continued from p. 144

Hamilton gives an example of a situation in which he built power and influence. His firm issued annual updates of a complex product. The development groups were responsible for writing parts of a document describing their ongoing changes for the benefit of the support organization. When the writers needed clarification of the change document so that they could write clear release notes at the end of the year, the developers didn’t want to provide information they thought they had already given about features they had finished with months earlier. Hamilton’s team took over preparing the ongoing change document, combined it with the release notes, and created an intranet site to simplify the developers’ task of providing the new information and reviewing what the writers did with it. Hamilton’s team thus gained control of the inputs they needed and won respect and gratitude for simplifying the developers’ jobs and providing the support organization with clearer information.

Hiring and managing writers

Hamilton begins his book with a short summary of what technical writers do and the challenges they face. In a few pages he lists the key elements of technical writing: product, developers, audience, tasks, deliverables, environment, and schedules. Many writers focus on a few of these elements but ignore or skimp on others. Hamilton explains why each element is important and touches on the key issues associated with them.

Hamilton focuses on the three main aspects of managing: people, projects, and technology. These discussions become successively more abstract, though none of them strays far from concrete real-world situations. The discussion of managing people, however, seems close to Hamilton’s heart. The examples are gripping; they deal with situations that should resonate at an emotional level with most readers.

Hamilton’s longest chapter by far is on hiring, which shows the importance he attaches to that key activity. I have reviewed a number of books that talk about résumés. They present different, often contradictory, viewpoints. The message seems to be that there is more than one way to skin a cat. Hamilton details his process for evaluating résumés, and you won’t go far wrong if you use it. He focuses on general themes rather than details, and he shows a laudable level of flexibility. For example, he says he would hire a writer with only Windows experience to work on a Linux project, but he’d be reluctant to hire a writer who specializes in software to write an airplane hardware maintenance manual.

Hamilton makes a good point: the qualities that make a great technical writer boil down to the ability to understand and the ability to communicate. This seems obvious, but if you look at job ads, they often focus on tools, technology, and prior experience.

Hamilton is aware of the legal complications of hiring, managing, and occasionally firing employees. He contrasts the situation nostalgically with his father’s situation as a personnel manager from the 1940s to the 1970s. He suggests making friends with people in the human relations department long before issues arise. This is the same advice Hamilton gives about building relationships with developers and with other managers. The pattern seems to be that good relationships lubricate formal processes. This is a tough lesson for introverts, a category that many technical writers inhabit. Many introverts would rather deal with processes than people.

Managing change

One of the most difficult problems of managing people concerns change. Most people adhere to the maxim, “If it ain’t broke, don’t fix it.” It usually takes real pain to make people want to change. For example, if your company must translate your documentation into other languages, the potential costs are painful to the company’s finances.

Hamilton uses former General Electric chairman Jack Welch’s metaphor of a burning deep sea platform. The managers standing on the burning translation costs platform feel the pain and are ready to act, but the writers may see it as someone else’s problem as they watch from shore. The people who must make the change work must first understand and feel the need to do so.

Another common problem with managing change is to confuse the means with the ends. Hamilton tells us to define the desired future state in terms that impose minimal constraints on the means you use to get there. In particular, that means keeping solution vendors—and consultants who are tied to specific vendors—out of the process until you have a clear idea of where you are going.

Managing technology

New technologies present technical as well as personnel problems. Hamilton provides good advice about managing technology. For example, not every problem can be solved by technology. If you have badly structured content, don’t count on a content management system to fix it. Figure out what you are trying to do and what obstacles are causing your pain. Articulate the desired final state. Then look for a solution, which might or might not involve new technology.

Once you have a solution, Hamilton suggests easing the personnel problems associated with the pain of adopting the change by phasing it in, providing good training to everybody, and keeping management in the loop—especially when problems arise. It is better for them to hear about problems from you before they hear from your critics. One aspect of phasing in change is to start with people who have a positive attitude toward the change. Get the bugs out before bringing the troglodytes along.

Hamilton is a specialist in using XML technology, but he is also aware of the hype associated with it. He quotes Richard Feynman: “For a successful technology, reality must take...
precedence over public relations." The reality in this case is that it "will not organize your documentation, eliminate your production back-end, nor allow you to hire fewer, less skilled, or cheaper writers." XML is a system for producing markup languages (for example, DocBook or Darwin Information Typing Architecture [DITA]). Such languages, when they integrate well with your documentation’s structure, help make your content easy to search, manipulate, reuse, and repurpose. Hamilton provides useful guidance in addressing these issues. There is no magic. You might have to make substantial investments before you begin to realize any of the benefits.

Many technical writing managers must deal with complex problems such as single sourcing and localization. For such problems, the solution often includes some sort of content management system. Hamilton explores the issues involved in setting up processes to support single sourcing and localization. He also writes about how to decide what sort of technological help you need in managing content.

While many vendors cite reuse as one of the main purposes of a content management system, Hamilton advises you to start by organizing your content in such a way as to minimize reuse. He considers it better to have just one logical place to look for a piece of content than it is to have it in different places, even if you have a system for synchronizing the content at those different places.

