

Tech writers as sales reps? Interface Software's award-winning docs boost brand, revenues, and customer satisfaction

March 8, 2005

by Gordon Graham, *SoftwareCEO*

CRM vendor Interface Software seems to have done everything right.

The company has grown so fast it's made Deloitte & Touche's Technology Fast 50 for Chicagoland five years in a row, with a cumulative growth rate of 357 percent. It also hit Deloitte's Technology Fast 500 and the Inc. 500 list several times since 2000.

This high-profile growth helped Interface emerge as the world leader in their niche of CRM for professional service firms.

And in December 2004, the company was acquired by legal information provider LexisNexis for an undisclosed sum.

Ask anyone at the company, and they'll tell you all this success flows from their ongoing quest for total customer satisfaction.

Lots of companies say that, but at Interface they walk the walk.

For example, the company picked up a 2004 Market Leader award from CRM magazine last fall when it placed near the top of the pack in customer satisfaction.

On a scale from 1 to 5, Interface rated 4.5 — ahead of 38 other vendors, including Microsoft, Oracle, PeopleSoft, and Siebel.

"In an industry dominated by stories of CRM failure and user dissatisfaction, we are proud to be recognized as a

vendor that towers over the competition in customer satisfaction," says Interface CEO Nathan Fineberg.

Defining an award-winning documentation team

The quest for customer satisfaction has soaked deep into the fabric of the company. It certainly defines their award-winning technical writing team.

When we spoke with Matt Thompson, director of documentation and training with Interface Software, we uncovered the following 14 best practices for documentation at his firm — plus seven bonus tips on how to hire and motivate good technical writers.

Thompson came to Interface in 1997 after working in several other firms as a technical writer and trainer. At that point, Interface Software had no training department and just one other technical writer.

"That year several members of our executive team were brought on, including the CEO who really drove the culture of total customer satisfaction," Thompson says.

Fueled by that drive, Thompson was able to hire more writers and set up a training team.

"Documentation and training make a strong contribution to our customer satisfaction and profitability, and that's what makes any company appealing," he says.

Best practice in documentation #1: Create a healthy workplace.

Ask most technical writers about their workplaces and you'll likely hear the same complaints:

- The developers don't follow any specs, and never tell me what they're up to.
- They get to slip their schedules by months, but the minute they're done I'm supposed to have the docs ready, like by magic.
- They never let me near any live customers.
- I report to somebody who has no clue what I do. Until the next reorg, anyway.
- I'd switch jobs, but everywhere I've worked is just as bad.

These "worst practices" create a toxic environment for a documentation team.

For his part, Thompson is happy he doesn't face these perennial headaches at Interface.

"It's pretty close to Utopia for writers here," he says.

"And that helps us with recruiting. We've never had a hard time finding people who would like to work here. An environment where they can make a difference, do excellent work, and be respected is a big motivator."

**Best practice in documentation #2:
Understand the value of good documentation.**

Documentation teams often struggle to prove their ROI in the absence of any established metrics for measuring it.

"Many technical communicators express ROI through a reduction in technical support calls," according to a June 2004 article by SAS documentation specialist Ron Statt in *Intercom*, the magazine of the 18,000-member Society for Technical Communication (STC).

Enhanced usability can count as ROI because any improvements lead to more satisfied customers and help bring in new customers, he adds.

Another approach is to "identify the costs associated with not employing their services." If a developer has to write his own documentation, it will likely take longer, affect the release schedule, cost more (programmers generally earn more than writers), and ultimately not turn out as well, writes Statt.

Fortunately for Thompson, he's never had to justify his existence at Interface.

"Early on, our executives identified that having high-quality documentation made a big impact on customer satisfaction," says Thompson.

"They could see how it increases usage, gives more successful implementations, and creates better references and happier customers, which then drive more sales."

**Best practices in documentation #3:
Use documentation to gain an edge.**

Clever software firms turn the money they sink into documentation into a selling point. In any competitive shootout, good docs are bound to help.

"It was a competitive advantage to show that we had significant documentation and our competitors didn't, especially when we were first starting," Thompson says.

"It shows prospects that they're not going to be left on their own to figure things out.

"If it was appropriate for a particular sales call, our sales reps would bring the documentation with them, and it showed very well.

"Early on, that made us seem bigger than we were. We've always wanted to do that. You want to look as big and professional as you can."

**Best practices in documentation #4:
Have a reasonable ratio of writers to developers.**

Many software firms employ 20 or 30 developers for every tech writer — especially if the founder is a technical guy who likes to hire people like himself. But this makes it next to impossible for their documentation to stay current with their software.

