12-1-2009

# Technical Writing for Software Documentation Writers: A Textbook on Process and Product

Elizabeth Warnke
*Kennesaw State University*, Elizabeth.Warnke@gmail.com

## Recommended Citation

Warnke, Elizabeth, "Technical Writing for Software Documentation Writers: A Textbook on Process and Product" (2009). *Dissertations, Theses and Capstone Projects.* Paper 50.

# Technical Writing for Software Documentation Writers

## A Textbook on Process and Product

By

Elizabeth Warnke

A capstone project submitted in partial fulfillment of the

Requirements for the degree of Master of Arts in Professional Writing in the Department of English

In the College of Humanities and Social Sciences of Kennesaw State University

Kennesaw, Georgia

2009

College of Humanities & Social Sciences
Kennesaw State University
Kennesaw, Georgia
Certificate of Approval

This is to certify that the Capstone Project of

Elizabeth Warnke

Has been approved by the committee
For the capstone requirement for the Master of Arts in
Professional Writing in the Department of English
At the December 2009 graduation

Capstone committee:

Elizabeth Giddens
Member

Mary Lou Odom
Member

# *Acknowledgements*

*To my loving husband, Michael, for his continuous support throughout my graduate career; to my mother, Patricia, for her encouragement to always reach for the stars; and to my advisors, Dr. Beth Giddens and Dr. Mary Lou Odom, for their guidance and support throughout my years in the MAPW program and during this process.*

# Table of Contents

# Preface

Up until my junior year of college, writing was always something I had to do; it was never a passion of mine. I never felt creative enough to write or had much confidence in my writing skills. So you can imagine my surprise when I was actually accepted into a writing program.  While I always managed to perform well in writing classes, science and math classes were my passion. Towards the end of my undergraduate career, I noticed that I had started to take writing classes as elective classes, just to have a change of scenery from the science building. I soon turned to writing for a creative outlet – as a way to add a splash of color to the back and white world in which I coexisted with physics. I then enrolled in writing-based physics and math classes, where I realized that I did in fact enjoy writing, so long as I was writing about things that interested me, such as scientific and technical subjects.

I spent my senior year figuring out how I could pair writing with science in the form of a lucrative job. One day, I walked by a poster for the Scientific Technical Writing program at the Massachusetts Institute of Technology. That was it! Technical writing … that is what I wanted to do. And so the job hunt began. After searching for a position as a scientific technical writer, I was hired for a job as the technical writer for a software development company and started four weeks after graduation. The writing position did not involve physics, but it was certainly technical writing. And more importantly, it was a job.

I was hired for this job having taken only a handful of writing classes and no technical writing experience. But I had a solid technical foundation. I must have impressed the hiring staff somehow during the interview, but thanked my lucky stars regardless because the company took a huge risk by hiring me. There were pros and cons to being the company's first and only technical writer. On one hand, I had the freedom to build the technical writing department as I pleased and to create my own writing standard, style, and deliverables. On the other hand, I had no guidance, no examples to follow, no direction on how to be a technical writer, and no idea how I would figure out this job or where even to start. My job task was described to me as "create product documentation" – that was it. Easier said than done.

Within my first few months on the job, I tried to learn about my position. I had to teach myself not only how to be a technical writer, but a writer for a software development company, in addition to brushing up on my writing and editing skills (and not to mention, learning about the products for which I was documenting). With no resources on staff to lean on, I relied on my researching skills to find books to teach me the skill set I needed to succeed in this job. I had a difficult time finding texts dedicated to technical writing for a software development company. I mostly found books about writing processes, which were not useful to me because I was not aware of what a "writing process" was or that it was applicable to my job as a technical writer.  To have a book that taught me the basic skills to be a successful technical writer for a software company would have been very helpful. If only there was such a resource.

Despite the struggle I had in my job during the first year, I really enjoyed being a technical writer. I always knew I would eventually go back to school to pursue a graduate degree, but I did not know what I wanted to go to school for. If anything, now that I was writing in the IT industry, I imagined I would return to school to earn an advanced degree in computer science. Since I was hired by a successful software development company as a writer with no writing experience or writing education, I did not think a writing education was required to be a technical writer. Helpful – of course, but required, no. After gaining about a year and a half of work experience, I realized how wrong I was about the skills required to be a technical writer. If I wanted to pursue a career as a technical writer, my writing practices had to be up to par with my technical skills. Now was time to go back to school.

I chose the Master of Arts in Professional Writing (MAPW) program at Kennesaw State University (KSU) instead of a program solely dedicated to technical writing because the MAPW program was advertised as producing well-rounded writers, not strictly technical writers. While I love being a technical writer, my passion may change, just as it has shifted from physics to writing. With a degree limiting me to a specified field of writing, my fear was that I would be confined to write in a technical industry. I entered the program on the Applied Writing track to learn more about technical writing. My goal within the MAPW program was to learn the skills necessary to be a successful technical writer so I could be better at my job, and in the event I ever left my job, to have the credentials to be hired for another technical writing position, perhaps even outside the IT industry. I knew my writing skills could use a little brush-up, but that was not the primary reason I entered the writing program; I wanted to learn how to be a technical writer. However, within my first semester, I quickly came to the disappointing realization that my writing mechanics were not up to par. How did I manage to get into the MAPW program when my basic editing and writing skills (in my opinion) were not at the graduate level? After considering dropping out of the program, I realized that I was going to have to work a little harder than everyone else.

Throughout my years working for a software development company, I learned a lot about the industry. I learned that many software development companies do not have technical writers or do not have individuals on staff dedicated to documentation. Many developers are tasked with documenting the products they have built, which leaves the documentation of each product to its own standard and writing style. Before beginning classes in the MAPW program, my mentality was aligned with that of many software companies – anyone can be a technical writer as long as he or she knows the technical side. If my technical skills were equal to or superseded my writing skills, I would be successful at technical writing. After all, anyone can write, right? The writing skills I gained in high school got me into and through undergraduate school – how bad could they be?

Throughout the MAPW program, I applied as many of my new skills to my job as I could. My practical experience as a writer juxtaposed with my education gave me great confidence as a writer and showed me the importance of having a technical writer as an integral part of any software development team. When the time came to develop a capstone topic, the idea to write a textbook on technical writing for a software development company came very quickly to me. I wanted to use the culmination of my career in the MAPW program as an opportunity to document what I have learned from the program paired with my knowledge in the field. The goal with my capstone project is to provide a powerful resource to aspiring technical writers working for a software development company and to developers who document their own products. I wanted to write the book I could have used when I first became a technical writer.

While researching for my capstone project, I found limited resources on technical writing in software development companies. Reflecting upon the great need for technical writers in software development companies and learning that no one has published the thoughts and experiences in a textbook validated that my capstone project could fill a great void.

I had the desire and the motivation to create a text that, had such a text existed two years ago, would have been immensely useful to me when I entered the technical writing profession. I also had the beginnings of such a text already written. During Dr. Mary Lou Odom’s class *Understanding Writing as a Process*, I completed an assignment – “create a project about the writing process” – by writing the first chapter of a hypothetical technical writing textbook. The chapter provided an overview of deliverables and communication processes and dove deep into the writing process, more specifically, the process for technical writing. This 20-page chapter would serve as the starting point for my capstone project.

After weeks upon weeks of researching, I stumbled upon two books that catered to my exact needs, both written by the same author. Michael Bremer's UnTechnical Writing teaches the mechanics of writing about technical subjects for non-technical audiences, while his The User Manual Manual discusses how to research, write, and edit software manuals. While these books were written ten years ago, they still provided great value and validated that what I was writing about was useful to the field. However, since technology has exponentially grown over the last ten years, these books are already outdated. My capstone would be aligned with Bremer's books but would discuss how the latest documentation trends, such writing quick reference guides and tailoring a document to audiences of various technical levels, and the most recent communication processes, such as Instant Messenger services and WIKIs, impact and shape the writing process used by today's technical writers.

I spent several weeks sifting through textbooks on product and process and on technical communication. Since I am relatively new to the technical writing profession, my experience level is minimal and narrow. I needed to vicariously gain experience through veteran technical writers. I then looked for recent scholarly articles written in well-known technical communication journals, such as *Technical Communication* and *Technical Communication Quarterly*. I found a lot of work written by active technical writers that gave me a good perspective of current practices and methodologies. While these journals contained great ideas for my capstone, I wanted more recent information – work produced by technical writers within the last few months or even days. Recalling a paper I wrote on the latest trend in teaching writing – academic blogging – I decided to research blogs in order to learn about current technical writing trends and practices. Throughout my capstone research and writing process, I began following select technical communication blogs, such as *I'd Rather Be Writing* (http://www.idratherbewriting.com/) and *The Creative Tech Writer* (http://www.creativetechwriter.com/). Through these bloggers, who were dedicated to documenting their experiences as technical writers, I learned a lot about what other industries do and what makes writers in the computer software industry unique. In addition, I learned strategies and practices being used in the field today and practices that technical writers no longer use.

While researching for this capstone, I discovered a lot of my self-taught technical writing practices were on target. But I also discovered that some practices and methodologies I developed were not exactly in line with industry standards. For obvious reasons, I did not want to write about things I knew I was doing incorrectly. However, there were deliverables, communication processes, and elements to the technical writing process that I knew were important to include but I had no experience or knowledge of (other than what I had researched). In the technical communication blogs I followed, I noticed common trends that made me believe the elements I had not previously considered added great value to the profession of technical writing. These foreign elements included deliverables I have never produced (for example, quick reference guides) and practices of which I was aware, but did not fully understand the importance (such as writing for multiple audiences). In the end, I decided that despite my lack of ethos that prevented me from accurately writing and instructing about something for which I have no experience, I decided to include these elements into my textbook anyway.

What I lacked in ethos I made up for in determination and confidence. Because there was no existing documentation when I was hired, I had taught myself the fundamentals of technical writing, how to write for a software development company, how to produce deliverables useful to my target audiences, and how to teach myself about each software product. I viewed learning these important attributes to technical writing as another chance to teach myself – and others – something new. In my textbook I wanted to paint a well-balanced and accurate picture that represented working as a technical writer in a software development company. This complete picture would in turn enhance my document and enable the textbook to be useful to a wider range of audiences.

As I began drafting my capstone, I continued researching to find additional information to further enhance my textbook. I used the researched material and my practical experience to create an outline of each chapter.  In my capstone proposal, I mapped out the contents of each chapter on a high level. During the drafting phase of the writing process, I added details, such as sub-headings and quotes. My next step was to extract reusable information from the textbook chapter I had written for Dr. Odom's course. I then continued through the outline and researched sources to draft each chapter. I began drafting the chapter that was the easiest for me to write (Chapter 4: Deliverables) and finished with the most difficult (Chapter 2: Ethics).

Because the field of technical writing is broad, narrowing the scope of my capstone project to writing for software development companies was not enough of a focus. Keying in on a specific audience helped determine what information to include, and more importantly, what information to exclude from my textbook. Since the audience of the textbook is writers with technical backgrounds, I wanted to explain the fundamentals of the writing process, which is an element most technical writing books omit. I also decided to exclude a lot of technical information about software development because I assumed this is information the audience already knows.

Many articles I referenced in my textbook, such as Shaun Slattery's "Technical Writing as Textual Coordination: An Argument for the Value of Writer's Skills with Information Technology" and Matthew Friedman's "Advances Useless without Knowledge: As Computer Software and Hardware Gets Increasingly Complex, Companies are Becoming More Aware of the Importance of Technical Communicators," have a similar trend to each other. These articles were written to justify the need and the importance of technical writers, not only within the software industry, but as a profession. As a technical writer currently working in the field, I am aware of the value my work brings to the company.

Articles like these brought insight on the skill set required by technical writers. In justifying the importance for every software development company to have a writer dedicated to documenting products, these articles discuss the traits and characteristics someone in a technical writing role should have. Slattery summarizes the skill set for writers in the IT field well in his *Technical Communication* article. He states that technical writers who work for software development companies:

> Must be broad thinkers who can visualize projects before they're developed and see connections between seemingly unrelated material . . . They must move easily among genres . . . and media. They must be comfortable designing interfaces and interaction . . . have a good "eye" for the visual look of a communication product and a good "ear" for its verbal tone (Slattery 353).

I wanted to convey the importance of this duel skill set Slattery and many other writers mention. Technical writers must encompass all that is described in the job title – these individuals must be as technically inclined as they are solid writers. As I wrote my capstone, I wanted to express that the audience of the textbook should have both skill sets. The future of the technical writing profession depends on writers who have backgrounds (education and/or experience) in writing as well as in a technical field. This concept, which I unfortunately learned the hard way, is very important in order for a person to be a well-rounded technical writer.

Composing a technical writing textbook targeted towards writers aspiring to work for a computer software company and towards working technicians has been a great culmination to my years in the MAPW program at KSU. I enrolled in the program to learn more about technical writing, to be able to apply my education to my current job as a writer, and to grow professionally. I love what I do for work, and after concluding the MAPW program with this capstone project and learning more about writing for the software industry, I love my job even more. Completing my capstone project opened my eyes to what I have been doing correctly and what practices, deliverables, and methodologies of technical writing I need to modify in order to grow professionally.

If I had time to expand on my project or if I decided to eventually publish the textbook, there are a few elements I would change. While reading about how to perform a task is a fine base to learning, most people learn best by doing or by looking at examples. You can read one hundred books on how to play baseball. But until you watch someone play and throw a ball or swing a bat yourself, you have not fully learned America's favorite pastime. The same methodology applies to technical writing. Therefore, I would incorporate exercises as well as additional real-world examples into the textbook. The exercises would encourage the reader to practice writing for different audiences, narrowing the scope of a deliverable, and pulling information from other sources to aid in the writing process. For examples of deliverables I would include entire documents to provide the reader with finished products. Examples of entire deliverables would be very valuable to the textbook because many software company CEOs do not permit the distribution of software documentation without the purchase of the software, which often makes finding concrete examples of technical writing deliverables a challenge.

I entered the MAPW program with the desire to learn about technical writing and with the assumption that the writing skills I learned in high school and in college would suffice for the technical writing career I yearned for. This program taught me not just about technical writing, but about writing and the writing process. I exit the program as a more balanced writer than I ever anticipated.

# Chapter 1: Introduction to Technical Writing

## Chapter Contents

## Chapter Objectives

The following are objectives of this chapter:

- To outline the skill set required to be a successful technical writer.
- To illustrate the history of the technical writing profession in order to understand how exponentially the profession has grown.
- To identify a technical writer's role in a software development company.
- To access the process and product relationship and evaluate the significance of each component.
- To summarize the writing process used to create technical documentation.
- To enumerate the most popular deliverables produced by technical writers in the IT industry.

This textbook is designed for the technically literate technical writer within computer software and information technology (IT) companies. In the most ideal situation, a software development company has a team of writers, a team of editors, and a team of usability testers for its product and training documentation. In a less than ideal situation, the company has a team of writers and editors to create its product and training documentation. However, in a more typical situation, software development companies may not have any writers who are dedicated to produce documentation or may have a single writer who is also the editor. The writer in this realistic situation is the audience for which this textbook is targeted.

This chapter provides an overview of the technical writing field as well as how technical writers are utilized by software development companies. An important element to being a technical writer in a software development environment is knowing the value of your work. Documentation is important because it provides clients with the information they need in order to make decisions that are important to the business, such as buying a piece of software, upgrading to the most recent version, or learning how to use the software. An uncomfortable reality is that documentation is not nearly as important in the eyes of the developers and other subject matter experts – it is almost always put at the bottom of the priority list and viewed as an "extra," not a "necessary," element to the software. This reality is unfortunate because technical writers are dependent on developers and other subject matter experts for content.

## A Technical Writer's Skill Set

The skill set of a successful technical writer is extensive because technical writers are not just writers – they are designers, mediators, trainers, and technicians. The skill set outlined in this chapter will help you to get hired as a technical writer as well as be evaluated and be promoted within your current position.

### Writing and Editing

- **Editing Skills** – Have excellent editing skills and, if possible, have editing experience as well. Technical writers are known for being meticulous and detailed-oriented, not only with the content of their deliverables, but also with the mechanics of the prose.

- **Technical Writing Skills** – Have technical writing experience and/or a writing/technical writing education. Writing technical prose is a key skill to have.

- **Instructional Writing Skills** – Be aware of how to write clear and direct instructions that guide the reader through a process. Writing instructions can be an easy task, but writing poor instructions is an even easier task.

### Technical

- **Programming Concepts** – Learn basic programming concepts. A technical background may not be required, but a strong interest and ability to understand technical concepts is a necessity. In order to write about technical material, writers must be able to "wrap their brains" around the concepts. Technical writers must know the basic concepts of programming and computers, such as database components, server connections, etc.

- **Software Knowledge** – Be fluent with as many types of document and design programs as possible. With an array of document and design programs available, software companies have many choices of preferred programs. Each company you work for may use a different program. The following is a good set of software tools to know:
  - Microsoft Office, including Word, Visio, PowerPoint, Publisher, and Excel
  - InDesign
  - Acrobat
  - Project Management Software, such as MadCap Flair, Doc-To-Help, and Adobe FrameMaker

## Design

If the company does not have a designer or production manager, the technical writer is responsible for document design. Therefore, learn basic document design skills to enhance the usability and readability of each deliverable.

## Personality

- **Self-motivation** – Be aware of the development process and be up-to-date with release dates, testing cycles, etc. Know what needs to be done for each release and do it.

- **Strong Interviewing Skills** – Have strong interviewing skills to interview subject matter experts. Be courteous and sincere when interviewing to establish a solid rapport in order to obtain the information you need.

- **Solid Organization Skills** – Have solid organization skills in order to develop and manage the writing process.

- **Ability to Learn Quickly** – Be willing to dive into the product head first and learn the software. You may not always know everything about each product, but you have to make the attempt to and pretend that you do. Subject matter experts will not always have time to teach you the software.

- **Good Time Management Skills –** Be very familiar with deadlines and make sure you are kept up to date if there is a change (which can happen frequently). Be aware of how long it takes you to accomplish each stage of the writing process so you can properly manage your time and complete the project for the targeted release date.

- **Communication Skills –** Have strong verbal and written communication skills. Do not be afraid to ask questions. In order to write about a product, you need to learn the product. Verbal and written communication skills are imperative for technical writers. If you state your questions clearly, you are more likely to get a clear and direct answer than if you are unable to phrase your point with precision.

- **Sense of Humor** – Maintain a sense of humor. Do not let the stress of the high demands of being a technical writer wear you down. Maintain a sense of humor, so you do not lose sight of the assignment.

## What is “Technical Writing”?

The field of “technical writing” is difficult to define because there are many avenues and contexts that contribute to this style of writing. Traditionally, technical writers are viewed as writers who compose technical documents about science, computers, medicine and other technical subjects. However, this once narrow field has morphed into the broad classification of writing composed in the workplace. This textbook focuses not on general workplace writing (memos, reports, emails, etc.), but specifically on writing produced in a software development company.

