule two events for the same time: "Error-00001: unique constraint (PA_REPORT_HEADERS_U1) violated." Instead, it would be better worded: "You can't schedule two events for the same time."
4. **Interaction bloopers** are violations of high-level design principles that make life difficult for software users. One such blooper is imposing arbitrary limits. How many person-hours have been wasted worldwide since 1980 by people who were trying to name documents and data-files in fewer than eight characters? Another common Interaction blooper is: "Cancel button doesn't cancel."

Continued on page 7.

GUI Bloopers (continued from page 6)

5. **Web bloopers** are problems that are specific to web sites and web-based applications. One common Web blooper is failing to mark links so that users can see whether they've already been there, done that. Another blooper that I often see is a web site that unintentionally reveals a company's internal turf battles.
6. **Responsiveness bloopers** are aspects of a product or service's design that interfere with users' work-pace. Most software doesn't provide sufficient feedback about what it is doing. There are others, too: buttons that don't respond to clicks in the required 0.1 second (so we click them again), scrollbars that can't keep up with the mouse, and lengthy operations that don't indicate progress or that can't be stopped before completion.
7. **Management bloopers** are management-level mistakes that affect the usability and usefulness of software and electronic appliances. Examples include: assigning important GUI components to novice programmers, entrusting the writing of error messages and other software text to programmers rather than to trained technical writers, and failing to reconcile differences in the design of different parts of the same program.

My hope is that *GUI Bloopers* will help the software industry realize that if it wants to reach a mass market, it must learn how to develop user-friendly products. •

*Jeff Johnson is President and Principal Consultant at UI Wizards, Inc. He has written numerous articles and books on a variety of topics in Human Computer Interaction and Technology Policy (see www.uiwizards.com). He is the author of *GUI Bloopers* (Morgan Kaufmann, March 2000).*

See "The Bookshelf" for a review of *GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers*

Results on a Study of Usability Testing

By Elaine Ostrander

A study I conducted as part of my graduate work at the University of Houston shows that technical communicators find many benefits in usability testing of documentation but cannot quantify them. The study's purpose was to identify an exact return on investment *figure* that could be used to convince otherwise unwilling management to initiate usability studies. While the data failed to produce such a figure, impact analysis indicated that the return on investment *is* probably high.

The study focused only on informal, "over-the-shoulder" (rather than laboratory) usability testing of *documentation*, not of software or products. Participants reported the amount of time and hourly costs spent on usability testing. Participants also choose benefits of usability testing they had experienced from a list of possible benefits.

Participants reported costs with a high degree of confidence and produced a qualitative list of benefits, but could not quantify those benefits. However, impact analysis demonstrates that the quantitative benefits may be large, and shows considerable promise for a future study in which participants are trained to track and tabulate the benefits with greater accuracy. There is a strong indication that the return on investment for performing informal usability testing of documentation is significant.

Further study on this topic will need to include training for participants on how to track and tabulate the value added by usability testing of their documentation.

If you would like to participate in such a study, either as a participant or as a trainer, please contact me via email at eostrander@lgc.com or by mail: 18102 Bambridge, Houston, Texas 77090.

For the complete version of the study, see <http://stc.org/pics/usability/newsletter/newsletter-archives.html>.•



Quality, Usability, and the Ontological Argument for the Existence of God

By Craig Marion

I used to study philosophy. In a class on the philosophy of religion, we studied several arguments for the existence of God. One that I could never get into was proposed by St. Anselm of Canterbury in 1078 and became known as the ontological argument. My professor set it to the tune of *Waltzing Matilda*. One of the verses, as I recall, went like this:

If that than which nothing greater than can be
conceived
Can be conceived not to exist
Then 'tis not that than which nothing greater than can
be conceived
This is unquestionable, I insist.

Which is to say, Anselm defined God as the greatest being that could be conceived. From this he reasoned that if you had another being that was like God in every way but lacked the attribute of existence, then you could conceive of an identical being who did have the attribute of existence. The second being would be greater than the first. Ergo, God exists. Q.E.D.