When Thompson arrived at Interface Software in 1997, he was the second writer at the growing firm.

"That was two out of 28 people focused on technical writing, which was a pretty big commitment, given our size."

Today he's a full-time manager, with 25 developers for his three writers. That reasonable ratio of 8:1 shows management cares about documentation.

**Best practices in documentation #5:
Place technical writers somewhere sensible in your org chart.**

Technical writers sometimes get shuffled from one department to the next with each corporate reorganization. Many manager neglect the documentation team like an unwanted stepchild.

At Interface Software, technical writing is close to two related teams in the org chart: training and customer support. As director of documentation and training, Thompson reports to the VP of customer and partner services.

What's the fit between documentation and training?

"It's all about how to most effectively present the information to meet our customer needs. A lot of what makes good training makes good documentation. It's just a different medium," says Thompson.

"From the beginning, we've had documentation within customer support, as opposed to being part of development or some other team.

"We're sitting next to the people talking to our customers every day. That gets us closer to our customers, so we can really understand what they need."

**Best practices in documentation #6:
Keep technical writers in the loop on development plans.**

Although their whole job is about communication, technical writers are sometimes the last to hear about future development plans or last-minute changes to a release.

Not so at Interface. To start, their developers work from solid design documents and release plans. And they keep the writers informed every step of the way.

"We're involved from the beginning in design reviews," says Thompson. "The docs team is on the developer's e-mail list. We go to all developer meetings.

"We see every time a piece of code is checked in, so we literally know every time there's a change in the software.

"We're talking on a week-by-week basis about what we've done and what we're going to do next. So, we're seen as members of the development team."

That close interaction takes the rough edges off development ideas and gets the documentation done faster. And it's always been that way at Interface.

"Tech writers are good advocates for the user, always thinking about the user's point of view," Thompson says.

"Early on, we had that kind of representation in making design decisions or choosing the wording that appears on the screen. This definitely benefits the usability and acceptance of the product."

**Best practices in documentation #7:
Encourage technical writers to meet customers.**

Technical writers do tend to be the user's advocate, arguing for clarity and logic in everything from error messages to how features are implemented interactions. But that's hard to do when you never talk to any customers.

In many firms, managers actively forbid technical writers from meeting any live customers. But Thompson doesn't have to fight this attitude.

"We take every opportunity we can uncover to get close to our customers and understand what they need," he says.

For instance, when partners and customers are in-house for training courses, any employee of the company can sign up to have lunch with them. And his writers do.

"It's another point where you can talk to customers, find out what they need, and put faces to people you talk to on the phone. The customers love it, and so do we."

**Best practices in documentation #8:
Use customer advisory boards to get feedback on documentation.**

Interface also has customer advisory boards driven by product management. And the technical writers are encouraged to take part, Thompson says.

"We have current customers brought in, and I get on the agenda so we can ask who uses the documentation, the help, or the support web site. And then we ask things like: 'How is it being used? Does it meet your needs? How can we improve it?'"

What they hear is sometimes surprising.

"When we talked to our customer advisory board, we weren't sure whether the technical people were using our 'What's This?' help." This is field-level help that pops up when you press F1 or click a question-mark icon.

"It takes a lot of time to maintain that type of help, and we were skeptical that they were using it. So we asked, and it turned out they were.

"This was great feedback to have, and it helped motivate our team to spend the time to do it."

**Best practices in documentation #9:
Make the right tradeoffs.**

Like any manager working with finite resources, Thompson sometimes has to make tradeoffs. This is when the perfect is the enemy of the good.

"If we were able to spend more time crafting our sentences, we could improve the readability, and that would be better," he notes.

"But that's not as important as getting all the right information, recommendations, and navigation into the document. Well-written sentences that customers can't find don't do them any good.

"In a utopian world for tech writers, you would have as much time as you wanted to get things phrased just the way you want. But we have to get our docs done at the same time as the software.

"We have business needs for releasing new versions of software, and we're not going to hold that up. So, we try to make good business decisions about how to spend our resources and prioritize our projects."

**Best practices in documentation #10:
Pick the right medium for each deliverable.**

One challenge most CRM implementations face is getting people to use the system.

This is critical for Interface, whose primary end users are executive-level professionals such as lawyers, investment bankers, accountants, and venture capitalists.

These professionals' busy assistants maintain the CRM databases and generate reports. And their IT people configure and tweak the system as needed.

To support all these different users, Interface created an awesome suite of documentation: 3,000 pages of print, all that again as PDFs, and 3,200 topics in various forms of online help.

Not to mention added material on their web sites, in Wizards, and as toolkits for champions and trainers.