Today, technology influences many aspects of everyday life. From the iPhone to Hybrid cars to Microsoft’s Office 2007 and Windows Vista, new technology has touched nearly every American. For the most part, Americans have reaped great benefits from modern technology, such as more efficient ways to communicate and safer ways to travel. However, there have been a few drawbacks to the technology boom. One detriment to the increase of technology is realization of the lack of technical literacy among everyday people. Technology is intimidating since it so rapidly became integrated into everyday life. As a solution, technical companies began hiring technical writers to serve as intermediaries between the engineers and the laypeople. The technology boom has caused a high demand for skilled technical writers.

In the context of this text, “technical writing” is limited to external writing composed by a software development company. In this sense, documentation is written for external audiences – end users of the software (i.e. clients) – and internal audiences – users who must be knowledgeable about the products (i.e. sales and marketing departments).

This section outlines the progression of technical writing as a profession from the very beginning and into the future. As a technical writer, you need to understand how much the profession has grown in order to appreciate the importance of your function in a technical environment. Evaluating your skills as a writer now compared to where they need to be to complete in the job market is beneficial to growing professionally as a writer.

### A Brief History

Understanding how the technical writing profession has evolved throughout history is important to recognize the progression of this profession and to appreciate the foundations the IT industry laid by founding technical writers.

The exact beginning of technical writing as a field or an occupation is uncertain; however, there is evidence that reports written on technical subjects (engineering, specifically) were exchanged well before the birth of Christ. One of the oldest and most famous reports is De Architectura written by Marcus Vitruvius Pollio around 40 B.C. This document discusses city-planning, hydraulics, and technical instruments. (Rathbone 26). But as Robert Rathbone, an engineering and scientific technical writer summarizes, "Although the art of technical writing is almost as old as science itself, the profession of technical writing has a recent beginning" (Rathbone 26). During World War I, the scientists and engineers were also the writers and documented their own inventions (similar to many software development/technical writing environments today). However, during the 1920's editorial assistance was provided to technical writers due to the prosperous economy produced by the industrial boom. Specialized writers were hired to document the inventions produced by developers. When the Depression hit and thousands of Americans lost their jobs, then these writers often lost their positions. World War II marked the true beginning of technical writing as a profession. During this time period a team of writers was formed to deliver paper work during the war. Members of these teams soon earned the respect of the engineers and became the sole writers. Major companies within the engineering business and the IT industry soon hired technical writers.

## Technical Writing in Recent Years

Today, technical writers greatly contribute to a variety of technical industries. These writers provide technical fields with the knowledge and the writing skills to document what has been created by the developers, leaving the invention to the technicians and the writing to the writers.

Technical writing, as a field, a course of study, and as an industry, has become widely popular in recent years. But why? Throughout history, the need and the demand for people to perform this specialized type of writing has always been evident. However, documentation within a software company or any other technical industry has always been thrown to the bottom of the priority list. After all, everyone knows how to write and string words together into sentences, right? While this is true, not everyone has the skill set to write well – a detail that few are willing to acknowledge. Therefore, hiring people designated to write remains a low priority for upper management.

Technical writers of recent years have proved that there is a need and a demand for this specialized classification of writers. Not only do these writers document the latest forms of technology, they are technically literate enough to translate the software jargon of the developers into a standard and clear format that a less technically literate audience can understand. Too few writers are learning technical fields, and technicians often lack essential writing skills. The individuals who possess both skill sets are rare, but they are in the greatest demand.

## The Future of the Technical Writing Profession

Recognizing the growth of your skill set as a technical writer will help you strive to be a better writer in the future.

While this genre of writers has been cast as just that, writers, the future requires a more technical background for this group. The role of technical writers has changed throughout history, and it will continue to change as society and culture change (Giammona 350). As technology increasingly occupies nearly every aspect of our lives, writers have to become technically literate to match the demand of technical documentation. A technical background provides a concrete understanding of the technical concepts. A better understanding means these writers are easily able to "translate" the concepts so the layperson can understand.

A technical background also contributes to a document's usability, and it can open up more avenues to communicate with the audience. People tend to be visual learners and most dislike reading. Therefore, to help increase communication with the audience, writers will need to become technically versed with modern software programs and concepts. Text can be placed directly in the software program in the form of page or field help. In addition, because so many people dislike reading (especially technical documentation), writers can also use writing skills to compose tutorial videos (Johnson). A simple three-minute video is a very powerful form of documentation that can appeal to many audiences.

Technical writers will be expected to have technical degrees in addition to writing degrees. However, the necessities for a technical writer will not stop there. Their skill set will expand to make writers better versed in the entire life cycle of the software development process. In addition, these writers need to acquire skills in software development and project management.

Technical writers are here to stay.

## Technical Writing in the Computer Software Industry

End users depend on software documentation almost as much as they depend on the software program itself.

Technical writers contribute the following to the computer software industry:

- Document products for clients and co-workers.
- Develop a process that incorporates all key players into a document's creation.
- Establish and enforce a company-wide writing style.

### Documenting Products

To an end user, documentation is deemed just as important as the software itself. If end users do not know how install, to configure, or to use the software, they may not use it. While many software users dive directly into the product before reading the corresponding documentation, the documentation is necessary in case the software is not as intuitive as the end user initially might have thought.

In addition, having documentation to accompany the product adds credibility to the software company. There are many companies that refute the importance of documentation because upper management demands that the software be self-explanatory, self-documenting, and intuitive to all audiences. If the software is easy for everyone to follow – who really needs documentation? What these individuals and companies do not understand is that since there is such a wide range of skill levels within the products' targeted audiences, there is no way to cater the software to all users. The documentation fills the gaps between the software and the skill set of the audiences.

## Incorporating Key Players

Many individuals contribute to the production of a document – far more individuals than one may think. The technical writer is often responsible for the mediation between the subject matter experts (SMEs), the writers (often themselves), the technical editors, the quality assurance teams, and the project managers. Since files are distributed between the individuals involved in the writing process, a process must be defined to ensure there is only one working version of a document and that every edit is accounted for. The facilitator role often falls on the shoulders of the technical writer since this individual is viewed as the "owner" of the document. A system that works well for all users is crucial to the success of the document.

Some companies utilize a document authoring tool, such as Doc-To-Help (http://www.doctohelp.com/) or MadCap Flare (http://www.madcapsoftware.com/products/flare/). These third-party programs help facilitate and control the writing process by utilizing single-source editing and workflow emails (and usually have high-quality and detailed documentation). Other companies use inside resources, such as an internal shared drive on a shared network, to facilitate the process. The foundation to any process that incorporates many individuals is to always ensure that only one person edits a document at any given time. Losing site of the "active" document is often the source of missing edits, duplicate information, and incorrect formatting.

Whatever the process may be, it must be usable with little effort on the part of the participants. The focus must be on perfecting the content of the document, not on the next step in the document life cycle.

## Standardizing a Style

While many technical writers despise being known as "document formatters," this role is a huge part of the job. Unlike other professional writers, such as journalists or novelists, technical writers must be skilled in document design because the "look" of a document can affect its readability and usability.

Documents within a single company must have cohesion, which enables readers of any document type to see that the document is part of a set. This structure, in turn, builds credibility for the company.

A standard writing style must not only be consistent within a single document, but also across each document type. A style incorporates aesthetic and linguistic considerations. An aesthetic standard refers to the “look and feel” of a document, including colors, headings, spacing, and font selection. If a document is not visually appealing to a user, the user may not bother reading it. Once the reader’s attention is caught and maintained, the importance of a language standard surfaces. Since aesthetics are not within the scope of this textbook, a wonderful resource is Robin Williams’ The Non-Designer’s Design Book. In a clear and easy to understand format, Williams describes the four basic principles of document design: contrast, repetition, alignment, and proximity. He speaks to individuals with no prior design experience and instills a sense of confidence in the reader to design any document type.

A standard language style leads to comprehension on the reader’s behalf and usability on the document’s behalf.  For example, if a document is littered with corporate jargon that does not pertain to anyone outside the corporation’s walls, the end user will be lost. Or, if there are inconsistencies with word choice, such as “web application,” “webapp,” and “application,” the end user will be unable to digest the content. If a different word is used, even if all three words have the same definition, the implication is unclear to the reader because a different word implies a different definition.

An easy way to facilitate documentation standards (both aesthetically and verbally) is to develop a series of design guidelines. There are two primary types of design guidelines: a style guide and a style sheet. A style guide is used company-wide and is written to define writing conventions established within an organization. A style sheet pertains to a specific document or document type. This document includes mechanical choices, such as punctuation, spelling, capitalization, and hyphenation, that recur within a document.  A good reference on creating and using style guides and style sheets is Technical Editing by renowned writer Carolyn Rude. In this text, Rude provides examples, templates, and guidelines for creating styles guides and style sheets for a company or documentation project.

## Process versus Product

Process and product. Both are equally important; however, a technical writer typically emphasizes the product over the process. Technical writers who stem from a writing background generally understand the concept of a “writing process” and the impact it has on the overall success of a document. However, those who stem from a technical education (which has become increasingly common among technical writers) may not be familiar with the writing process or the infamous debate of process versus product.

### The Process

Well-written documents can only be produced if the act of writing is treated as a process. There are four primary stages of a writing process: prewriting, drafting, revising, and editing. For technical writing, however, often the process is slightly different because all the stages are very collaborative and they often occur simultaneously. The writing process varies from genre to genre, company to company, and person to person.  The writer must learn about his or her writing style and determine (often through trial and error, and available resources) what works best. For example, some writers prefer to edit online while others prefer to edit hard copies and transfer the emendations to the electronic version (however, if high volume of printing is unavailable, due to budgetary constraints, for example, this process may not be an option).

The writing process for technical writing is distinctive because the stages, though individual, often occur simultaneously and overlap. The writing process is more of a circular than a linear process, since stages are often revisited even after they are considered "complete." In addition, collaboration is a significant entity to every stage of the process. Looking specifically at drafting, the writer is often not the subject matter expert. Therefore, he or she is dependent on other team members for the content. Collaboration also occurs in the editing phase as additional writers often lend their expertise to perfect the prose.

The writing process for technical writing is very distinct when compared to that of creating and academic writing processes. This topic is covered in greater detail in Chapter 4: The Writing Process.

## The Product

The product is the end result. But in technical writing, is there really a "finished" product? Just when you think you have completed a document, a new feature is added to the software or you will push commas around until you have considered the prose perfected.

The director of the Writing Programs at Washington State University, Bill Condon, comments on the process and product relationship, "Process and Product, in a way, are just meaningless … as we know it, no piece of writing is ever finished; it's just due" (Take 20: Teaching Writing). Each writer must create his or her own writing process; because after all, the facilitator of the process produces the end result.

With stringent software release dates, product documentation always has a date where it must be considered "finished" (or more appropriately, "just good enough").

Technical writing within a computer software company produces many deliverables. This textbook covers some of the most widely used and popular documents produced by technical writers in the field.

- **User Manuals** – Document a software product's functions, including navigational instructions and tricks and tips. Educate the audience about the product, answering questions such as *What does the product do?* and *What is the intended use of the product?*

- **Release Notes** – Provide an overview of how the current release differs from the previous release, including new features, enhancements, resolved issues (bug fixes), and known issues (bugs that will be addressed in a future release).

- **Product Descriptions** – Illustrate a high-level overview of the product, which outlines the product's goals and key features.

- **Training Materials** – Educate the end users about how to use the software, beyond the limitations of a user manual. Include exercises with real-world examples that can be completed during class and used as reference material once the class is over.

- **Quick Reference Guides** – Provide an abbreviated version of a user manual. They are targeted towards a more advanced audience.

This topic is covered in detail in Chapter 4: Deliverables.

## Chapter Summary

The following key points have been covered in this chapter:

- The qualities of a successful technical writer include a still set that contains strong writing and editing skills, technical skills, technical writing skills, and organizational skills, and a highly motivated personality.

- The technical writing profession continues to grow and transform with the advancement and expansion of technology.

- The role of a technical writer has evolved from strictly that of a writer to a more technically literate individual who is an integral part of the entire production process, not just the writing process.

- While more attention is placed on the end product in technical writing contexts, the process is equally important.

- The writing process is circular, since a product is never “finished.”

- The following are the most widely written and used documents produced by a technical writer in a software development company:
    - User Manuals
    - Release Notes
    - Product Descriptions
    - Training Manuals
    - Quick Reference Guides

# Chapter 2: Ethics

## Chapter Contents

## Chapter Objectives

The following are objectives of this chapter:

- To describe ethical considerations a technical writer may face.
- To understand how copyright laws protect technical writers.
- To introduce the legal issues involved with technical writing.

The decisions you make as a writer do not only affect you individually, but also your employer and the technical writing profession. Because your writing represents the company, writing ethically is a direct representation of the company's morals and values.

## Ethical Considerations

Ethics is a topic rarely discussed or even thought about in a collaborative environment. Ethical decisions are based on knowing right from wrong. Sometimes corporate policies or politics could challenge a writer's judgment in making an ethical decision. For example, knowing there is a detrimental bug in the software that could negatively affect a client but not mentioning its presence in the release notes because the bug could potentially cause the account manager to lose the sale, is an unethical decision. The writer can pretend to not aware of the bug, but he or she will have bigger issues when a lawsuit is filed against the company for the insufficient testing that could have potentially unveiled the bug.

Ethics for technical writers are applicable both when collaborating (during the planning and drafting phases) and when writing.

### When Collaborating

Work ethics can be pertinent when functioning in a collaborative setting. When working with others, keep the following in the forefront of your mind to avoid an unethical situation or being forced to make an unethical decision:

- **Be aware of your personality type.** If you have a strong personality (chances are if you have such a personality, you are well aware of it), be cautious of intimidating your co-workers. Everyone has an equal voice, and if you come off strong, timid employees will be afraid to speak and to contribute to the meeting.

- **Claim credit for your work and only your work.** Unfortunately, plagiarism within a work environment is common. Make the ethical decision to give credit where credit is due. Also, if you witness credit being stolen, ignoring what has taken place is unethical. If this situation surfaces, encourage the victim to confront the individual who claimed credit, without serving as the intermediary or getting too involved.

- **Do not withhold important information.** If you know the solution to a problem or how to conduct a process, speak up. Hording information is an unethical decision because the act may put your team or your company in a difficult situation. Revealing your knowledge may produce more work for you, but it is the ethical decision.

- **Have a solid process for documenting version control.** Since drafts are circulated between various participants of the writing process, such as writers and developers, include each individual's edits. If you lose valuable edits on functionality, for example, the document will contain false information and therefore will misrepresent the product.

## When Writing

As a writer who produces deliverables targeted towards external audiences, there are various considerations to keep in mind while writing. Considerations, such as the following listed, help you stay honest and credible with the public.

- **Do not withhold information.** If you know a feature is not working correctly, tell a developer so it can be fixed. If the software is released with a bug or if new system requirements are mandated, for example, document the problem.

- **Write clearly so as not to sound ambiguous or misleading.** Being too detailed and convoluted in your writing can be interpreted as a cover-up for something. When you write directly and clearly, you give the audience only one way to interpret the information.

- **Avoid writing in the singular third person.** Using "he" and/or "she" pronouns can make the company sound sexist (your writing, after all, is a representation of the company and its values). Avoid this issue by pluralizing pronouns to "they" or "their" instead.

- **Include accurate and correct information.** All developers (subject matter experts) review documentation before it is finalized in order to verify all the information is correct. This process also adds integrity and credibility to the document because the experts have provided their input.

- **Include the names of the developers on the credits page of the document.** Listing the names of the subject matter experts in the document is not only ethical because it verifies the information is from a valid source, but also having the expert's name included establishes the credibility of the document.

## Legal Issues

When writing for a company, you are often protected from legal issues that could surface. Be aware of your rights as a writer and of laws that speak specifically to you and your work. As a public communicator you are bound by certain laws. Ask your company paralegal about these laws or conduct research at the U.S. Copyright Office.

### Copyright Laws

As soon as a document is "created and fixed in a tangible form that is perceptible either directly or with the aid of a machine or device" it is copyright protected (U.S. Copyright Office). For a writer, this means that as soon as a document is written or printed, it is copyright protected.

#### Infringement

Using information from outside sources (beyond information found through research) is a legal hazard and is an act of copyright infringement. As defined by the U.S. Copyright Office, copyright infringement is specified as using another's copyrighted work to conduct the following:

- Reproducing work or copies of work
- Preparing derivative works based upon the work
- Distributing copies of the work to the public by sale or another transfer of ownership, or by rental, lease, or lending
- Performing the work publicly
- Displaying the work publicly

In a technical writing setting, copyright infringement can be viewed as using copy of a third-party company. If the software you write utilizes or can be used with a third-party software program, information about this third-party program may have to be documented.

Another common act of infringement can be seen with press releases or public media. An article about the software company or its products written by a reporter belongs to the newspaper or magazine. Lifting copy of a press release written by a third-party company is an act of infringement.

Infringement is a serious violation with an array of consequences, depending on the severity of the crime. A conviction of infringement can result in one or a combination of the following punishments:

- Payment of a fine
- Impound of infringing articles
- Restoration of damages
- Forfeit of profits
- Payment of victims attorney's fees

Since copy produced for your employer is intellectual property of the company, in most cases, an infringer may not personally be held liable (from a legal standpoint – there may be internal consequences, such as termination of employment). Instead, the company will be held accountable for your actions. Details of who is held legally accountable are usually outlined in an employment contract.

## Fair Use

The Fair Use doctrine of the copyright law entitles the owner of the copyright (i.e. the software company) the "right to reproduce or to authorize others to reproduce the work in copies" (U.S. Copyright Office). Every piece of writing you produce for the company (as well as writing produced by others) becomes the intellectual property of the company (U.S. Copyright Office). Simply put, even though you wrote the document, it does not belong to you.

Being the writer does not automatically grant you the authority to reproduce the work – permission must be obtained from the proper internal resources. Fair use, however, allows you to use direct copy of a co-worker. Since all work produced by employees of the company is copyrighted to the company, work is produced from the same source and therefore the use of a co-worker's copy falls under the Fair Use doctrine.

Common applications of this exercise are using marketing material to find the right language for a product description or lifting text directly from a specification document for a user manual.

Using material from other employees within the company is beneficial to the cohesion of documentation between departments. This cohesion verifies the same terminology and tone are used across the entire company.

Reproduction of an outside copyrighted source is considered fair under the following circumstances:

- Criticism
- Comment
- News reporting
- Teaching
- Scholarship
- Research

In a technical writing setting, fair use can be seen when researching third-party applications to gain an understanding of how the applications are used with the software for which you are writing about.