I was happy to find that Thomas Aquinas skipped this particular argument in his *Summa Theologica* a century later. In subsequent centuries, though, it may have been ignored, but it wasn't refuted. And then, in 1787, Immanuel Kant drove a stake through its heart. Insisting that being was not an attribute of a concept at all, Kant pointed out that if you had a real sauerbraten and an imaginary sauerbraten, and they both looked and smelled and tasted the same, then there was absolutely no conceptual difference between them. The fact that one existed and the other didn't was irrelevant. Saying that something exists or not, doesn't tell us anything more about it conceptually.

After the exam, I tucked this esoteric knowledge away until the early Nineties, when the company that employed me became enamored of the quality movement and reengineering. We spent the better part of a year delineating our processes, scrutinizing

them, and seeing where we could improve them. Then we would reengineer those that could be improved. For some reason, the ontological argument popped into my head again. I drew the analogy between processes having "quality" and being having "existence" for one of my co-workers, and asked what the attribute of "quality" told us about processes. She grunted, as I recall, and suggested that we get back to work.

Well, now it's the first decade of the new millenium, and I've been involved with the usability movement for about five years. It's occurred to me more than once: what's the difference between "usable" software and just plain "software that works the way it's supposed to"?

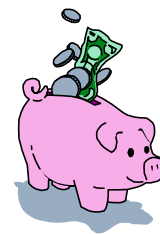
What's the
difference between
"usable" software
and just plain
"software that
works the way it
supposed to"?

The answer—for both quality and usability, I believe—hinges on the meaning of "works the way it's supposed to." Once, processes just evolved. They were changed sometimes, but they were never systematically evaluated. Having standards applied made them quality processes.

In a slightly different sense, software has always been evaluated, but only to satisfy internal QA departments. Recently the meaning of "works the way it's supposed to" has been redefined. It isn't enough for software not to crash; now it has to satisfy its users. Only software that does is usable software.

Still, there's something to the analogy. Usability, per se, isn't an attribute, just as quality isn't. It's a way of looking at something and determining whether it fulfills its purpose.

Continued on page 13.



SIGs: A Good Investment

George F. Hayhoe

In a recent discussion on TECHWR-L, an e-mail discussion list for technical communicators (not affiliated with STC) several posters debated whether STC SIGs are worth the \$5 per SIG assessment the Society instituted a few years ago for SIG membership. Some writers noted that they didn't see any value returned from the SIGs that they belonged to, while others believed that their experience had been quite different.

I belong to seven STC SIGs. Admittedly, some provide more value for the dollar than others (at least in terms of tangible benefits such as newsletters and listserv activity), but all of them have been well

worth the investment. I'd like to offer a few observations based on my experience as a SIG member as well as four years on the STC's board of directors and two years on the SIG advisory committee in the early 90s.

Is membership to an STC SIG worth \$5?

SIG Dues Aren't Paid to Volunteers

As with chapters, each SIG's level of activity is entirely a result of volunteer effort. None of the \$5 SIG dues is paid to volunteers, who donate their time. The larger, more enthusiastic, and giving the membership, the more bang the members get from the money. The most tangible SIG products are their newsletters. Paper, production, printing, and postage are incredibly expensive, and those costs continue to rise. As with most professional organizations, the cost of delivering SIG services exceeds the fee the organization charges its members. The SIG membership surcharge recovers only a portion of the cost per member for the services provided.

STC's board of directors debated charging for SIG membership for at least four to five years before instituting the additional fee a few years ago. The argument for SIG dues was to shift part of the expense of SIGs directly to those who use the services. The board delayed implementing the surcharge for several years to allow the SIGs to grow sufficiently so that the additional fee wouldn't have an adverse effect on membership. Despite the additional cost, SIG membership has grown each year, not declined.

How SIGs Add significant value

The thoughtful and informative Annual Conference sessions offered by many SIGs (amounting essentially to SIG stems in a couple of cases) that have added significant value to the conference, which was an extraordinary value to begin with.

The Information Design (ID) SIG became the first SIG to sponsor a section in an issue of STC's journal, *Technical Communication*. The May 2000 issue featured an introduction, three commentaries, a bibliography, and three articles on information design.