"We provide all different kinds of tools to help our customers market the software internally," says Thompson.

"We've got PowerPoint presentations. We've got a Flash movie that's very slick. We've got cheat sheets and little HTML tips they can e-mail. We've got artwork they can customize and put up.

"We provide some things in electronic form only, because that's how they're going to be used.

"For example, our Education Toolkit provides material for our internal trainers in Word format, because they need to customize it and integrate their own business processes and policies."

What's the key to choosing how to publish each piece?

As always, ask your customers.

"We go back to the customer, and ask, 'How is this information going to be used?'

"If they're just looking up a single point in the reference material, online might be the best approach. If it's something that we actually expect them to sit down and read, we have it printed."

**Best practices in documentation #11:
Provide print for those who need it.**

Speaking of print, in the past 20 years printed software manuals have gone from thick to thin to invisible.

Most software firms now provide PDFs or online help only. Naturally this saves big time on printing and shipping.

But in some cases it's irritating and self-defeating.

What good is online help on troubleshooting software when the software won't run in the first place? And why should software publishers get away with outsourcing all the printing costs to their customers?

Interface Software hasn't bought into this approach.

"We haven't eliminated printing. We don't look at it as a cost decision. If information is going to be used in multiple ways, we'll put it in multiple formats," says Thompson.

That's why they provide a printed library of 3,000 pages to every customer. And if they want extra sets, Interface sells them, at their cost.

"For the one or two technical administrators or power users with each customer who use it, it's worthwhile for us to print the reference material," says Thompson.

Best practices in documentation #12:

Give your writers the right tools for the job.

Microsoft Word is fine for routine office documents, like letters and memos. But it quickly runs out of gas when you try to do intense formatting on a 400-page manual.

Most tech writers can tell you war stories about having an entire manual corrupted by Word's notorious "master document" feature. Or losing a whole day struggling to get a buggy feature like auto-numbering to work properly.

Most software firms eventually adopt Adobe FrameMaker, the industry standard for long, structured documents. Even at \$799 a seat, most firms find it pays for itself quickly with more robust features.

Interface Software followed this well-worn path.

"We started out using Word, and we ran into all the usual limitations that Word has with large documents. Now we work primarily with FrameMaker," says Thompson.

There are dozens if not hundreds of postings on the web seeking answers to questions like "How do I convince my boss to let me get FrameMaker?" and "What's the difference between Word and FrameMaker?"

Sometimes, the exchanges among writers are both passionate and useful; here's an example. For summaries of the pros and cons of each app from a technical writer's perspective, you might want to look at the TECHWR-L and scriptorium.com websites.

Best practices in documentation #13:

Try out conditional text.

One of the most powerful features offered by FrameMaker is conditional text: passages formatted so they appear or disappear depending on what switches you flip when you print or generate a PDF.

How does this help anyone? An example from my own experience: One software firm where I worked made a UNIX GUI builder distributed by HP, IBM, and other vendors.

Using conditional text, we could switch between various product names, copyright notices, and other elements to generate a unique manual for each vendor in a few seconds.

Any other method, such as maintaining multiple versions of the same documents or doing multiple search and replace sweeps would have taken far longer.

"We use a lot of conditional text," says Thompson. "While we're writing a particular topic, we think about the different places it will show up, and the different media it will be in.

"So, we write unique content for the web version, the printed version, and the help version."

Best practices in documentation #14:

Explore single sourcing.

"Single sourcing" is a docs approach in which a writer stores everything in a one document, then from that outputs deliverables tweaked for different mediums.

The benefits are clear: Instead of maintaining a version of a document for a printed manual, another for online help, and so on, you've got a single source.

This approach is attracting lots of fans among publication managers. "Single-sourcing allows you to create more deliverables with fewer resources," says Thompson.

"And it's another way to look bigger than you are. We wouldn't be able to support our enormous help systems and 3,000 pages of documentation if we didn't use single-sourcing.

"Basically, we set up different templates so we can create PDFs, web pages, online help, and Word documents all from the same document.

"It requires a change of mindset for the writers, because they have to think about the different types of output while they're writing a topic. But once you get in the habit, it's pretty efficient.

Bonus tips: How to hire and motivate technical writers

If you've ever hired or managed a writer, you know it can be a daunting challenge. Finding someone with the right combination of language skills, technical understanding, and customer orientation can be tough.

Beyond all the best practices outlined above, here are seven bonus tips on how to find and motivate good technical writers.

Bonus tip #1: Hire writers with the right stuff.

What does Thompson look for in a technical writer?