### Infringement versus Fair Use

There is a fine line between infringement and fair use. Talk with the company's paralegal about how these laws pertain to you in your current work environment.

When deciding if use of a document or reproduction is fair or an act of infringement, consider the following factors, as enumerated in Section 107 of the U.S. Copyright Law:

1. The purpose and character of the use, including whether such use is of commercial nature or is for nonprofit educational purposes.

2. The nature of the copyrighted work.

3. The amount and substantiality of the portion used in relation to the copyrighted work as a whole.

4. The effect of the use upon the potential market for, or value of, the copyrighted work.

## Trademarks

Trademarks protect names within a company, such as the product names, the slogans, and the company name. Always have the most recent version of the company's trademark list, which can usually be obtained from the company paralegal or from the marketing department.

The following are symbols of protected material as directed by the U.S. Patent and Trademark Office:

- **Trademark (™)** – A brand name that may include a word, a name, or a symbol. The intent of the brand name is to identify the product as an entity of the company. The user must legally register the brand name with the federal government. Before it is registered, the company is not fully protected by copyright or infringement laws.

- **Service Mark ($^{\text{SM}}$)** – A unique service, such as advertising. Service marks are often used to promote businesses. Federal registration is not required.

- **Registered Trademark (®) –** A brand name that may include a word, a name, or a symbol. The intent of the brand name is to identify the product as an entity of the company. This symbol indicates the brand has been legally registered with the federal government and is therefore protected at the highest level from copyright or infringement laws.

- **Copyright (©)** – Protects copy, which may include websites, user manuals, and marketing materials. An entity produced by the company, including software, is copyright protected.

Use the correct trademark on a brand name. Trademarks are highly important in order to protect the intellectual property of the company. Your company probably has branding guidelines that instruct writers how to use, and more importantly, how not to use company trademarks. A branding guide can often be obtained from the company's paralegal or marketing department.

The following are common rules of trademark use:

- Refer to a trademark using the correct spelling and with the correct capitalization. Do not alter the trademark in any way.

- Include the appropriate symbol (™, ®, or $^{\text{SM}}$ ) to represent the trademark status.

- Do not use a trademark in the possessive form.

- Do not use a trademark in a plural form.

## Chapter Summary

The following key points have been covered in this chapter:

- Technical writers often encounter ethical situations when collaborating and when writing.

- Technical writers are copyright protected by the fair use doctrine and infringement policies. Using text produced by another source within the company is fair use.

- Common legal issues of technical writing include copyright laws and trademarks.

# Chapter 3: Communication Processes

## Chapter Contents

## Chapter Objectives

The following are objectives of this chapter:

- To describe how academic, creative, and technical writers utilize collaboration.
- To evaluate the benefits and challenges of collaborating on a project.
- To analyze the various electronic and oral communication processes and decide the best one to use in any given situation.

The magnitude of collaboration throughout the entire writing process for technical documentation makes technical writing a unique writing profession.

Technical writing is a group effort primarily between the writer and the developer(s). The writer depends on the developer for content and the developer equally depends on the write to “translate” very technical prose so that a layperson can understand the content (and thus use the software program to its highest potential). Unlike academic and creative writers, technical writers are often not the subject matter expert, an arrangement which therefore creates a dependent relationship with the developer. In addition to collaborating with the subject matter experts, numerous technical writers often work on a single project together. While having developers with their knowledge on the product review the documentation for accuracy and completeness, having an individual with writing and editing experience read it is just as important. Another writer is beneficial to the collaborative effort to verify correct use of grammar, clear language, and document standards.

When collaborating on a project, whether it is a single document or a document set for an entire software release, communication is a skill that every member of the team needs. Collaboration is achieved through both electronic and oral processes.

## The Importance of Collaboration

Collaboration is essential in any work environment. This section outlines how collaboration is applied in a technical writing profession when working with developers and with other writers.

When working with co-workers or team members, remember that collaboration is about creating a final product. Collaboration is not about competition. All collaborators work for the same company and have to keep the good of the project (and of the company) a priority. Being part of a team allows you to learn from as well as to teach others. Collaborating on a project yields many benefits as well as challenges to the writing process.

Collaboration is used to not only produce well-rounded, well researched documents, but it can enable the participating voices to learn about other writing processes that exist within a given corporate environment. While a collaborative project is often time consuming and labor intensive, the benefits to collaboration far exceed the drawbacks. As Julie Zelenick of Iowa State University proclaims, "To become a good technical writer, you not only need to write effective documents on your own, but you need to know how to write documents, collaboratively with your colleagues" (Zelenick 68). Good writers become great writers through collaboration and discovering new ways to improve their writing process. In addition, by collaborating with subject matter experts, designers, and publicists, technical writers quickly learn about other fields and the ways writing impacts various disciplines.

## Benefits of Collaboration

Collaborating on a project with a team provides many benefits not only to the success of the deliverable, but it aids the team members to grow professionally. The following are some key benefits to collaboration:

- **Keeps the writer honest.** When working on a deliverable as an individual writer, concepts, organization issues, or mechanical or content errors are often overlooked. Splitting your attention between content and writing mechanics is a difficult task. Having other people look at your piece provides multiple perspectives that may help you recognize areas that have been overlooked. In addition, reading the content of others can help you spot mistakes in your own piece.

- **Provides a "sounding board**." Making decisions before consulting at least one other person, especially when the decision impacts other members of your team is never a good idea. Never work in a vacuum. Talk to others about your ideas before implementing changes.

- **Instills confidence in team members.** Collaboration provides a sense of confidence and makes team members feel important within the workplace. If an individual is shy about speaking in front of his or her boss or the entire company, a more intimate group can be a less intimidating setting to voice one's opinion or to ask questions. Collaboration instills a sense of empowerment – individuals gain confidence when they feel like their opinions are not only heard, but are considered valid.

- **Encourages team building and bonding.** Team building is a great additive to collaboration. Collaboration enables you to work with individuals who you may not otherwise work on a project with. It creates a sense of cohesion between the team as well as within the office work environment because everyone is striving for a common goal.

## Challenges of Collaboration

Working with others can often present a series of issues or complications that could negatively affect the outcome of a project. Collaborating with co-workers introduces multiple work ethics, personalities, knowledge levels, and skill levels. Having respect for your team members and pushing personal feelings aside (positive and negative feelings) is essential to maintain a positive collaboration environment. Your goal is to produce perfect, cohesive deliverables, not to make everyone on your team your best friend.

Conflicts often surface when working with numerous personalities, opinions, and status levels. In a healthy team environment, everyone's opinions and voices are heard equally.

The following are practices that can solve or even prevent conflicts:

- **Be the leader.** As the technical writer, you are often the team lead in a project. You orchestrate who writes what sections of the deliverable and control the flow of the document-approval process. Take command of this role – every team needs a leader and whether you acknowledge the position or not, you will fulfill this void.

- **Set objectives and goals with deadlines for each team member.** Conflicts often surface when individuals do not know their responsibilities within a project. Team members must be aware of their responsibilities and when milestones or deadlines are approaching.

- **Provide thoughtful feedback.** If a team member offers an opinion, even if you entirely disagree, internalize it, and consider it before completely banning it from thought. If you instinctively respond negatively to another team member's input, he or she may become defensive and respond negatively (out of instinct) to whatever you may offer to the group (being the team lead does not exempt or shield you from criticism).

- **Be a team player.** As the common phrase goes, "there is no 'I' in 'team.'" While every member of the team may not agree on a topic of discussion, weigh the good with the bad and decide what is best for the project.

- **Check attitudes at the door.** You will not like everyone you work with. Preventing your personal feelings from negatively affecting the cohesion of the group is vital to the success of the project.

## Communication Processes

There are two primary ways to communicate: orally and electronically. In this booming age of technology, communication is becoming more streamlined and efficient with new forms of electronic communication. However, electronic communication does not undermine or eliminate the necessity for old-fashioned oral communication.

There are many contingencies that dictate which communication processes to utilize within a collaborative environment. The following are examples of such contingencies:

- **Size** – If the team consists of only a few members, the method to communicate differs from a team of twenty or more members.

- **Location** – All members of a writing team will not always reside in the same location. Your team members' physical offices may consume multiple floors of a building. Alternatively, in an extreme situation, the team members may be spread across various offices located throughout the country, or even the world. Having team members scattered throughout the world presents the issue of different time zones, which in itself is an issue that may dictate the communication process that is utilized.

- **Resources** – The company's financial state or policies may dictate what communication resources are available. For example, the company may not have the financial or the worker resources to create a wiki. Or corporate policy could prohibit the use of Instant Messenger services. In these instances, face-to-face interaction or phone conversation may be the most effective means to communicate.

Different communication processes suit different situations. Do not use one process for all communication efforts.

## Electronic Communication

Many companies are adopting modern means to communicate, not only internally among co-workers, but also externally with clients. Software development companies, which are commonly within the circuit of using the latest technological gadgets, often make use of the most recent communication trends.

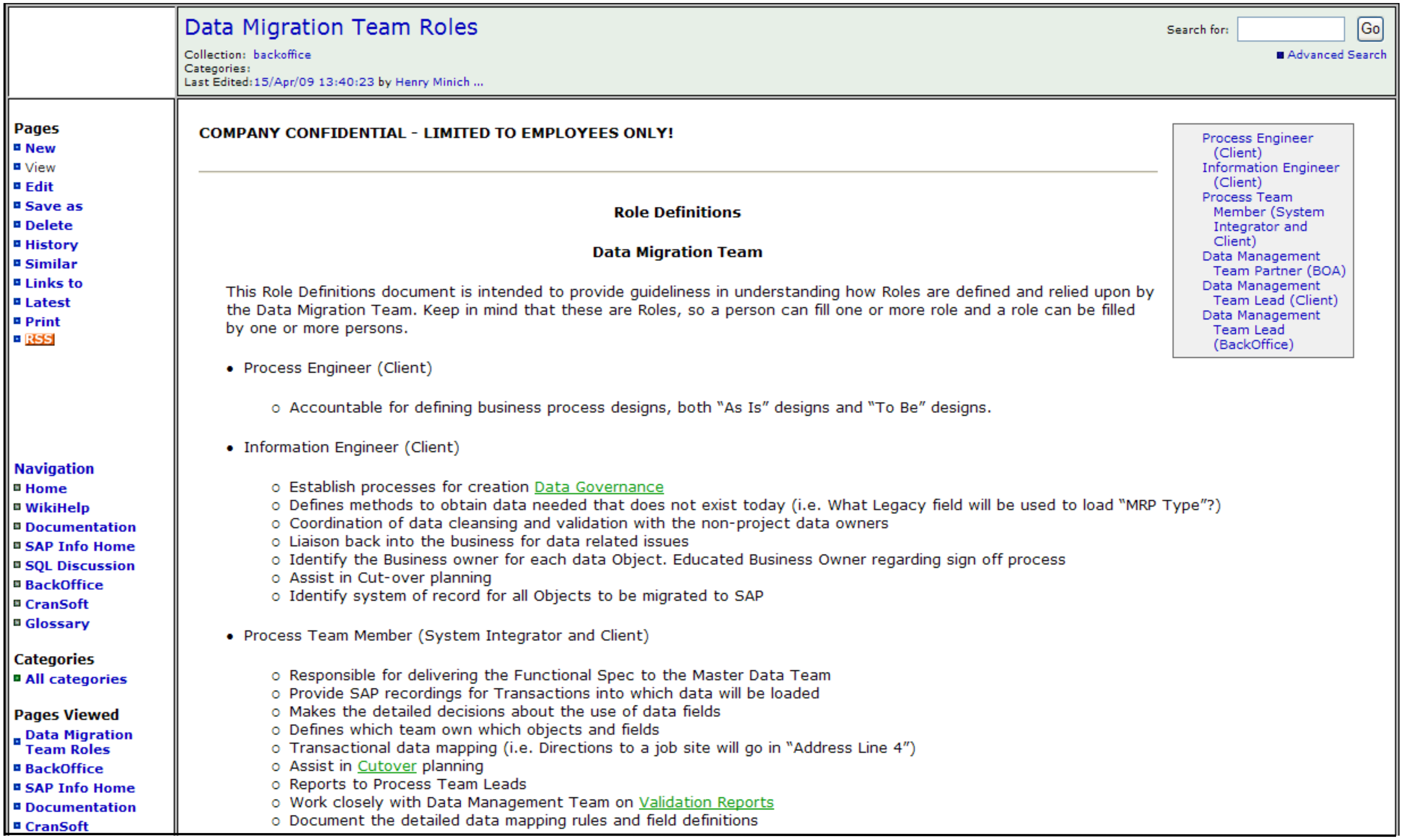
When collaborating with developers and other writers, there are many electronic collaborating tools that can facilitate communication. This textbook speaks specifically to the following electronic communication methods:

- WIKIs
- Email
- Instant Messaging
- Webinars

### Wiki

A wiki is an online encyclopedia for a community (in this sense, a corporation or a department) where rights are granted to members to edit, to delete, and to add content. WIKIs are part of the latest trend in collaborative writing and communication because they allow subject matter experts to generate content in order to educate others.  Due to the nature of a wiki, where anyone with access can add or modify content, WIKIs are generally for internal-use only.  The theory of WIKIs is that while anyone can contribute content, if incorrect information is presented, someone will fix it.

Figure 1 is an example of a wiki entry for a software development company’s internal wiki. Note that there is a date and time stamp along with the editor’s name at the top of the page to record information about the most recent changes to the page. This particular company chose to stress the importance of the wiki containing internal documentation by including a confidentiality line on every wiki page.



Data Migration Team Roles

Collection: backoffice
Categories:
Last Edited:15/Apr/09 13:40:23 by Henry Minich ...

**COMPANY CONFIDENTIAL - LIMITED TO EMPLOYEES ONLY!**

- Process Engineer (Client)
- Information Engineer (Client)
- Process Team Member (System Integrator and Client)
- Data Management Team Partner (BOA)
- Data Management Team Lead (Client)
- Data Management Team Lead (BackOffice)

**Role Definitions**

**Data Migration Team**

This Role Definitions document is intended to provide guideliness in understanding how Roles are defined and relied upon by the Data Migration Team. Keep in mind that these are Roles, so a person can fill one or more role and a role can be filled by one or more persons.

- Process Engineer (Client)
  - Accountable for defining business process designs, both "As Is" designs and "To Be" designs.
- Information Engineer (Client)
  - Establish processes for creation Data Governance
  - Defines methods to obtain data needed that does not exist today (i.e. What Legacy field will be used to load "MRP Type"?)
  - Coordination of data cleansing and validation with the non-project data owners
  - Liaison back into the business for data related issues
  - Identify the Business owner for each data Object. Educated Business Owner regarding sign off process
  - Assist in Cut-over planning
  - Identify system of record for all Objects to be migrated to SAP
- Process Team Member (System Integrator and Client)
  - Responsible for delivering the Functional Spec to the Master Data Team
  - Provide SAP recordings for Transactions into which data will be loaded
  - Makes the detailed decisions about the use of data fields
  - Defines which team own which objects and fields
  - Transactional data mapping (i.e. Directions to a job site will go in "Address Line 4")
  - Assist in Cutover planning
  - Reports to Process Team Leads
  - Work closely with Data Management Team on Validation Reports
  - Document the detailed data mapping rules and field definitions

Figure 1 – Wiki Example

### Benefits

A wiki is a great way to share knowledge across a company and to conduct research.

Not deviating far from its Hawaiian origin, “wikiwiki,” which means “quick” or “hurry” (Ebersbach 11), the wiki provides an expedited relay of information to an entire corporation. Users can publish information to the Internet without having prior programming experience.

Another benefit to using WIKIs is version control. When a change is made to an entry, the previous entry is archived, which makes the new version the “current” copy. WIKIs allow users to track who is contributing to the project as well as allow for peer review and editing.

Lastly, WIKIs provide multiple perspectives on a single topic and therefore encourage collaboration. For example, developers, with theoretical application, and consultants, with practical experience may view a feature differently. A technical writer can greatly enhance a document by pairing practical experience with theoretical application.

### Challenges

Because anyone with access to a wiki can modify the content, this useful collaboration tool can quickly lose its integrity. Therefore, having a moderator who constantly verifies the validity of the text is important to the success of a wiki.

Since individuals with a range of skill and experience levels contribute to the content of a wiki, this corporate encyclopedia can become an open forum for debate. Methodology, functionality, and usage of the program are hot topics of contention between software developers and consultants. In the situations that do not yield a “right” or “wrong” answer the wiki could become a battlefield between opinions. Serving as a repository of opinions is not a purpose or a goal of a wiki.

Being an information repository and a resource opinions are beyond the scope of a wiki.

Lastly, maintaining and updating content are additional challenges of WIKIs. As new software is developed, as features change, and as methodologies are created, someone must manually update the wiki. If the wiki moderator does not delegate maintenance responsibility to specific subject matter experts, information can quickly become out of date.