Resisting the dark side

One reason I am so enthusiastic about this book is that Hamilton is not in awe of sacred cows. Two of them are performance evaluation and performance metrics.

Business success depends on clear expectations between managers and their employees. They need a framework for ongoing discussion of goals, objectives, and other workplace issues. Weekly one-on-one meetings often provide this framework. On top of that, however, most companies use an annual performance evaluation process. This could be a beneficial way to focus on long-term issues, but most companies use it for a different purpose: sorting employees into winners and losers. This robs the process of many good effects it might theoretically have had. Neither managers nor employees can engage openly and honestly in the process, because they are aware of its negative consequences.

Nonetheless, Hamilton recognizes that almost all companies have a performance evaluation process. Eisenhower said something to the effect that plans are useless but planning is indispensable. Hamilton applies that model to performance evaluation. He does what he can to make the process of performance evaluation helpful. He suggests ways to do as little harm as possible with the resulting documents and the ensuing discussions with employees.

Hamilton devotes approximately a quarter of his book to managing documentation projects. He makes points I haven’t seen in substantially longer works on the subject, but I especially like his discussion of metrics. There is a cliché to the effect that if you don’t measure something you can’t manage it. Furthermore, keeping good records of predicted and actual values of some metrics helps to improve planning and estimating. Despite their benefits, however, metrics have downsides. Writers know that management might use them to evaluate performance, so the measured aspects improve to the detriment of the unmeasured aspects. Unfortunately, the easiest aspects to measure are not likely to be the most important. For example, it is easy to count pages but hard to measure quality.

As with performance evaluations, Hamilton shows how to make the best of a bad tool. He delegates measurement to individual writers to use as they see fit, and he solemnly promises not to use those measurements to reward or punish writers—or report the measurements to anyone likely to use them in that way.

Hamilton has much more to say about managing people, projects, and technology. This short book is full of useful information and hard-earned opinions. Whether you manage writers or would rather just sit in your cubicle and write, you need to read this book. It is the most practical and realistic book I have seen on the subject, and I recommend it highly.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

IEEE Intelligent Systems delivers the latest peer-reviewed research on all aspects of artificial intelligence, focusing on practical, fielded applications. Contributors include leading experts in

- Intelligent Agents
- The Semantic Web
- Natural Language Processing
- Robotics
- Machine Learning

Visit us on the Web at www.computer.org/intelligent
Before and during the dot-com boom, technical writing jobs in the US were plentiful and well paid. The ability to write coherent English sentences and use packaged software was enough of a qualification. Training in technical communication was not required, but certificate programs blossomed, and many new college graduates entered those programs. With the dot-com bust and the availability of high-speed communication between the US and India, many technical writing jobs became commodities and rapidly moved offshore. The number of people in the US identifying themselves as technical writers suddenly exceeded the available work. That number is now shrinking sharply to fit the demand. Many certificate programs have closed for lack of students.

Long before the dot-com boom and bust, however, a small cadre of academicians and reflective practitioners established the underpinnings of a profession of technical communication. They showed how to determine the needs of their audiences and studied the most effective ways to satisfy those needs. Some learned how to quantify the value that their work added to the products they documented. A few even learned the language of the business executives who determine the value that their organizations place on documentation. These academicians and practitioners mostly survived, and they form the core of the technical communication workforce that exists today. Many, including me, are members of the Society for Technical Communication (STC), the leading professional organization for technical writers, as well as the IEEE Professional Communication Society and ACM’s SIGDOC. A variety of online communities and mailing lists also support the field.

To be a successful technical writer today, you must understand the technical and business aspects of the work you do and how it affects the organizations that pay for that work. The book I look at this issue covers all of the bases you need to touch.


Richard Hamilton began his career as a software developer in an organization that took software development seriously. His organization moved him directly from leading software teams to managing documentation, because they had begun to take documentation seriously as well. In the nearly 20 years since his first documentation management job, Hamilton has learned a lot about technical communication and how it fits into organizations like AT&T, Novell, and Hewlett-Packard. He has become an expert at applying XML technology to documentation projects and serves as a member of the OASIS DocBook Technical Committee. He is a member of STC.

Hamilton approaches the subject from a manager’s viewpoint, and he carefully covers everything that a technical writing manager should know. But writers who are not managers need this information as well. Technical writers in today’s environment must understand the product, project, technology, and business issues. They must be comfortable communicating with their organizations’ top leaders, or those leaders will likely regard them as commodities and their work as a cost that comes directly from the bottom line.

On the first page of his book, describing the challenges facing technical writing managers, Hamilton writes:

Most important, technical documentation is viewed with disdain by many engineers and lives at the bottom of the power hierarchy in most companies. A significant amount of your time as a documentation manager will be spent working to gain respect, power, and leverage so you can do your job.

This sad fact applies to individual contributors as well. They must earn respect for themselves as members of the technical staff and of their project teams.

continued on p. 142