The answer may surprise you: It's not just wordsmithing.

"Obviously they need to write well, but that's not really what I focus on," he says. "I want somebody who has a customer focus and really good investigative skills."

To help detect these qualities during interviews, Thompson asks applicants to tell him about some past projects.

"A lot is just their attitude. We don't want people who want everything to be handed to them. We don't want people to just translate technical documents.

"We want people willing to roll up their sleeves and be independent, try out the software, and put themselves in the customer's shoes."

Bonus tip #2: Hire writers who ask smart questions.

Being in the customer's shoes involves asking questions to clarify interactions that may not be clear.

"As a technical writer, you have to know what questions to ask developers," Thompson says. "If you don't know what questions to ask, you're not going to get the right information."

That takes preparation.

"We've built up a tremendous amount of credibility with our development team. That's because we do our due

diligence before we talk to them, so that we don't ask them stupid questions."

Duh; what's a stupid question?

"A question that you could have found the answer to on your own. That's just not the most efficient use of a developer's time."

By the same token, a smart question is one that makes them go "hmmm..." or "Well, we haven't figured that out yet."

Bonus tip #3: Don't get hung up on tools.

Despite the power of today's publishing software, experienced publication managers warn not to get too hung up on tools.

You don't need to be a power user of FrameMaker to follow the company style sheet. Software tools can be learned quickly. And the one power user in the team can usually solve everyone else's problems.

Many interviewers focus way too much attention on the candidate's software experience — perhaps because these are the easy questions to ask.

Don't neglect to probe how well they can solve problems, handle o-the-fly research, or perform under pressure. Those factors are more likely to determine their success than in-depth knowledge of any software tools.

Bonus tip #4: Provide lots of feedback.

Effective technical writing is a learned skill. Beginners can learn a lot from more experienced writers who take time to coach them.

Thompson schedules regular coaching sessions with each of his writers, from once to three times a week, depending on the individual.

"These sessions provide frequent opportunities to mentor, adjust priorities, and reinforce our core values. In addition to these, we have formal quarterly manager reviews and yearly peer reviews," he says.

On top of frequent interactions with customers and an annual survey on customer satisfaction, that adds up to a lot of feedback. All the better to help any writer improve their performance.

**Bonus tip #5:
Insist on timely reviews of drafts.**

Many product managers and developers don't regard the review of draft manuals as part of their job.

So, they get a draft from the docs people, then let them gather dust in the corner. Or, they give them a cursory once-over to circle the odd word they think might be misspelled.

And they get away with it, because their managers don't think it's important either.

When I worked as a technical writer, I once waited a whole year for the technical lead in a software company to review a draft manual I'd written.

Meanwhile, he stayed busy fighting fires and flying off to install systems. He had loads of time on those multiple airplane flights to read. The message was clear: my work didn't matter to him or the company.

This is not the way to motivate your writers and take care of your customers. Interface doesn't have this problem.

"Our culture of customer satisfaction helps us get internal reviews done by development and product management," Thompson says.

"Because everyone has bought into how it's important, they're willing to spend the time to make sure that our docs are accurate."

And they give reviewers explicit directions, asking them to focus on three areas: technical accuracy, product positioning, and appropriateness of the recommendations.

**Bonus tip #6:
Enter publication competitions for feedback.**

Interface's 3,000-page library recently won a prestigious award for Distinguished Technical Communication from the Chicago chapter of the STC, and went on to win a Merit award in the international competition.

Understandably, Thompson is proud of these awards. But he also values the judges' comments.

"I found the comments we received were well thought-out and gave us some good feedback. The suggestions the judges made were for the most part valid. It was a useful exercise."

Valuable feedback and maybe an award for your documentation: who wouldn't want that?

**Bonus tip #7:
Give your technical writers a chance to do more.**

"It would be great if executives were thinking about the skills that technical communicators bring to the party, other than writing books," says current STC president Andrea L. Ames, who in her day job works at IBM.

"Someone really looking at information in a strategic way can help gain customer loyalty. Customers appreciate getting information that's easy to use and consistent and really aimed towards helping them do their jobs."

"And enabling a company to communicate this way is a really interesting skill that a lot of executives don't see," she says.

"They may see that as a marketing communication role. But the added dimension a technical communicator brings is that they actually know how the product works."

"They've been the first 'victim' of the product, and they're a much better user advocate than a marcom person who is really advocating for the company they work for."

"So, remember that technical communicators can help at that very strategic level."

Your technical writers are, we suspect, dying to make a more significant contribution to your company. Many are chock-full of ideas on how to do it.

And if you get out of their way and let them, they will.