### *Applications*

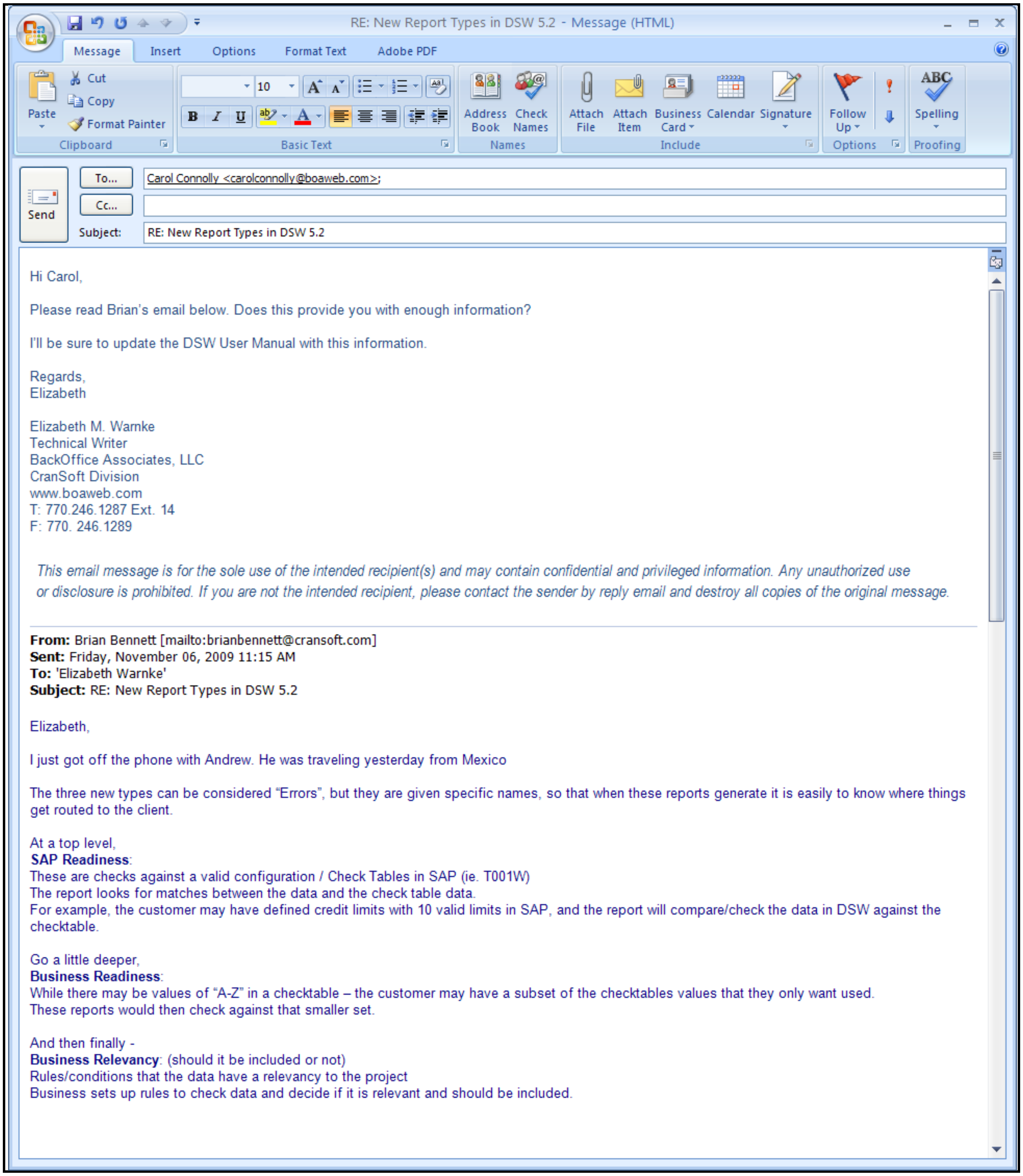
There are numerous applications of a wiki, many of which depend on how this form of communication is used within the company. In a software development environment, WIKIs are specifically helpful in the following situations:

- Serve as a database for nonstandard product documentation. Technical writers can use content to enhance, not replace, product documentation.
- Enable developers to share development tips and tricks learned through experience.
- Act as a blog or an open forum that could easily be rewritten to enhance existing or generate new product documentation.

## Email

Electronic mail (email) remains the most popular and widely used form of electronic communication. It enables letters and memos to be sent, literally, with the click of a button.

Figure 2 is an example of an email conversation. In this example, email is the preferred method of communication because terms are being defined and multiple individual are included in the conversation.



To... Carol Connolly <carolconnolly@boaweb.com>;

Cc...

Subject: RE: New Report Types in DSW 5.2

Hi Carol,

Please read Brian's email below. Does this provide you with enough information?

I'll be sure to update the DSW User Manual with this information.

Regards,
Elizabeth

Elizabeth M. Warnke
Technical Writer
BackOffice Associates, LLC
CranSoft Division
www.boaweb.com
T: 770.246.1287 Ext. 14
F: 770. 246.1289

*This email message is for the sole use of the intended recipient(s) and may contain confidential and privileged information. Any unauthorized use or disclosure is prohibited. If you are not the intended recipient, please contact the sender by reply email and destroy all copies of the original message.*

**From:** Brian Bennett [mailto:brianbennett@cransoft.com]
**Sent:** Friday, November 06, 2009 11:15 AM
**To:** 'Elizabeth Warnke'
**Subject:** RE: New Report Types in DSW 5.2

Elizabeth,

I just got off the phone with Andrew. He was traveling yesterday from Mexico

The three new types can be considered "Errors", but they are given specific names, so that when these reports generate it is easily to know where things get routed to the client.

At a top level,
**SAP Readiness**:
These are checks against a valid configuration / Check Tables in SAP (ie. T001W)
The report looks for matches between the data and the check table data.
For example, the customer may have defined credit limits with 10 valid limits in SAP, and the report will compare/check the data in DSW against the checktable.

Go a little deeper,
**Business Readiness**:
While there may be values of "A-Z" in a checktable – the customer may have a subset of the checktables values that they only want used.
These reports would then check against that smaller set.

And then finally -
**Business Relevancy**: (should it be included or not)
Rules/conditions that the data have a relevancy to the project
Business sets up rules to check data and decide if it is relevant and should be included.

Figure 2 – Email Example

## *Benefits*

Email enables people to communicate with ease and allows for a leisurely response. You can address multiple people and audiences at once or can keep an email conversion between two people.

Time-efficiency is the primary drive to use email as a communication medium. Emails can be sent within seconds and from computers as well as from phones (making email not only efficient, but portable as well).

Email also provides a virtual paper trail. All emails, both sent and received, are recorded and can be used for future reference. The date and time is stamped on emails to provide an instant record history.

### *Challenges*

Like any medium of electronic communication, email is dependent on the Internet. If the email provider is not functioning, email cannot be utilized as a reliable means of communication. This dependency on the Internet can become a problem because with the efficiency of email, users can be detrimentally reliant on this medium. If email becomes unavailable, users may feel like they have lost their only source of communication.

In addition, with the high volume of spam email, there is the chance that email will get lost or will not send properly. Not all email providers offer a way to verify the target audience received the email and that the information was relayed. A sent email does not mean the intended recipient received it. Unlike the wiki, where the target audience is known (i.e. all users with security access to the wiki), there is zero control of what happens to the email once it is sent. The email could be forwarded or printed and distributed to others, completely against the wishes of or unknown to the author.

### *Applications*

Despite the negative attributes to email, it is the primary source of communication in a software development company. For a technical writer, email is an efficient way to communicate with developers and writers. The following are common applications of email as a means to communicate:

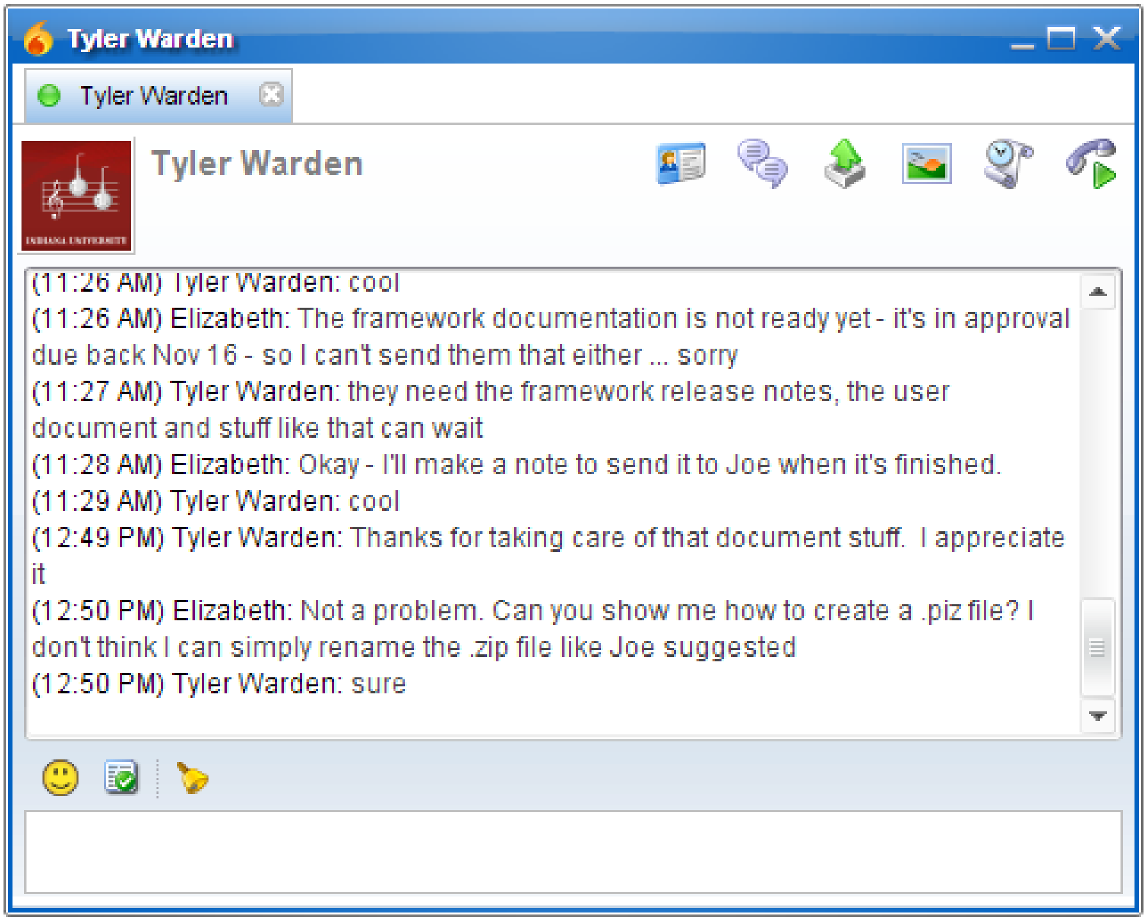
- Confirm the exact spelling of a product or phrasing of a sentence.
- Interview developers.
- Share files.
- Review and approve documents.

## Instant Messenger

Instant Messenger (IM) is a new trend, not just in technical communication, but in communication in general. IM services enable the continuous flow of communication in a conversational format.

IM services will never replace email or live interaction, but they do provide an efficient alternative. IM enables users to carry on multiple “conversations” at once while completing other tasks, such as answering emails or editing a document. It also provides the ability to have chat rooms (group conversations) with multiple co-workers without tying up webinar or phone lines, while at the same time documenting the meeting.

Figure 3 outlines an example of a brief instant messaging conversation between two co-workers. Notice that the conversation is informal and short. This conversational style of communication enables information to quickly be relayed so participants can resume their work.



(11:26 AM) Tyler Warden: cool
(11:26 AM) Elizabeth: The framework documentation is not ready yet - it's in approval due back Nov 16 - so I can't send them that either ... sorry
(11:27 AM) Tyler Warden: they need the framework release notes, the user document and stuff like that can wait
(11:28 AM) Elizabeth: Okay - I'll make a note to send it to Joe when it's finished.
(11:29 AM) Tyler Warden: cool
(12:49 PM) Tyler Warden: Thanks for taking care of that document stuff. I appreciate it
(12:50 PM) Elizabeth: Not a problem. Can you show me how to create a .piz file? I don't think I can simply rename the .zip file like Joe suggested
(12:50 PM) Tyler Warden: sure

Figure 3 – Instant Messenger Example

## *Benefits*

IM is a fast way to converse with multiple people and allows for real-time, text-based communicating. The formality of email is lifted, which enables the participants to communicate in a directed and expedited fashion. *Fortune* magazine claims "instant messaging is rising fast in corporate America" due to its expedited flow between individuals (Varchaver 102). Faster communication between co-workers and clients increases productivity.

In addition to communicating in real time, IM increases productivity because it enables users to multi-task. Users can communicate with others while at the same time, continue to work on the current project. IM began as a utility for social communication. The systems architect in the Office of Information Technology at Duke University, Mark McCahill comments, “chat is moving towards not just what you do with your closest, best friends but a way to work with all your collaborators. If you know someone is awake and potentially available for conversation, you can probably do a chat instead of sending an email” (McCahill).

Another clear benefit of IM is the ability to overcome a language barrier. Not every individual you communicate with is a native English speaker. Since IM is a conversation-based way to communicate, it enables participants to speak in a less formal format and it encourages collaboration. IM alleviates the tension and the pressure to “speak” well in written communication.

### *Challenges*

Since instant messaging services began as a means for social communication, turning a work-related conversation into an open forum for social dialog is a natural digression. The informality and conversational style of communication yields to social interaction, which could, in turn, lower the productivity level.

IM can possibly become *too* conversational, to the extent that users may depend on it entirely instead of orally communicating. There is great value in oral communication that can be lost if IM is used as its replacement.

### *Applications*

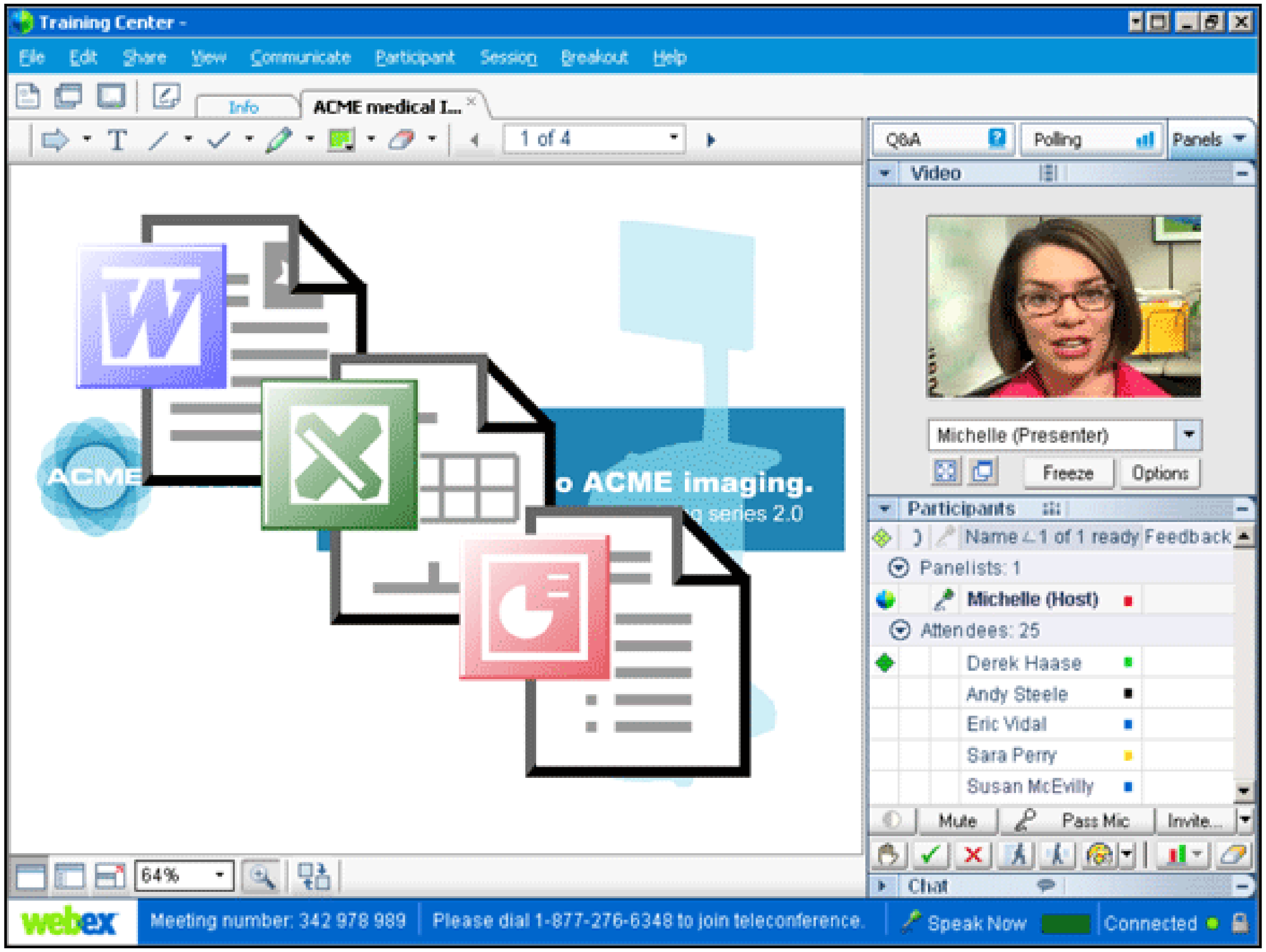
IM is a great tool if you are looking for a quick answer to a question or if you are pressed for time and cannot depend on email. IM is not intended for conveying paragraphs of information to be communicated – that is what email is for.

## Webinar

Web seminars or conferences, *Webinars*, (also known as WebExs) are a new trend in electronic communication. Since it is common to have co-workers reside in offices around the country (or even around the world), webinars enable conferences and meetings involving many participants to be conducted without having everyone physically in the same room.

Webinars include phone conferencing as well as online presentations facilitated by a single user with many participants.

Figure 4 is an example of a webinar. Notice how easy a meeting can be facilitated by using Web cameras, seeing who is online, and presenting information electronically.



Panelists: 1

Michelle (Host)

Attendees: 25

Derek Haase

Andy Steele

Eric Vidal

Sara Perry

Susan McEvilly

Meeting number: 342 978 989

Please dial 1-877-276-6348 to join teleconference.

Figure 4 – Webinar Example

### Benefits

Some webinar services allow for virtual face-to-face interaction. Most, however, allow for presentations and projects to be shared over the Internet to an invited audience. Webinars provide flexibility and convenience that is not always available through other avenues of communication. Employees can communicate globally without leaving their office, which is a huge financial benefit.

### Challenges

Like other avenues of technical communication, webinars are dependent on the Internet. If a connection is lost mid-presentation, everyone is disconnected from the session. Reconnecting to the meeting could waste valuable minutes; in extreme conditions, the meeting may need to be rescheduled.

The setup of webinars is never as seamless a process as it appears. Individuals with webinar experience know how to set up a webinar (i.e. invite participants and prepare a presentation) and how log into a session. However, since webinars are a relatively new invention, most participants have never facilitated or participated in a webinar.

### Applications

In a software development company, there are many ways webinars can be utilized to increase the flow of communication and to share knowledge between co-workers. As the technical writer, you may be asked to conduct the following tasks via webinars:

- Demonstrate applications to the sales and marketing departments. These departments rely heavily on the developers and technical writers to communicate information about new products.

- Expand training. Training documentation can easily be turned into a PowerPoint presentation to be used in online training.

- Facilitate the documentation process between developers and other writers who may be located in various offices.

- Collaborate on documentation. One writer can share a desktop, and therefore share the document with other writers. Collaboration on content, organization, and grammar is therefore easily facilitated.

## Oral Communication

There is something to be said for the old-fashioned use of oral communication. Taking advantage of the electronic means that expedite and streamline communication is an easy task. But there are many elements that oral communication – face-to-face interaction as well as phone conversations – can offer that are unattainable by electronic communication.

### Benefits

There are various benefits oral communication has over electronic. For starters, oral communication provides the opportunity for immediate feedback. With electronic services, such as email or instant messaging, if the user is not in the presence of his or her computer, you may not get the immediate response that you are looking for. Oral communication eliminates the ambiguity of if the message was delivered or understood.

In addition, oral communication provides the opportunity for all participants in the conversation to utilize nonverbal cues, such as body language and inflection. You can read each other's expressions (both facial and body), which may provide insight on your understanding or immediate reaction to the information being communicated. Listening to someone's voice provides tone, stress, and emphasis, which is often difficult to convey through electronic communication processes.