Would I prefer that there was no additional fee for STC SIG membership? Of course I would, but I know that it isn't a realistic expectation, and I'm quite satisfied with the return on investment in my seven SIGs. If I belonged to any SIG that was not returning adequate value for my financial investment, I'd have two alternatives: I could either volunteer to help solve the problem by taking an active role in the SIG or "vote with my feet" and not renew my SIG membership next year. •

George F. Hayhoe is a North Carolina STC Chapter Fellow and the editor of Technical Communication. This article was reprinted with permission of News and Views.



Musing on Metrics: Why Measure Usability?

By Steve Jong

ISO defines usability as “a measure of the effectiveness, efficiency, and satisfaction with which specified users can achieve goals in a particular environment.” To those of us who are interested in documentation quality metrics, this definition is marvellous. We quibble over what typeface maximizes readability, and to what extent readability affects quality, but there’s no arguing with the value of reducing time to task completion. Usability testing closes the loop in the product development cycle; without it, development can spiral off in the wrong direction. Upcoming revisions to the ISO quality standards not only incorporate usability as a quality requirement but also mandate its measurement. To me, usability is an excellent working definition of product quality, and the field of usability is rich in metrics. The briefest web search yields dozens of metrics, such as time to completion of a task, percentage of tasks completed without errors, number of commands or keystrokes or mouse clicks used, number and type of errors per task, percentage of satisfied users and their degree of satisfaction. Many are critical success factors. In my view, usability metrics are concrete, meaningful, objective, causative, and clearly useful.

Usability metrics are evaluative because they describe the results of a development effort. Gathering usability data can be expensive; gathering data on all of a company’s products (or documentation) simply isn’t practical. The question “is this product usable?” leads to the general quality question: “what makes your product usable, and how can I make mine usable too?” Usability professionals offer heuristic rules—suggestions that, when followed, lead to more usable products. We have heuristics of our own. I believe measuring adherence to these rules is a cost-effective way to predict the usability of products or their documentation.

For example, how many steps are there per procedure in your document, and are they all at about the same level of detail? A procedure with 70 steps (yes, I’ve seen one!) takes longer to complete than one with seven. Obviously, the usability of your document is intertwined with the usability of the product it describes; you’re well advised to tell your developers that a 70-step procedure is unusable. Did you adhere to a process of technical review and editing, so that command, argument, window, field, menu, and button names are accurate? Did QA verify your procedures? If so, you can be confident your document’s not causing user errors.

Usability experts dispute the interpretation of their metrics. They wonder if reducing the task time by 12% means a 12% improvement in productivity or a 12% increase in stress. Which carries more weight: time to task completion or user satisfaction? I recognize this yearning for a single usability score. The issue is whether the distribution of values for a given metric is normal. If not, comparisons have little meaning.

The usability literature also speaks of establishing goals, but I would caution against setting them arbitrarily (say, “the user shall complete the task within thirty seconds”). As Dr. Deming said, the data is what it is. A better approach would be to establish the usability of the system by taking measurements, plotting run charts, determining control limits, and then looking for the root causes of variation. Reducing variation, or continual improvement, is the great promise of Deming’s methods.

If you believe that everyone stumbles in the same place or for the same reason, then studying a few users is sufficient to form a reliable opinion of the product’s usability. Otherwise, I think a usability study, like any other statistical effort, requires a randomly selected, statistically significant sample to be valid.●

SIGNificant News

By Ralph Robinson, Quality SIG Manager

Summer is now over and the annual STC Conference in Orlando is just a memory now. I hope all of you had a wonderful and relaxing summer vacation.

Speaking of the STC Conference, Quality SIG members were very active in making presentations and taking part in panel discussions. Steven Jong, Robbie Rupel, Amy Perry, Carolyn Watt, Karen Schriver, Lori Fisher, and Ralph Robinson (sorry if I missed any others) played an active part in promoting document quality by sharing their experiences with the attendees. From the audience critiques that I have seen, all their presentations were well received and rated highly by those who attended.

New Fellows

Two of our members deserve special mention for their lifetime achievements as members of the STC: Karen Schriver, on being named an Associate Fellow of STC, and our past SIG Manager, Lori Fisher, on her appointment as a Fellow of the STC. Congratulations gals and thanks for your dedication and commitment to the profession of Technical Communication.