### Challenges

While oral communication has its benefits, it also has a few detriments. Unlike electronic communication, where you can edit and reword what you want to say, oral communication is immediate and often impulsive. Once you say something, there is no "delete" button to take it back. You have to be very careful about what you say and be aware of any unintentional body language.

Oral communication requires social skills, which not everyone has. Poor speakers can often confuse the people on the receiving end. For a technical writer, oral communication requires strong interviewing skills. When talking to subject matter experts, the researching and drafting processes are expedited when  you know the type of questions that will generate answers with enough information for you to write expedites the drafting process.

### Applications

As a technical writer in a software development company, you communicate often with the developers to obtain the information you need in order to produce deliverables. Since documentation falls to the bottom of the priority list for many developers, waiting to hear back from a developer via email can take weeks or you could even be ignored through instant messenger. Oral communication can be the most efficient and effective way to obtain the information you need. With new versions of software being released often – sometimes every month – you are always working against the clock to meet a deadline and may not have weeks or even days to wait for a developer to respond.

Even though you are a writer, some topics are easier to explain in person than in writing. Or maybe you cannot put into words what you want to write. Oral communication provides a means to talk through your writer's block, even if you are communicating aloud with yourself.

### Oral versus Electronic Communication

The Table 1 provides a cross reference of the benefits, the challenges, and the applications of electronic and oral communication processes:

| Source | Benefits | Challenges | Applications |
|---|---|---|---|
| **Electronic Communication Processes** | | | |
| **Wiki** | • Conducts research<br>• Provides version control<br>• Encourages collaboration | • Has potential to misrepresent information<br>• Yields constant evaluation for up-to-date content<br>• Has potential to quickly become an open forum of opinions | • Enhances documentation through database research<br>• Provides a forum for product tips and tricks |
| **Email** | • Communicates at leisure<br>• Addresses multiple audiences<br>• Provides time efficiency<br>• Provides a virtual paper trail | • Yields difficulty to reflect tone | • Conducts interviews with developers<br>• Provides means for writers to collaborate.<br>• Enables files to be shared within the writing process<br>• Allows deliverables to be distributed to the end user |
| **Instant Messenger** | • Encourages multi-tasking<br>• Provides a continuous flow of conversation<br>• Is self documenting | • Promotes communication to be *too* conversational and informal<br>• Has potential to replace oral communication | • Helps users looking for a fast and brief answer<br>• Enables any two people to hold a conversation |
| **Webinars** | • Connects individuals across the globe<br>• Provides efficiency | • Has a difficult setup process | • Provides a forum for live meetings with developers and writers<br>• Recycles information to be used in training material<br>• Collaborate on a document |

| Source | Benefits | Challenges | Applications |
|---|---|---|---|
| **Oral Communication Processes** | | | |
| **Face-to-face and Phone** | • Gives immediate feedback<br>• Senses tone and nonverbal cues | • Cannot edit or delete speech<br>• Requires social skills | • Provides means to interview developers<br>• Enables writers to bounce ideas off each other |

Table 1 – Oral versus Electronic Communication

## Chapter Summary

The following key points have been covered in this chapter:

- Collaboration with developers and other writers is essential for technical writers to produce well-crafted documentation.
- Collaborating on a project yields many benefits as well as some challenges to the writing process.
- Each medium – wiki, instant messenger, email, webinars, face-to-face interaction, and phone conversation – is useful for particular situations.

# Chapter 4: Deliverables

## Chapter Contents

## Chapter Objectives

The following are objectives of this chapter:

- To provide a firm understanding of the basic deliverables produced by technical writers in a computer software company.
- To explain the primary goal of any deliverable.
- To recognize the audience and scope of each deliverable.

The deliverables produced by technical writers in the computer software industry vary. This chapter outlines the most common deliverables in this industry and is limited to the following:

- User Manuals
- Release Notes
- Product Descriptions
- Training Manuals
- Quick Reference Guides

While this textbook does not describe an exhaustive list of deliverables, it provides the essential document types.  While each deliverable type is written for a single software product, each serves a distinct purpose and caters to a different audience.

## User Manuals

The user manual is the most widely used document type by clients to learn about a software product and therefore consumes the majority of a technical writer's time. The user manual is essential for every piece of software because it describes and explains every feature and potential use of the program. This is the first piece of documentation end users reach for when using and learning about a new piece of software.

### The Audience

User manuals have the widest audiences of all deliverables produced by a technical writer in the IT industry. When writing for developers and clients of all technical skill levels, targeting a user manual towards a single audience is virtually impossible. For this reason, user manuals are written to cater to a broad range of audiences by including various sections that each considers a smaller group of audiences or even a single audience.

For example, the instructional steps are aimed at less technical audiences, since the steps literally walk the user, step-by-step, through a function. A technical and more advanced user does not necessarily need this level of detail to understand a function.

The Reference section, however, is written for all audiences. This section defines all objects (menus, pages, buttons, etc.) within the software program and is written with a medium level of detail. The detail is sufficient for a technical and a semi-technical audience, and if a non-technical audience requires more information, the audience can use the index to search for information on a specific function, concept, or word.

## The Scope

When deciding on the content of the user manual, ask yourself the following questions:

- What does this product do?
- Why would anyone use it?
- What are the major features?
- How does this product fit in with the suite of inter-dependent applications or with other products?

While there are countless ways to organize a user manual, the following is a recommended list of sections to include:

- Table of Contents
- Overview
- Additional Resources
- Getting Started
- Tutorials
- Reference
- Appendices
- Index

### Table of Contents

The Table of Contents (TOC) is a very important aspect of the user manual. Users rarely read a manual cover to cover because they are only interested in the information relevant to them at that moment. The TOC is used to search through the document to find information on a topic or the answer to a question.

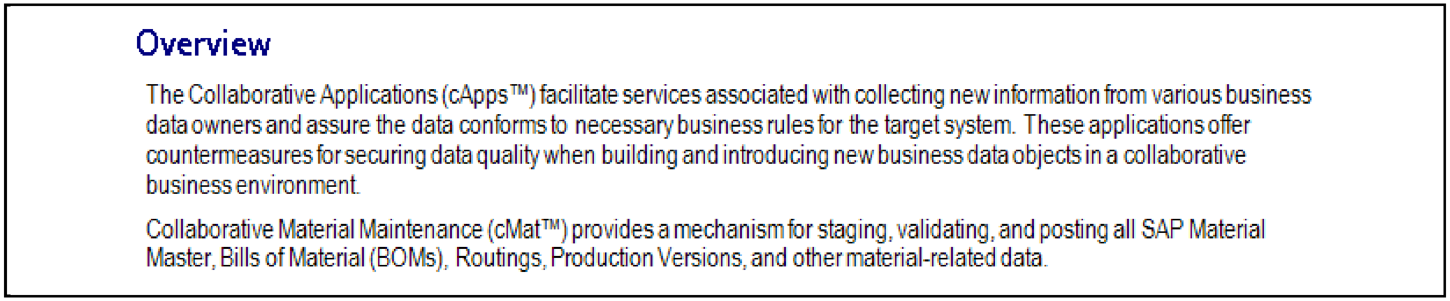
### Overview

The Overview section provides a high-level description of the product and is no more than a few sentences. If the overview cannot be written in less than five sentences, the writer probably does not know enough about the product and should conduct further research before attempting to write the remainder of the manual. The Overview section provides the reader with an understanding of what the product is, what benefits it provides, and how it applies to the reader’s job function.

The Overview can also include any prerequisite information. If the user is expected to be at a certain technical level before completing the manual tutorials or if the product is dependent on other products, the Overview is the best section to include the requirements.

Indicate how (if applicable) the product fits in with a toolset or if/how it depends on other products. If it is part of an entire set, include the other products within the toolset and a brief description of the purpose each supplementary or dependent application serve. In addition, mention if the product is able to function without the installation of other products.

Figure 5 provides an example of the Overview section within a user manual for a product called *cMat™*. cMat™ is part of a toolset called the *Collaborative Applications*.



## Overview

The Collaborative Applications (cApps™) facilitate services associated with collecting new information from various business data owners and assure the data conforms to necessary business rules for the target system. These applications offer countermeasures for securing data quality when building and introducing new business data objects in a collaborative business environment.

Collaborative Material Maintenance (cMat™) provides a mechanism for staging, validating, and posting all SAP Material Master, Bills of Material (BOMs), Routings, Production Versions, and other material-related data.

Figure 5 – Overview Section of a User Manual Example

## Additional Resources

If a product is so complex it requires several manuals, perhaps a training manual, a manual dedicated to a single feature, or the installation manual, provide a list of additional resources in the user manual.

## Getting Started

If a configuration manual is not created for the product, include information to guide the user through what needs to be set up and configured before the product can be used.

## Tutorials

The tutorials are the primary focus of a user manual. These sections are often the reason the user is reading the manual in the first place. Tutorials are organized by core tasks or features of the product.

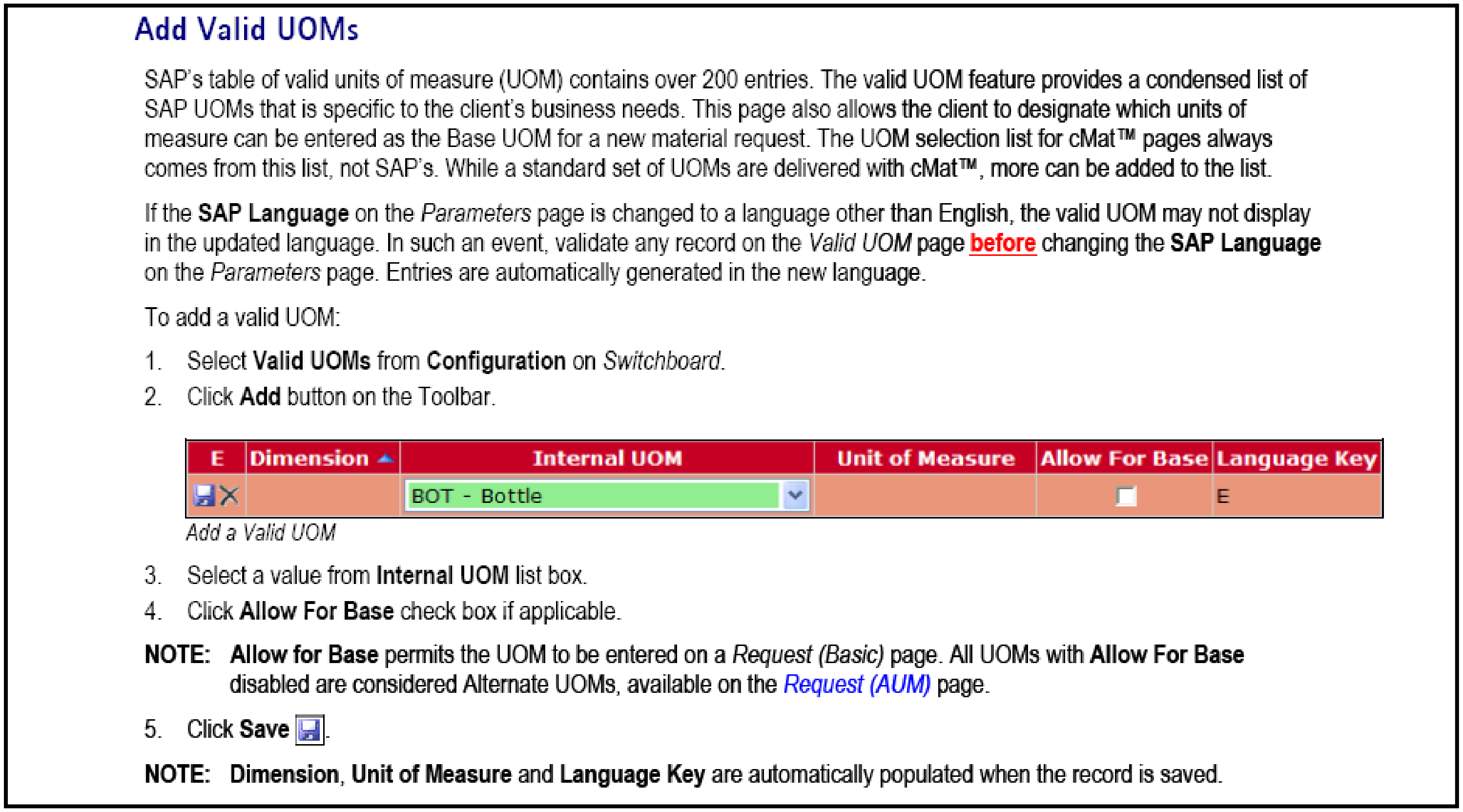
Provide a brief introduction to the feature or task. Is the task part of a larger task or function? How does the task relate to the general scope of the product? State the purpose of the process or the task. Readers tend to successfully complete the task on the first try when they have a general understanding of the entire process and the purpose (Houp 16). An overview of the process also enables the readers to recognize at the beginning of the tutorial if they make a mistake when completing the task, which in the end, will save them a tremendous amount of time.

Tips for Writing Tutorials

1. Use a "To" line to introduce the tutorial, such as *To create a page:*.
2. Number steps.
3. Place steps in chronological order.
4. Describe the object (feature, field name, page name, etc.) exactly as it appears in the software.
5. Include screen shots of the program populated with data.
6. Begin each step with a present tense verb.
7. Include one step per line.

After the tutorial is written, test the steps. Verify the steps are in the correct order, the object names are exactly as they appear in the program, etc. Since testing your own material is often a difficult task (since you wrote it, you therefore are very familiar with the prose) have another writer walk through the manual. After numerous renditions of editing and revising, the smallest errors could easily go unnoticed by the original author.

Figure 6 is an example of a tutorial within the cMat™ User Manual. Notice how this example showcases each tip for tutorial writing. The reader is provided with a purpose of the overview and is marched, step-by-step, through the function. The screen shot provides an example of data as well as validates that the user is on the correct page and performing the correct steps.



## Add Valid UOMs

SAP's table of valid units of measure (UOM) contains over 200 entries. The valid UOM feature provides a condensed list of SAP UOMs that is specific to the client's business needs. This page also allows the client to designate which units of measure can be entered as the Base UOM for a new material request. The UOM selection list for cMat™ pages always comes from this list, not SAP's. While a standard set of UOMs are delivered with cMat™, more can be added to the list.

If the **SAP Language** on the *Parameters* page is changed to a language other than English, the valid UOM may not display in the updated language. In such an event, validate any record on the *Valid UOM* page **before** changing the **SAP Language** on the *Parameters* page. Entries are automatically generated in the new language.

To add a valid UOM:

1. Select **Valid UOMs** from **Configuration** on *Switchboard*.
2. Click **Add** button on the Toolbar.

| E | Dimension ▲ | Internal UOM | Unit of Measure | Allow For Base | Language Key |
|---|---|---|---|---|---|
| 💾 ✕ |  | BOT - Bottle |  | □ | E |

*Add a Valid UOM*

3. Select a value from **Internal UOM** list box.
4. Click **Allow For Base** check box if applicable.

**NOTE:** **Allow for Base** permits the UOM to be entered on a *Request (Basic)* page. All UOMs with **Allow For Base** disabled are considered Alternate UOMs, available on the *Request (AUM)* page.

5. Click **Save** 💾.

**NOTE:** **Dimension**, **Unit of Measure** and **Language Key** are automatically populated when the record is saved.

Figure 6 – Tutorial of a User Manual Example

## Reference

The Reference section is a very important attribute to the user manual because it helps to make the manual appeal to a wide range of audiences. While technical audiences may not find the tutorials useful (because the exercises may be too basic) the Reference section provides them with a reason to read the user manual. This section of the manual provides descriptions of every function, feature, and control of the product.

Exclude any database design or information that is too technical and out of scope for the intended audience. The Reference section serves more as a glossary or a dictionary of terms used within the software product and the documentation than a methodology on programming.

The following are items to include in the Reference section:

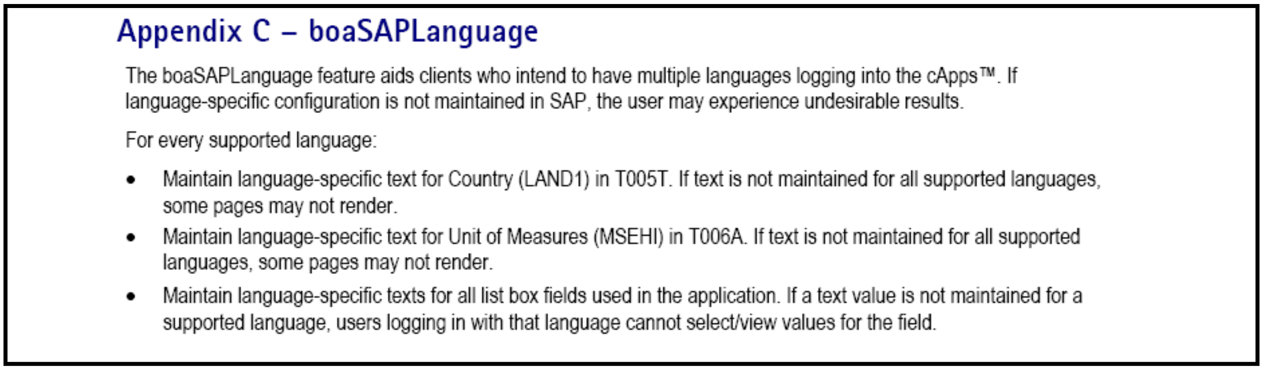
- Menus, sub-menus, and menu options
- Buttons and/or icons on the toolbar
- Pages with a brief overview of their function and purpose
- Controls, including buttons, field, and icons

## Appendices

Appendices contain additional or supplementary information that does not fit into another section of the manual. The following are examples of information within an Appendix:

- Troubleshooting tips
- Frequently asked questions
- Additional steps that do not pertain to the “average” audience, but are nonetheless important to include.
- Additional sections that are important to the product, but do not fit into a tutorial.

Figure 7 demonstrates good use of an appendix. This appendix provides additional information that is pertinent to cMat™, but does not fit into a tutorial.



## Appendix C – boaSAPLanguage

The boaSAPLanguage feature aids clients who intend to have multiple languages logging into the cApps™. If language-specific configuration is not maintained in SAP, the user may experience undesirable results.

For every supported language:

- Maintain language-specific text for Country (LAND1) in T005T. If text is not maintained for all supported languages, some pages may not render.
- Maintain language-specific text for Unit of Measures (MSEHI) in T006A. If text is not maintained for all supported languages, some pages may not render.
- Maintain language-specific texts for all list box fields used in the application. If a text value is not maintained for a supported language, users logging in with that language cannot select/view values for the field.

Figure 7 – Appendix of a User Manual Example

## Index

An index is important to a user manual for reasons similar to the table of contents. Indexes and cross-references increase the usability of the document by facilitating the process of finding information. Specific words (key words that the average audience searches on) are marked and then referenced according the page or pages the topic appears on.

### Helpful Hints for Writing User Manuals

- Make each topic heading a present tense verb, such as “Manage Security” or “Build Source Tables.” Users utilize the Table of Contents to search for a task to complete.

- Realize that defining each and every field or column may not be possible.  A constructive developer makes the field or column names descriptive enough or includes online page help.

- Take screen shots of fields populated with data (and not data that is obviously test data). Not only does the screen shot enable the user to follow along in the product with the manual, but it provides a clear example (without having to state an example in the manual text).

# Release Notes

Release notes is one of the most important deliverables produced by a technical writer. This deliverable outlines the differences between two versions of a single piece of software. The purpose is to educate the target audience on what the current release contains in comparison to the previous release.

Release notes are usually the first piece of documentation written for a release. This document type helps to define the scope of the remaining documentation for a specific product release.

## The Audience

The primary audience of release notes is an existing client. When a client is looking to upgrade to the latest version of software, its biggest concern is justifying the need to upgrade. What benefits does upgrading bring to the client? Release notes holds the answer to this question.

The secondary audience of release notes is the company's sales and marketing departments. In order to sell a client the upgrade package, the sales department must be aware of the new features that could be the main selling points in the sale. While release notes also include resolved issues (also known as "bug fixes"), the sales department must refrain from advertising them to the client. Everyone knows that all software contains bugs, but there is no need to broadcast their existence to the client.

The marketing department also benefits from release notes in order to tailor the marketing material to the target audience. When a new feature that is beneficial to a broad range of clients is introduced, release notes help the sales department close its sale by incorporating the features of the new release into marketing brochures, trade shows, and demonstrations.

Because the reader of release notes is typically not very technical, the release notes is written in a way that a semi-technical person can understand. Although, the audience must still understand the software in order to appreciate and gain valuable information from the release notes.

Release notes outline the differences between two versions of software, so it is not provided for the initial release of the software program. Product descriptions outline a high-level overview of the product and are used as the first line of selling the original release of a product to a client.

## The Scope

Release notes is organized by topics to enable the audience to navigate quickly through the document. While release notes can contain numerous sections, the following are the four basic sections that all release notes include:

- New Features
- Enhancements
- Resolved Issues
- Known Issues

Additional sections can be included, depending on company policies. These four are the primary sections that align with industry standards.

Release notes do not contain instructional steps to complete a task, too many details on *why* the change was made (because the client does not necessarily care), instructions on implementing a new feature, client-specific information (such as the client site where a bug was discovered), etc. Be brief and to the point. The following is a list of additional information to include in the release notes document:

- The release date
- The product version
- The previous product version (because not all upgrades are the next sequential number)
- Any dependent applications with required version numbers

The following are optional sections for release notes (but are beyond the scope of this textbook):

- **Installation** – Outlines instructions for installing the software. This section could also include new information required for the current installation of the product, but that information is most likely detailed in the Installation Manual. If a separate installation manual exists for the product, the Installation section may not be necessary in the release notes.

- **System Requirements** – Indicates any new system requirements for the current version of the product. For example, if the software migrates from SQL Server 2000 to SQL Server 2005, 2005 is the recommended operating system because the company may no longer support SQL Server 2000. Information like this is very useful in the release notes, as well as in a system requirements document.

- **Frequently Asked Questions (FAQs**) – Provides a hypothesized list of questions (and answers) most likely to be posed by the intended audience.

Avoid being too technical when writing release notes. Release notes is designed for clients (a non-technical audience), so details about database design, for example, are beyond the scope of release notes.

### New Features

New Features is the most important and sought-out section of release notes. It enables the client to decide if upgrading to the most recent software version is necessary or relevant. Since this is the most applicable section of the release notes, it is usually towards the beginning of this document type.

The New Features section includes items that are new to the product – functions that did not exist in the previous version. The following are examples of new features:

- New button for a specific page
- New piece of functionality
- New field name

Include information about why the feature is useful and applicable. For example, if there is a new Web page in the software, indicate what function it serves and why this new feature appeals to a user.

### Enhancements

Enhancements usually follow the New Features section. The Enhancements section includes changes made to existing features. When writing enhancements, include a justification for the change. The reader must be able to clearly understand why this change in the software is beneficial the client's business needs.

The following are examples of enhancements:

- Change in how a feature works
- An item, such as a page, a button, an icon, etc. removed from the product
- Modifications to increase performance

When writing an enhancement element for release notes, make sure the entry is truly an enhancement. Indicate what feature or function has been extended and why the enhancement is important.

### Resolved Issues

Resolved Issues are items that were problematic in the previous version but are fixed in the current version. Some issues are reported by clients and either are imperative enough to result in a patch or were deferred for in the next release. And some items were knowingly problematic in the previous release, but were resolved in the current release.

Since release notes document the difference between released versions of software, differenciating between bugs that existed in the previous version and bugs that were generated during development is an important element to this section. (The distinction can be provided by the project manager or a developer.) While developing a new feature, an existing feature can break or may need to be modified to support the new feature. Software bugs like this example are not be included in the release notes.

The following are examples of resolved issues:

- Features that did not work as designed
- Buttons or icons that nagivated to the incorrect page
- Pages that did not render when navigated to

Include a solution to the resolved issue. The reader wants to know what about the issue did not work correctly in the previous release and what was done to resolve the issue. This is important information to include because the bug could have been fixed by the installation or upgrade of a third-party program (such as TaxWare or an antivirus program). Or the bug could have been caused by a bug in such a third-party application. In this instance, the client has to resolve the issue with the third-party software company.

### Known Issues

Known issues are bugs in the software that the company knows exist when the software is packaged. The typical reason for known issues is a time restriction. If the release date is quickly approaching and the software developers do not have time to fix a bug that was unveiled during testing, the software may get released with the bug. If the bug is critical and detrimental to the software, the project manager's best interest is to delay the release in order to fix the bug. Nonetheless, in a realistic setting, a bug could exist in the new release.

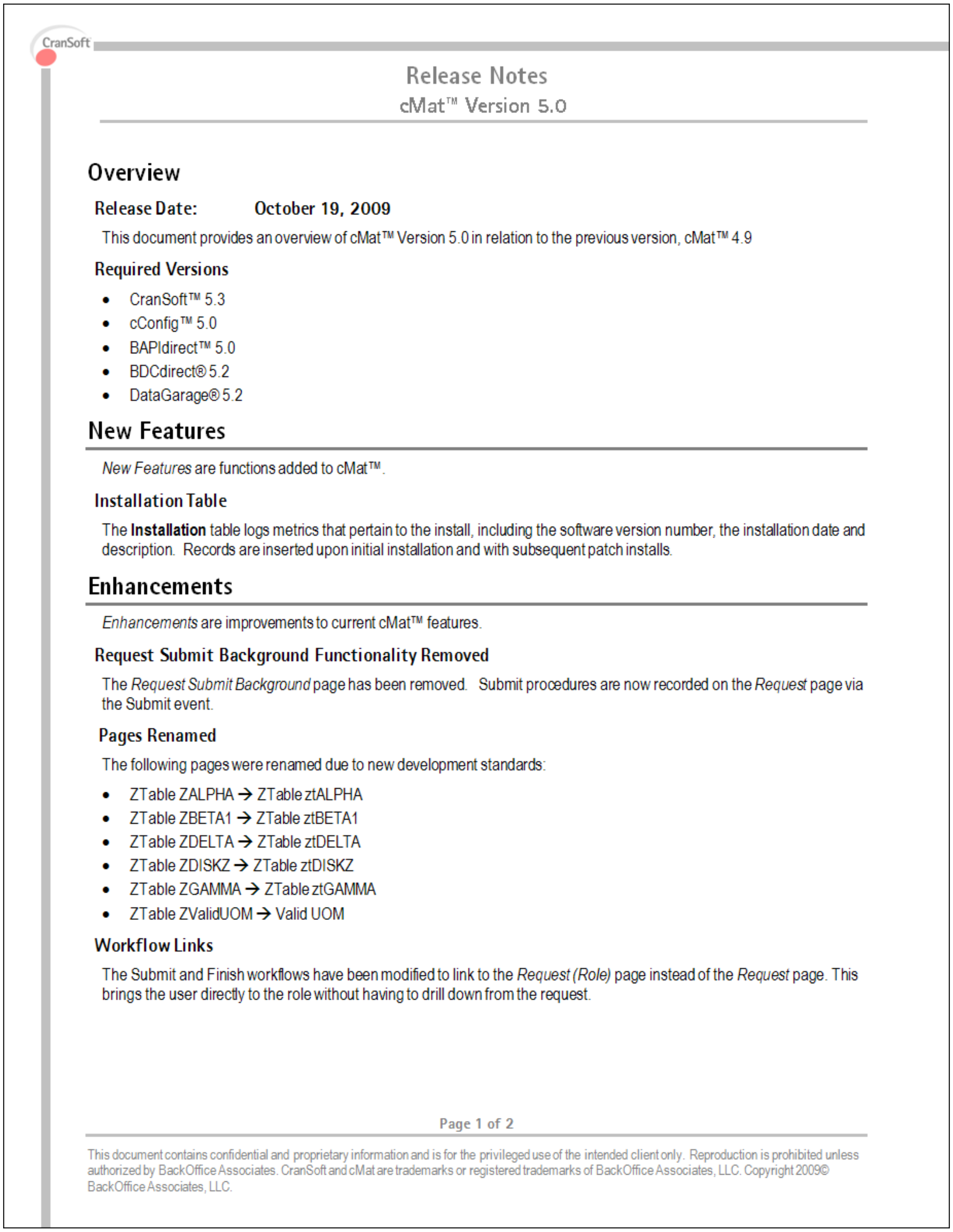
Even though a Known Issues section could make the software look faulty, this section is beneficial for two primary reasons:

1. Provides the client with issues that may or may not impact its system.
2. Gives the software company great credibility. The client appreciates being forewarned of existing bugs .

Verify that a Known Issue in one release becomes a Resolved Issue in the next. A client who frequently upgrades to the latest software release wants to see the known issue fixed before upgrading to the next release. This information in turn builds trust, confidence, and a positive rapport between the client and the software development company.

Figure 8 is an example of release notes for the 5.0 release of cMat™. It contains the sections that are applicable to the release: New Features, Enhancements, and Resolved

Issues. Elements listed as Resolved Issues were either Known Issues in the previous release or were bugs reported by clients with the previous release.



CranSoft

## Release Notes

cMat™ Version 5.0

## Overview

**Release Date:** **October 19, 2009**

This document provides an overview of cMat™ Version 5.0 in relation to the previous version, cMat™ 4.9

### Required Versions

- CranSoft™ 5.3
- cConfig™ 5.0
- BAPIdirect™ 5.0
- BDCdirect® 5.2
- DataGarage® 5.2

## New Features

*New Features* are functions added to cMat™.

### Installation Table

The **Installation** table logs metrics that pertain to the install, including the software version number, the installation date and description. Records are inserted upon initial installation and with subsequent patch installs.

## Enhancements

*Enhancements* are improvements to current cMat™ features.

### Request Submit Background Functionality Removed

The *Request Submit Background* page has been removed. Submit procedures are now recorded on the *Request* page via the Submit event.

### Pages Renamed

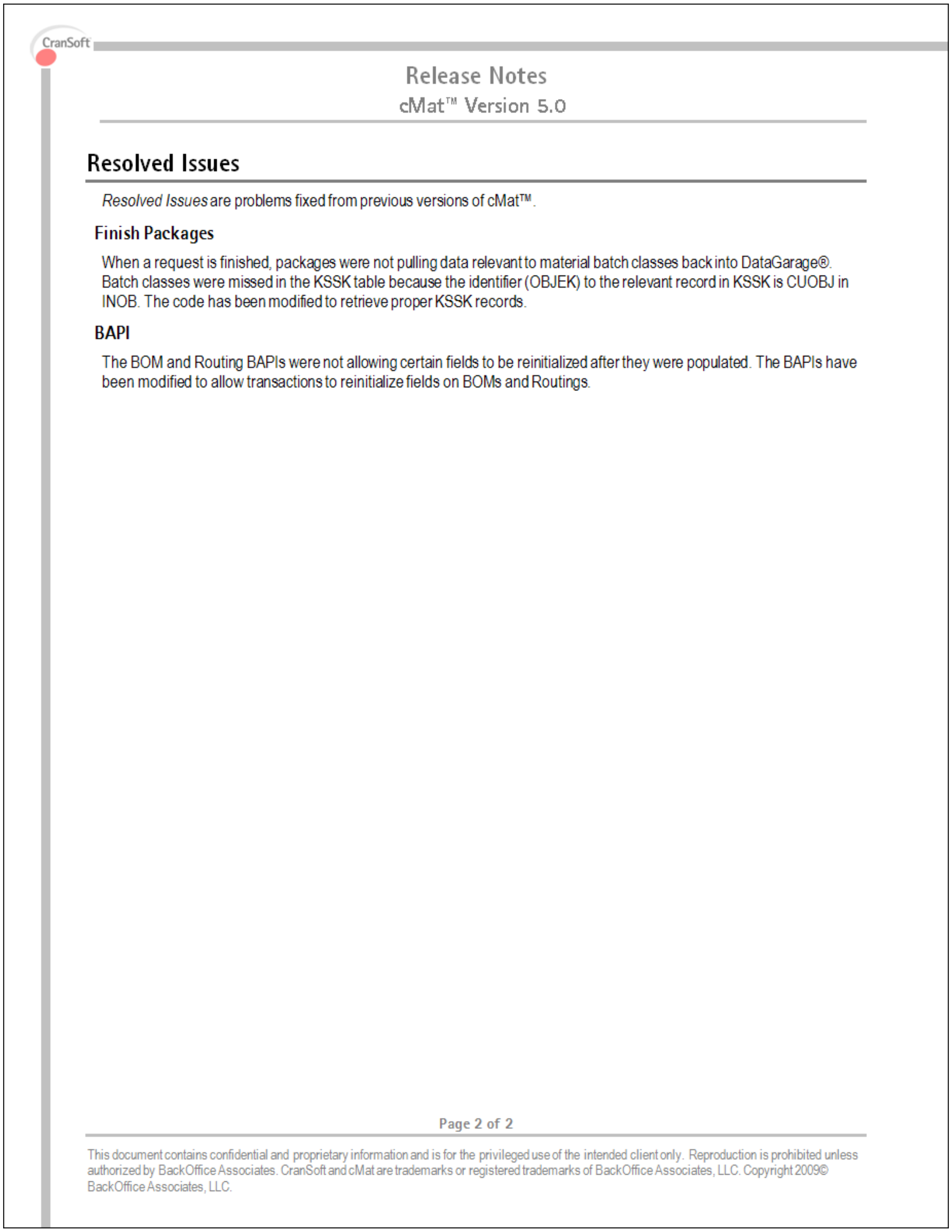
The following pages were renamed due to new development standards:

- ZTable ZALPHA → ZTable ztALPHA
- ZTable ZBETA1 → ZTable ztBETA1
- ZTable ZDELTA → ZTable ztDELTA
- ZTable ZDISKZ → ZTable ztDISKZ
- ZTable ZGAMMA → ZTable ztGAMMA
- ZTable ZValidUOM → Valid UOM

### Workflow Links

The Submit and Finish workflows have been modified to link to the *Request (Role)* page instead of the *Request* page. This brings the user directly to the role without having to drill down from the request.

This document contains confidential and proprietary information and is for the privileged use of the intended client only. Reproduction is prohibited unless authorized by BackOffice Associates. CranSoft and cMat are trademarks or registered trademarks of BackOffice Associates, LLC. Copyright 2009© BackOffice Associates, LLC.



CranSoft

Release Notes
cMat™ Version 5.0

## Resolved Issues

*Resolved Issues* are problems fixed from previous versions of cMat™.

### Finish Packages

When a request is finished, packages were not pulling data relevant to material batch classes back into DataGarage®. Batch classes were missed in the KSSK table because the identifier (OBJEK) to the relevant record in KSSK is CUOBJ in INOB. The code has been modified to retrieve proper KSSK records.

### BAPI

The BOM and Routing BAPIs were not allowing certain fields to be reinitialized after they were populated. The BAPIs have been modified to allow transactions to reinitialize fields on BOMs and Routings.

**Figure 8 - Release Notes Example**

# Product Descriptions

Product descriptions are among the shortest and the most condensed documents a technical writer creates. However, a brief document does not mean one that is easy to write.  Product descriptions provide a high-level overview of the product and often contain a list of features and flow diagrams of the product's process (and purpose).

The purpose of product descriptions is to provide potential clients with enough information to make an informative decision of whether or not to purchase the software based on how the product fits into its business needs. Sales and marketing departments often leverage the content from product descriptions in their presentations to clients.

## The Audience

The primary audience of product descriptions is potential clients. When learning about a product and evaluating if or how the target company will benefit from purchasing and utilizing the product, a high-level overview is the best place to start. If the client wishes to learn more about the product, other documents are available. If, after reading the product description, the client realizes that the product is not a good fit, the client has only read one or maybe two pages and has not lost a sufficient amount of time in a busy day.

The secondary audience is internal. As previously mentioned the sales and marketing departments can learn about the product through the brief product description and utilize the document in meetings with potential clients. In addition, new employees of the software development company use the product description to learn about the product.

## The Scope

Do not focus on details when writing product descriptions. The readers only need to obtain the general overview of the product and to learn how the product fits within the toolset.

Product descriptions can be divided into the following sections:

- Overview
- Features
- Diagrams

### Overview

Describe the product in less than five sentences. If you are unable to do this, then chances are you do not know enough about the product to accurately write a product description. As mentioned before, the reader wants a fast read so a decision can be made quickly about whether the product is useful to the company. In addition, include how the product is used with other products maintained by the company and/or with third-party software programs. Convey if the product is dependent on other products or if the product can function alone.

### Features

When writing the Feature section of a product description, use action verbs and state the primary features or functions of the product. This information is most readable if conveyed in a list format. This format enables users to skim the list in an expedited fashion. Each entry in the Features section is one sentence, two at the most. Details about the features are included in another manual, most likely, the user manual.