About the Joint Issue

The issue you have in your hands is something unique in the history of STC SIGs: a newsletter produced as a joint effort of two SIGs. Credit for the original brain wave goes to Whitney Quesenbery, Manager of the Usability SIG, who made the suggestion in early January of this year.

Congratulations go to the two newsletter teams headed by Shelby Rosiak, editor of our very own DocQment, and David Dick, editor of Usability Interface. They have done an outstanding job and I hope you like the end result

Get Ready for Chicago

Plans are already in the works for the next STC Annual Conference in Chicago, May 13-16. The Quality SIG will host a panel discussion on Quality issues. I'm sure that many members have submitted proposals for individual efforts. We'd like to help promote your efforts.

If you have a proposal accepted, please contact ralph.robinson@honeywell.com and give me the details.●

Usability SIG News

By Whitney Quesenbery, SIG Manager

It should be a good year for usability at the 2001 Conference, schedule for May 13-16, in Chicago. We've been peeking over the shoulders of the Theory and Research Stem Manager, and it looks like many good proposals were submitted focusing on usability topics, including case studies, workshops on techniques, and some overviews of usability processes. As usual, we'll be posting a list of sessions on the web site as soon as they are announced. If you have a session accepted, let us know, so we can help promote all usability-related events at the Conference.

Looking Ahead to Nashville

It's a year away, but when the Call for Proposals for the 2002 Conference in Nashville goes out, it will include a new lineup of stems. Assistant to the President for Conferences, Deborah Sauer, and Program Manager, Charles Fisher, announced a new "Information Design and Usability" stem. It's just another sign of the ever-increasing interest in usability. Let's thank them for the vote of confidence by submitting the best group of usability sessions ever. Start planning now—the deadline for proposals, August 1, 2001, is only ten months away.

P.S.—I have volunteered to be the Manager for the new Information Design and Usability stem.●



BookQueue

Review submitted by David Dick

Johnson, Jeff, *GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers*. Morgan Kaufmann, San Francisco, CA, 2000.

GUI Bloopers is about the outrageous and sometimes frustrating designs that have become common to most software. Johnson calls them “Bloopers” and he has identified 82 of them. Don’t be surprised if you find yourself thinking, “yes, I’ve seen that one before.”

Jeff reasons that software is developed by programmers who lack training in designing user interfaces or who don’t have access to people with such training. Even when software is developed by a sizeable organization, there may be no developers who have user-centered design expertise. Actually, many developers have user interface expertise in terms of the widgets and windows they construct—they just don’t understand the effect of their designs on users.

What’s the solution? If time and budget allows, a user interface consultant can be called in to review or test software. Although consultants can make a good living correcting common mistakes, I sense that Johnson believes the realistic solution is to educate software developers since there are so many of them and so few user interface design experts.

GUI Bloopers is organized as follows: First Principles, GUI Component Bloopers, Layout and Appearance Bloopers, Textual Bloopers, Interaction Bloopers, Web Bloopers, Responsiveness Bloopers, Management Bloopers, Software Reviews, and War Stories (which describes Johnson’s experiences as a user interface consultant). Each chapter presents bloopers, possible fixes, and design rules to avoid making the bloopers. I was particularly interested in the chapters about textual and web bloopers.

The chapter about Textual Bloopers stresses the importance of professional writing (for example, in labels and warning messages) for creating usable and elegant products. Ironic as it is, the so-called graphic user interface also includes text. It makes a strong argument for including technical writers on design teams. Technical writers can prevent textual bloopers by ensuring consistent terminology, correcting grammar and spelling, simplifying and clarifying error messages, and ensuring consistency of window titles.

The chapter about Web Bloopers has suggestions about web page design. The web, and the desire by everyone to create a web site, has turned traditional non-user interface designers into web page aficionados. You won’t be surprised (or maybe you will be) to learn that developers of web sites make the same user interface mistakes which affect the usability of a web site.

GUI Bloopers is a wonderful blend of practical guidance, case studies, and humorous illustrations. Unfortunately, the lack of figure titles is a cause for confusion.