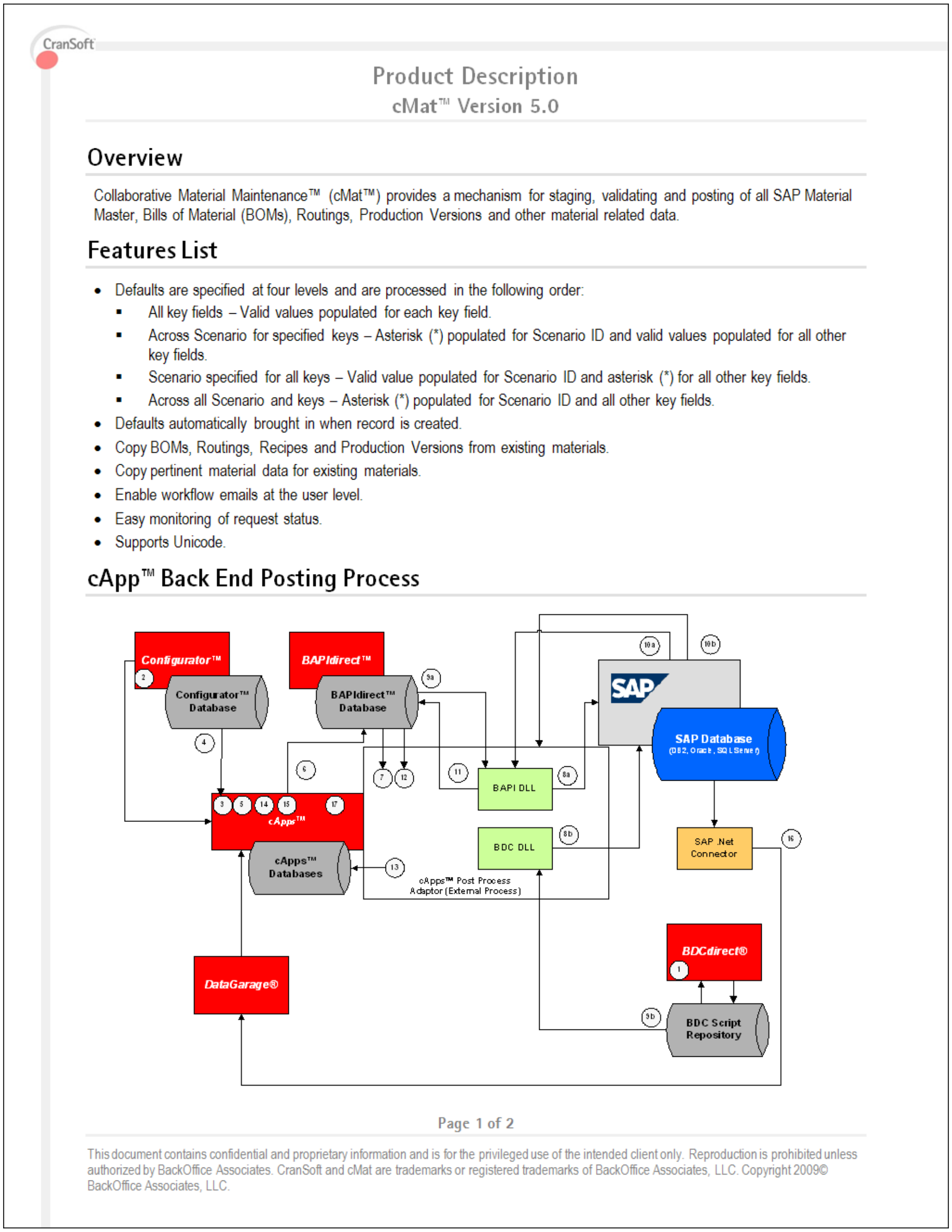
With many companies using their products to compete for the same market, such as Apple and Dell in computer manufacturing, ensure that all features that make the product superior to a competitor's are listed.

### Diagrams

Diagrams are always useful to visually represent the product. Screen shots of the actual product are not included in a product description; they are too specific and are beyond the narrow scope of this document type. A flow chart can be included to illustrate how the product fits within a toolset. Or, if the primary feature of the product includes many steps, a flow diagram can be included in the product description to represent the path.

A workflow diagram of the "big picture" of a product is invaluable for usability.  The diagram can include what the product accomplishes or the void it fills, how the target audience uses it, and how data are pushed through.

Figure 9 is an example of the product overview for the 5.0 release of cMat™. From release to release, the only changes made to a product overview are likely to be the addition of a few bullet points to the Features section.



CranSoft

# Product Description

cMat™ Version 5.0

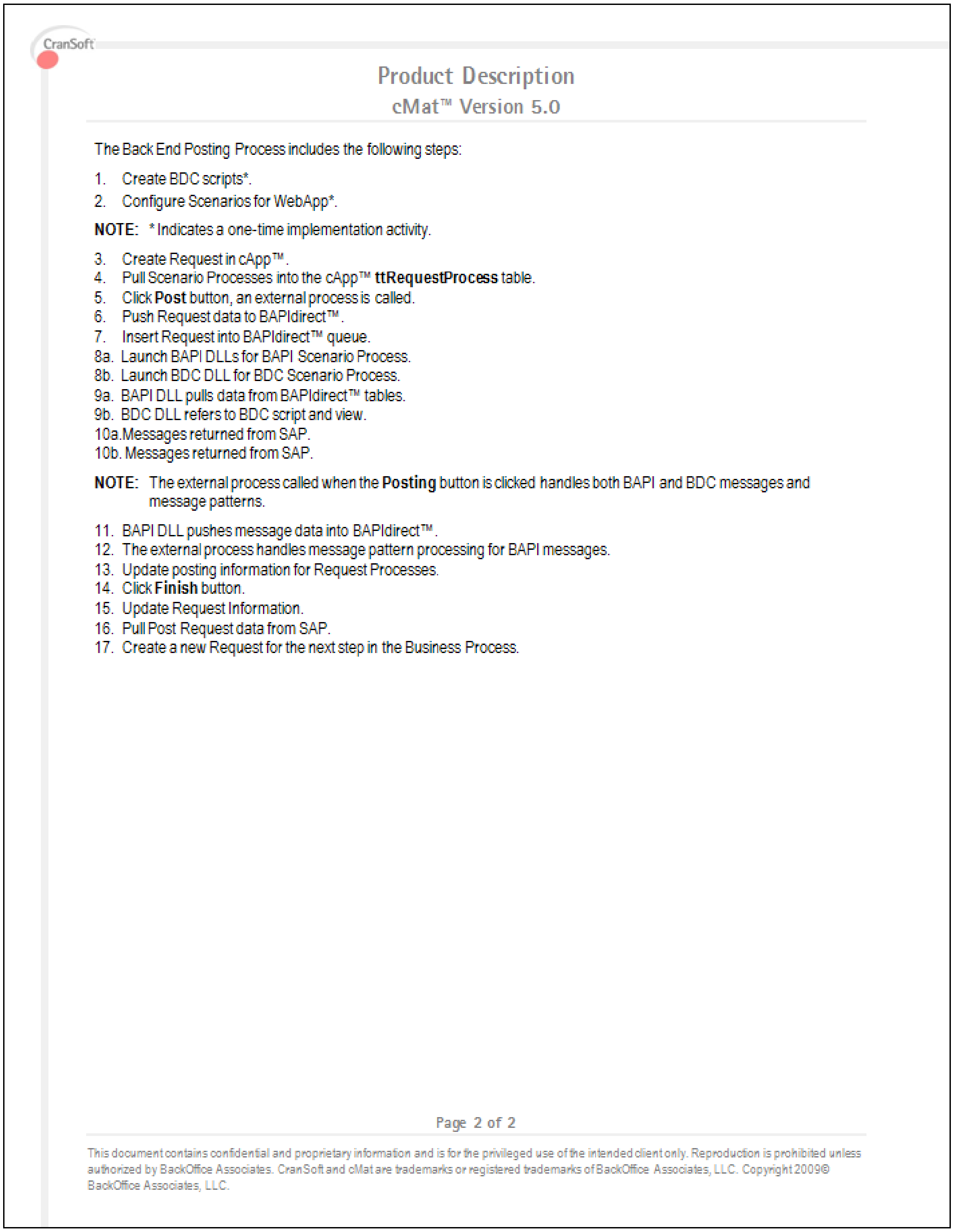
## Overview

Collaborative Material Maintenance™ (cMat™) provides a mechanism for staging, validating and posting of all SAP Material Master, Bills of Material (BOMs), Routings, Production Versions and other material related data.

## Features List

- Defaults are specified at four levels and are processed in the following order:
  - All key fields – Valid values populated for each key field.
  - Across Scenario for specified keys – Asterisk (*) populated for Scenario ID and valid values populated for all other key fields.
  - Scenario specified for all keys – Valid value populated for Scenario ID and asterisk (*) for all other key fields.
  - Across all Scenario and keys – Asterisk (*) populated for Scenario ID and all other key fields.
- Defaults automatically brought in when record is created.
- Copy BOMs, Routings, Recipes and Production Versions from existing materials.
- Copy pertinent material data for existing materials.
- Enable workflow emails at the user level.
- Easy monitoring of request status.
- Supports Unicode.

## cApp™ Back End Posting Process



CranSoft

Product Description
cMat™ Version 5.0

The Back End Posting Process includes the following steps:

1. Create BDC scripts*.
2. Configure Scenarios for WebApp*.

**NOTE:** * Indicates a one-time implementation activity.

3. Create Request in cApp™.
4. Pull Scenario Processes into the cApp™ **ttRequestProcess** table.
5. Click **Post** button, an external process is called.
6. Push Request data to BAPIdirect™.
7. Insert Request into BAPIdirect™ queue.

8a. Launch BAPI DLLs for BAPI Scenario Process.
8b. Launch BDC DLL for BDC Scenario Process.
9a. BAPI DLL pulls data from BAPIdirect™ tables.
9b. BDC DLL refers to BDC script and view.
10a.Messages returned from SAP.
10b. Messages returned from SAP.

**NOTE:** The external process called when the **Posting** button is clicked handles both BAPI and BDC messages and message patterns.

11. BAPI DLL pushes message data into BAPIdirect™.
12. The external process handles message pattern processing for BAPI messages.
13. Update posting information for Request Processes.
14. Click **Finish** button.
15. Update Request Information.
16. Pull Post Request data from SAP.
17. Create a new Request for the next step in the Business Process.

Page 2 of 2

This document contains confidential and proprietary information and is for the privileged use of the intended client only. Reproduction is prohibited unless authorized by BackOffice Associates. CranSoft and cMat are trademarks or registered trademarks of BackOffice Associates, LLC. Copyright 2009© BackOffice Associates, LLC.

**Figure 9 – Product Description Example**

### Helpful Hints When Writing Product Descriptions

- Be brief. Users read the product description to gain a quick overview of the product. If they want to learn more, additional documentation is readily available.
- Use bulleted lists when outlining features.
- Try to keep the document to one page; additional pages are okay if diagrams are included.
- Remember – product descriptions are for semi-technical and non-technical audiences, so do not include too much technical information.

## Training Manuals

Training manuals are designed to teach a reader how to use the software by completing real-world, hands-on examples. Training manuals evaluate the reader's knowledge of a particular subject area.

The following are common uses of training manuals:

- Recruitment training
- New hire training
- Teaching the product

Training manuals can also be used for recruitment training. In this situation, the manual tests the user's knowledge of a particular subject or skill. This process can then be used as an evaluation to determine if the user is suitable for the job.

New hire training manuals are very common and useful for large companies. This type of training manual not only teaches a new hire about his or her position within the company, but it can include details on basic office and company information, such as using the fax machine or where the gym is located.

Training manuals can also be used to document and to teach the audience about a product. In this scenario, training materials are created for every applicable software product. Not every product is complex enough to yield a training manual because the concept and primary features can be fully explained and explored within a user manual.

In such cases, a user manual perfectly suffices. Training manuals are exceptionally beneficial because they can be executed during an instructor-led class or, for those who learn best on their own, a self-guided reference tool. The latter is the most common use of a training manual within a software development company and is therefore within the scope of this textbook.

Training manuals are a very important deliverable in software development companies. They provide the following benefits to the audience:

- Review subject matter
- Allow students to participate in the training class versus focusing on taking notes
- Serve as a reference document when access to a subject matter expert is unattainable

## The Audience

The audience of training manuals can vary, which is why determining the audience before outlining the content of a training manual is imperative. The following is a list of common audiences of training manuals:

- Users of the software
- Employees of the software company

For the purpose of this textbook, the goal of training manuals is to educate end users on how to use a piece of software.

Once the purpose of the training manual is established, the audience is determined. The audience greatly controls the level of information that is included in the training manual. With software training manuals there is an expected level of knowledge the users must have. There is always a broad range of skill levels within a single training class, but the manual is written to cater to multiple audiences. For example, it is common to assume that the students are familiar with basic navigation and terminology of the framework, such as menu names, how to edit a record, and items on the toolbar. Training manuals are written with the assumption that the user does not know information specific to a product. For example, a Microsoft Word training class may assume the user knows the function of the “Save” icon or how to rename a file, which are basic Microsoft Office skills. This class may, however, not assume the user knows how to create an index or generate a table in Microsoft Word.

## Training Adults

A concept of training that many technical writers and trainers fail to take into consideration is that teaching adults is different than teaching children. Training manuals for software products are most likely targeted towards adults. Even if documenting children's computer software, the program is unlikely complex enough to yield a training manual, never mind the reality that the short attention span of children is not suited for a software training class.

Adults are more independent-thinkers than children and key in on information relevant to themselves. Malcolm Knowles, known as the "apostle of andragogy" (the philosophy of adult learning processes) conducted extensive research on the educational theory of teaching adults and determined the following five characteristics:

1. Adults have developed a self-concept that allows for the advancement of independence.

2. Adults have personal experiences that influence how they interpret new experiences and information and become learning resources.

3. Adults' readiness to learn is connected to the task or information being acquired and the need for such information.

4. Adults are more likely to learn the information they can put to immediate use and are more problem-centered than subject-centered.

5. Adults have internal motivation to learn that can be based on a number of wants and priorities including an interest in or a need to know the information. (Kaplan 318).

Considering these points enables the technical writer to create a more concise training manual. Do not include "busy work" or historical information into a training class because adults are only interested in information that is pertinent to them and their job. Students are not interested in how a feature used to work or why it was changed, only how it currently works.

# The Scope

The scope of a training manual is determined by the wants and needs of the targeted audience. Analyze the expected and assumed knowledge level of the audience.

Training manuals are typically organized by exercises (smaller tasks) that build upon each other to complete larger task.  Since training manuals walk the reader through a series of steps and functions, most of the information presented is procedural, which enables the audience to learn more difficult and complex subjects or tasks progressively.

Whatever the format is for each exercise, be consistent throughout the manual. The following are common sections within a software training manual:

- Overview
- Intended Audience
- Prerequisites
- Objectives
- Exercises

### Overview

The Overview section welcomes the student to the training class. It provides a high-level description of the product, toolset, and/or the software development company. This section also instructs the reader on how to use the training manual. For example, a training manual could be designed for individual, self-guided training or it could be developed for instructor-led training.

State early in the training manual what is covered during training, i.e. the scope. What will the students learn? What topics will be introduced? And equally as important, what will not be covered during training? The Overview enables the audience to quickly decide if this training manual will be beneficial and be helpful in order to accomplish the targeted goal.

### Intended Audience

The Intended Audience section briefly indicates who this training manual is designed for. It includes the assumed skill level and the type of user (for example, technical or business) who benefits the most from the training manual.

### Prerequisites

The Prerequisites section outlines elements that are required or recommended to complete before taking the training class or reading the training manual. Indicate what is a required prerequisite and what is only a recommended prerequisite. If the training class is designed around certain required prerequisites that students have not fulfilled, the pace of the class will be much slower and the other students (who did fulfill the requirements) will not benefit as greatly as they would otherwise. Students, just like instructors, must be well prepared and qualified to participate in a training class.

Prerequisites may include the following:

- Database and operating systems knowledge, e.g. SQL Server, Oracle, and Microsoft Vista
- Third-party applications knowledge or skill level, e.g. MS Excel or SAP
- A prequel training class

## Objectives

Developing training objectives is highly important, not only for the student, but for the instructor as well. Meeting objectives determines the overall success of a training class. Objectives provide the student with goals and a list of accomplishments to complete within the duration of the training class. In addition, objectives provide a sense of certification – a list of topics and tasks that the student is now certified to complete.

Mary Ellen Lepionka, founder of Atlantic Path Publishing and author of numerous higher education writing books and articles, categorizes objectives according to their description and provides sample action verbs that lead objectives. The Table 2 summarizes Lepionka's theory (Lepionka 56):

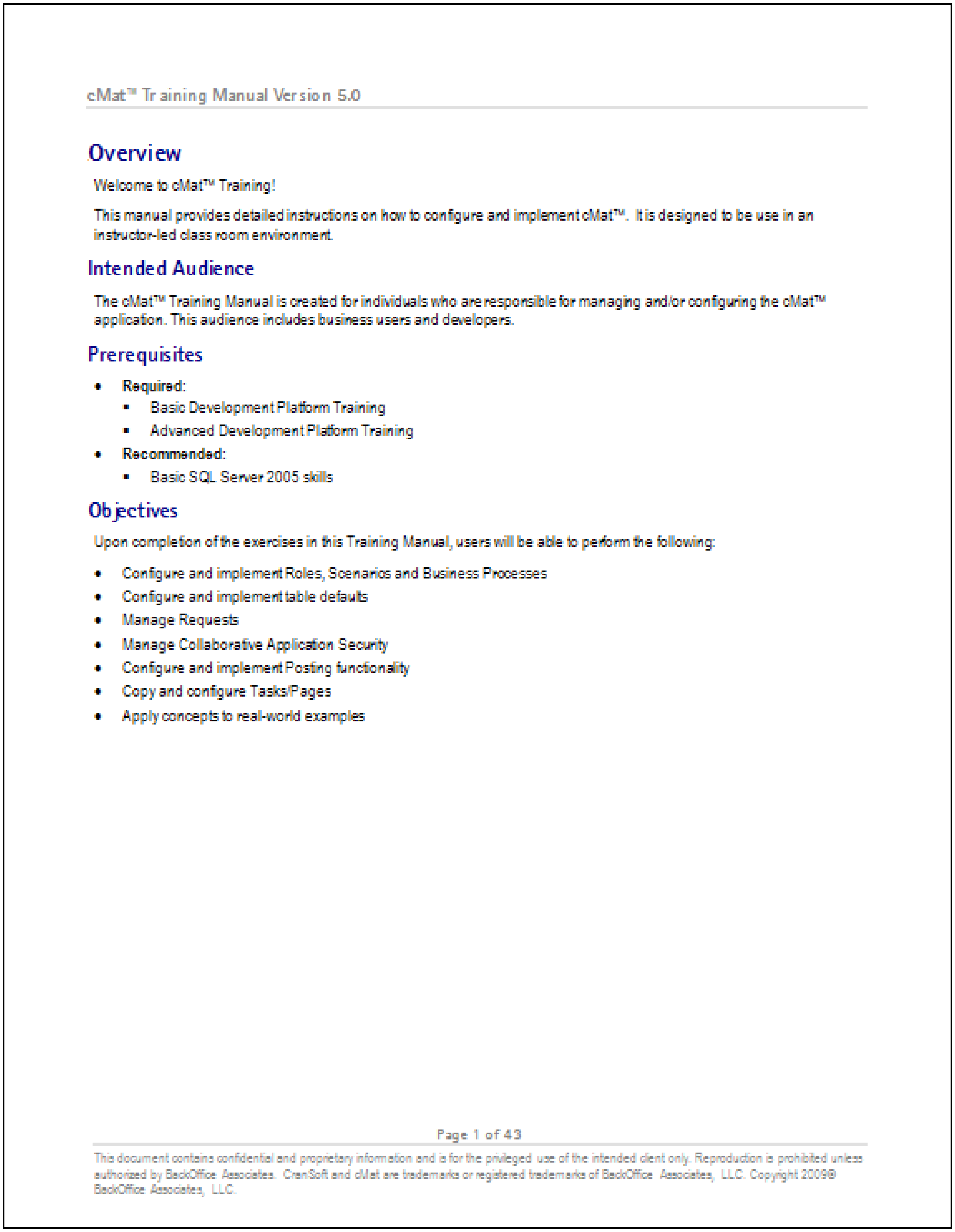
| Objective Category | Description | Example Action Verbs |
|---|---|---|
| **Knowledge** | Remembering or recalling information | • Define<br>• Describe<br>• List<br>• Itemize<br>• Name<br>• State |
| **Comprehension** | Understanding concepts and information. | • Explain<br>• Predict<br>• Translate<br>• Infer<br>• Exemplify<br>• Interpret<br>• Extrapolate<br>• Hypothesize |
| **Application** | Using information and abstractions to solve problems. Applying the concepts to other situations. | • Compute<br>• Apply<br>• Solve<br>• Prove<br>• Calculate<br>• Illustrate<br>• Show<br>• Manipulate<br>• Manage<br>• Decide |
| **Analysis** | Recognizing main points, drawing conclusions, understanding relationships. | • Compare<br>• Contrast<br>• Identify<br>• Analyze<br>• Trace<br>• Relate<br>• Organize<br>• Outline<br>• Distinguish |

| Objective Category | Description | Example Action Verbs |
|---|---|---|
| **Synthesize** | Being innovative and spontaneously creating something new. | • Design<br>• Draw<br>• Create<br>• Express<br>• Build |
| **Evaluate** | Understanding values and judging against standards | • Assess<br>• Evaluate<br>• Rate<br>• Judge<br>• Determine<br>• Accept<br>• Reject<br>• Recommend |

**Table 2 – Summary of Lepionka's Theory on Writing Objectives**

When writing objectives lead with a present tense action. Writing in this tense provides a sense of confidence to the audience and is a clear way to set firm goals. If the purpose of the training class is to evaluate the knowledge of students in addition to teaching students new concepts, objectives can be used as grading system.