Jeff Johnson is not a newcomer at sharing his knowledge and experience of user interface design. The book credits him with nine papers published by CHI, ACM, and IEEE. *GUI Bloopers* is his first book and heralds his talents as an author. I look forward to reading the sequel. ●

Famous Quotes

We could design the product with a simple point and click interface. Or we could require the user to choose among thousands of poorly documented commands, each of which must be typed right on the first try. Bear in mind, we'll never meet a customer ourselves.

Dilbert

Usability and Quality (continued)

We have seen some of those documents, and they were far from easy to use, sometimes were technically inaccurate, and the proper use of the English language never entered into the picture!

We've all encountered widely used products that are not usable. TV remote controls and VCRs come immediately to mind. In the world of documentation, many of us have seen documents with excellent graphics, carefully planned page or screen layouts, and expensive production aspects. However, they can be difficult to use because the information is hard to access, hard to understand, or irrelevant to the supposed purpose of the document.

A good definition of quality documentation can be summed up in the following phrase: *Fitness for the use it was intended*. As professional technical communicators, we are expected to be excellent users of the English language, so that factor should never enter the equation. Usability characteristics such as the ability to navigate through the document and find what you need, getting accurate information when you find it, and having the information that assists you in completing your tasks are all characteristics of documentation that is *fit for use*. It goes without saying that meeting all the requirements of the customer who contracted the writing will be critical to your long-term success as a writer.

We agree that quality and usability are closely related terms that attempt to focus attention on aspects of a documentation product that cannot easily be thoroughly determined by inspection or heuristics. In both cases "the proof of the pudding is in the eating," to borrow an old cliché. That is to say, the quality and usability of a documentation product are best determined when the people for whom the product is intended use it in the context for which it is intended. Only then can the questions of the product's quality and usability be answered.

This leads us back to this combined issue of the newsletter for the Quality and Usability SIGs. We hope that it serves as a way for our Quality SIG and Usability SIG members to gain better understanding of each other's perspectives. Our guess is that we'll find that we share more commonalities than we have differences. ●

Are Organizations Doing Enough? (continued from page 3)

I was still skeptical when I arrived home and found a note on my door stating that the repair had been completed at 5:15 p.m. Later that evening I received a call back from the telephone company asking if the repair had been completed to my satisfaction. It had—the static was gone.

Companies are improving customer satisfaction. We must remember that we are usability testers for virtually every product or service that we use, and our input directly impacts quality. The next time you are satisfied or dissatisfied with a product, pick up the phone, write a letter, or send an email to the company and give them your feedback. But be quick—they may call you first! ●

Quality, Usability, and the Ontological Argument for the Existence of God (continued from page 8)

The reason the quality movement became prominent in the early Nineties is that standards began to be applied. The reason the usability movement became prominent in the late Nineties is that standards changed.

In one sense, the adjectives "usable" and "quality" don't tell us any more about processes or software than "existence" does about a concept. Rather, they tell us something about the expectations they're measured against.

I haven't been following the challenges facing the quality movement today, but one of the largest and most important ones for the usability movement is to clarify expectations. And while Anselm may have regarded his argument as unquestionable, no one is likely to say that about usability standards in the near future. ●

Usability Interface & DocQment (Special Issue)
716 S. Riverside Drive
Shark River Hills, New Jersey 07753
U.S.A.

Upcoming Conferences



Nov 11-13, 2000—Doors of Perception

Amsterdam, the Netherlands. See www.doorsofperception.com/

Nov 16-17, 2000—ACM Conference on Universal Usability: Solutions, Systems, and Methods, Co-Sponsored by STC

Washington, DC. See <http://www.acm.org/sigchi/cuu/>



**Nov 28-Dec 1, 2000—Participatory Design Conference
Designing digital environments - Bringing in more voices**

New York, NY. See www.cpsr.org/conferences/pdc2000/

**Dec. 2-6, 2000—CSCW2000—Computer Supported Cooperative Work
sponsored by ACM SIGGROUP and SIGCHI**

Philadelphia, PA. See www.acm.org/cscw2000/