Figure 10 is an example of the Overview section of the cMat™ version 5.0 Training Manual. Notice how the intended audience is stated and how all the information is presented to the audience in a clear format.



cMat™ Training Manual Version 5.0

## Overview

Welcome to cMat™ Training!

This manual provides detailed instructions on how to configure and implement cMat™. It is designed to be use in an instructor-led class room environment.

## Intended Audience

The cMat™ Training Manual is created for individuals who are responsible for managing and/or configuring the cMat™ application. This audience includes business users and developers.

## Prerequisites

- **Required:**
  - Basic Development Platform Training
  - Advanced Development Platform Training
- **Recommended:**
  - Basic SQL Server 2005 skills

## Objectives

Upon completion of the exercises in this Training Manual, users will be able to perform the following:

- Configure and implement Roles, Scenarios and Business Processes
- Configure and implement table defaults
- Manage Requests
- Manage Collaborative Application Security
- Configure and implement Posting functionality
- Copy and configure Tasks/Pages
- Apply concepts to real-world examples

Page 1 of 43

This document contains confidential and proprietary information and is for the privileged use of the intended client only. Reproduction is prohibited unless authorized by BackOffice Associates. CranSoft and cMat are trademarks or registered trademarks of BackOffice Associates, LLC. Copyright 2009© BackOffice Associates, LLC.

Figure 10 – Overview Section of a Training Manual Example

## Exercises

The exercises are the most important elements to a training manual. They are designed to walk the students through core functions and features of the software program. Instead of having one large, continuous exercise, divide larger functions into smaller exercises. A training class with many exercises or modules provides mini-milestones for the students to achieve. Completing several milestones within a training class instills a sense of completion and boosts the students' confidence about learning new information. While there are many ways to organize the exercises within a training manual, exercises are the most beneficial when they serve as building blocks versus when they are independent from each other because building blocks require the audience to complete one task before beginning another.

Within each individual exercise, there are various sections or types of information to include. The following are recommended sections to include within each exercise of the training manual:

- **Overview** – Provides a brief explanation of why the exercise is important. Does it demonstrate how to implement a specific feature? If so, what does the feature do? Does the exercise outline how to accomplish a certain task? If so, why is the task important? If applicable, discuss how the exercise ties in with the training class on a whole.

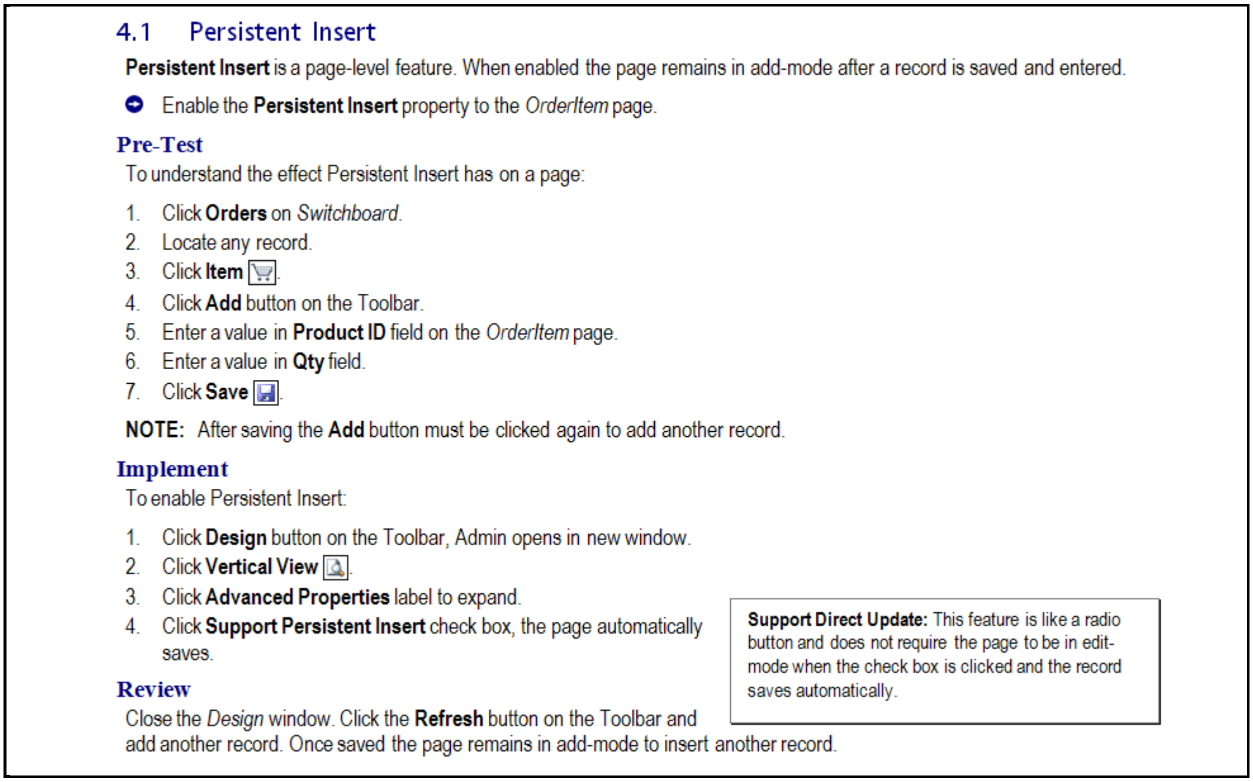
- **Purpose** – Outlines the significance of the exercise. While objectives for the entire training manual are stated, include the purpose of the specific exercise. What is the goal of the exercise? What will the student accomplish?

- **Pre-Test** – Navigates the user through the exercise before the purpose is completed. This section provides the student with the "before" understanding.

- **Implement** – Outlines specific step-by-step instructions on how to execute the exercise.

- **Review** – Navigates the user through the exercise after the purpose has been achieved. This section provides the student with the "after" understanding.

Figure 11 is an example of an Exercise section of a training manual. Notice that while there is not a heading for each section, all relative information is included. The first line of the exercise, "Persistent Insert is a feature … entered and saved" is the exercise overview but is presented in a less-formal format. The steps are clear and direct, which enable the users to follow along as they read the exercise while performing the task in the software.



### 4.1 Persistent Insert

**Persistent Insert** is a page-level feature. When enabled the page remains in add-mode after a record is saved and entered.

➔ Enable the **Persistent Insert** property to the *OrderItem* page.

**Pre-Test**

To understand the effect Persistent Insert has on a page:

1. Click **Orders** on *Switchboard*.
2. Locate any record.
3. Click **Item** .
4. Click **Add** button on the Toolbar.
5. Enter a value in **Product ID** field on the *OrderItem* page.
6. Enter a value in **Qty** field.
7. Click **Save** .

**NOTE:** After saving the **Add** button must be clicked again to add another record.

**Implement**

To enable Persistent Insert:

1. Click **Design** button on the Toolbar, Admin opens in new window.
2. Click **Vertical View** .
3. Click **Advanced Properties** label to expand.
4. Click **Support Persistent Insert** check box, the page automatically saves.

**Support Direct Update:** This feature is like a radio button and does not require the page to be in edit-mode when the check box is clicked and the record saves automatically.

**Review**

Close the *Design* window. Click the **Refresh** button on the Toolbar and add another record. Once saved the page remains in add-mode to insert another record.

Figure 11 – Training Manual Exercise Example

## The Instructor's Manual

Some instructors of a training class prefer to have a version of the training manual that is targeted towards the individual leading the class. Such a manual is referred to as the Instructor's Manual. This manual is identical to the student's version, but includes prompts, talking points, and areas that always cause issues for students.

### Helpful Hints When Writing Training Manuals

- Know the audience. If the exercises are too advanced or too basic for the audience, they will not be beneficial.
- Determine early on in the writing process what the training manual will and will not cover.
- Determine an example or a case study that will be the common theme throughout the manual.

## Quick Reference Guides

Quick reference guides are an abridged version of a manual, often a mere fraction of the length of the original. They are often no more than two pages: one double-sided sheet of paper. The purpose of this document type is to provide users who are knowledgeable enough about the topic with a brief document that spares many details.

The goal of writing a quick reference guide is not to cut words out to make a shorter manual, but instead to compress the manual, removing details that can be spared (Johnson, "Quick Reference Guides: The Poetry of Technical Writing").

While writing a shorter document may seem like a far easier task than writing a complete manual, it is not always the case. Mastering the quick reference guide takes a lot of concise writing practice.

There are many uses of quick guides:

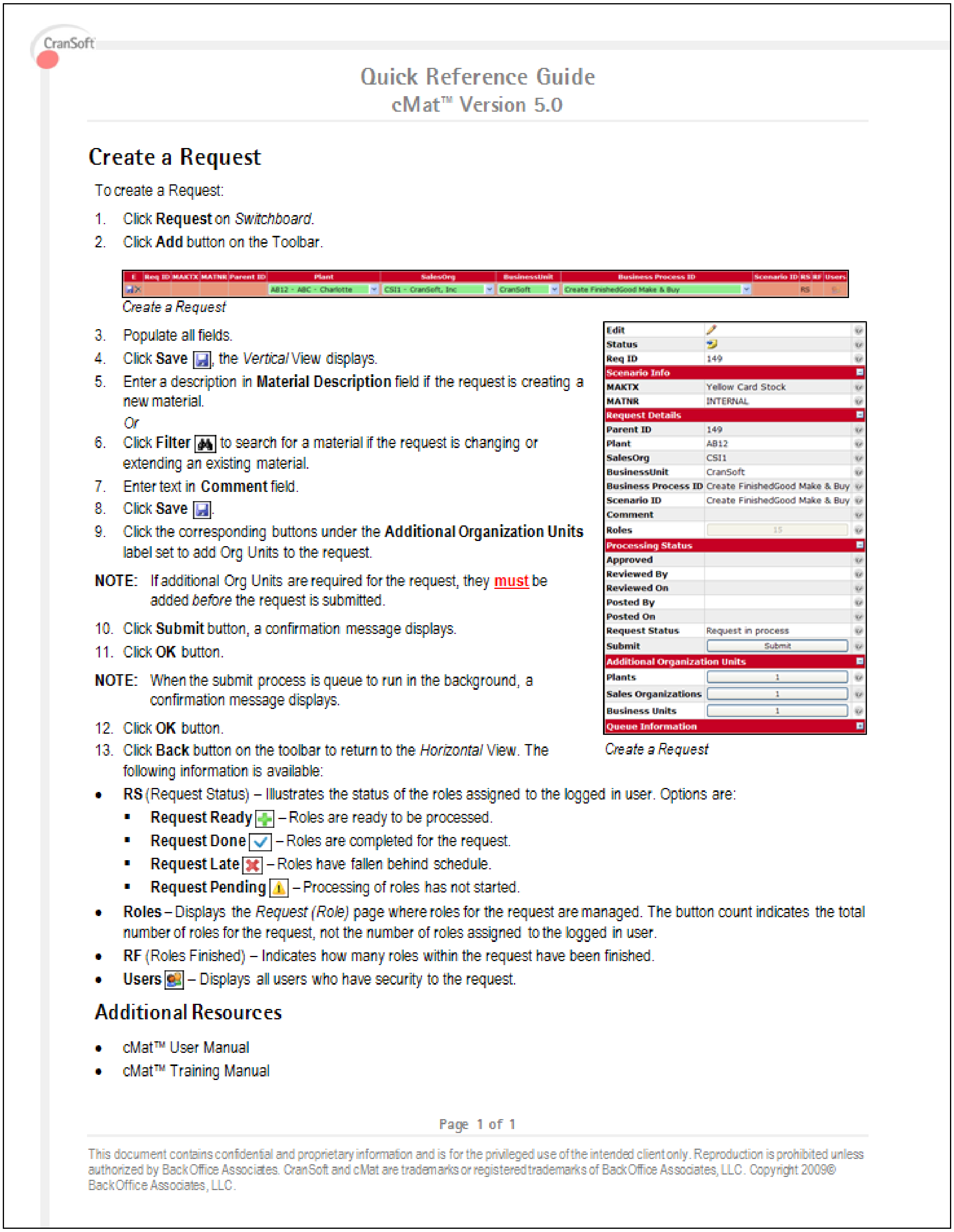
- **Supplement a user manual**. If the software program is easy to use and has only a few features, a quick reference guide can be used as the abridged version of a user manual (sacrificing some details, of course).

- **Itemize core tasks.** If a product caters to many audiences, and therefore includes a lot of features that may not be relevant to all audiences, a quick reference guide could include all the core tasks (i.e. tasks performed by all or most of the audiences). Or, several core tasks can be created for a single quick reference guide but cater to specific audiences.

- **Replace large documents.** Readers are often overwhelmed by large documents. Consider making several quick reference guides to replace a single large document.

The purpose of this document type is to satisfy advanced users who do not need many details in order to perform the function.

Figure 12 demonstrates a practical use of a quick reference guide. In this example, one core task, creating a request, is documented on a high level in a one-page document. If the user requires additional details or information, the Additional Resources section suggests full-length documents.



CranSoft

Quick Reference Guide
cMat™ Version 5.0

## Create a Request

To create a Request:

1. Click **Request** on *Switchboard*.
2. Click **Add** button on the Toolbar.

| E | Req ID | MAKTX | MATNR | Parent ID | Plant | SalesOrg | BusinessUnit | Business Process ID | Scenario ID | RS | RF | Users |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  | AB12 - ABC - Charlotte | CSI1 - CranSoft, Inc | CranSoft | Create FinishedGood Make & Buy |  | RS |  |  |

*Create a Request*

3. Populate all fields.
4. Click **Save** , the *Vertical* View displays.
5. Enter a description in **Material Description** field if the request is creating a new material.

   *Or*
6. Click **Filter** to search for a material if the request is changing or extending an existing material.
7. Enter text in **Comment** field.
8. Click **Save** .
9. Click the corresponding buttons under the **Additional Organization Units** label set to add Org Units to the request.

**NOTE:** If additional Org Units are required for the request, they **must** be added *before* the request is submitted.

10. Click **Submit** button, a confirmation message displays.
11. Click **OK** button.

**NOTE:** When the submit process is queue to run in the background, a confirmation message displays.

12. Click **OK** button.
13. Click **Back** button on the toolbar to return to the *Horizontal* View. The following information is available:

|  |  |
|---|---|
| Edit |  |
| Status |  |
| Req ID | 149 |
| Scenario Info |  |
| MAKTX | Yellow Card Stock |
| MATNR | INTERNAL |
| Request Details |  |
| Parent ID | 149 |
| Plant | AB12 |
| SalesOrg | CSI1 |
| BusinessUnit | CranSoft |
| Business Process ID | Create FinishedGood Make & Buy |
| Scenario ID | Create FinishedGood Make & Buy |
| Comment |  |
| Roles | 15 |
| Processing Status |  |
| Approved |  |
| Reviewed By |  |
| Reviewed On |  |
| Posted By |  |
| Posted On |  |
| Request Status | Request in process |
| Submit | Submit |
| Additional Organization Units |  |
| Plants | 1 |
| Sales Organizations | 1 |
| Business Units | 1 |
| Queue Information |  |

*Create a Request*

- **RS** (Request Status) – Illustrates the status of the roles assigned to the logged in user. Options are:
  - **Request Ready** – Roles are ready to be processed.
  - **Request Done** ✓ – Roles are completed for the request.
  - **Request Late** ✗ – Roles have fallen behind schedule.
  - **Request Pending** ⚠ – Processing of roles has not started.
- **Roles** – Displays the *Request (Role)* page where roles for the request are managed. The button count indicates the total number of roles for the request, not the number of roles assigned to the logged in user.
- **RF** (Roles Finished) – Indicates how many roles within the request have been finished.
- **Users** – Displays all users who have security to the request.

## Additional Resources

- cMat™ User Manual
- cMat™ Training Manual

Page 1 of 1

This document contains confidential and proprietary information and is for the privileged use of the intended client only. Reproduction is prohibited unless authorized by Back Office Associates. CranSoft and cMat are trademarks or registered trademarks of Back Office Associates, LLC. Copyright 2009© Back Office Associates, LLC.

Figure 12 – Quick Reference Guide Example

## The Audience

The audience of quick reference guides is narrow – these documents are not for every user. The primary audience is the advanced user of the software. If this user is knowledgeable and skilled with the software program, the standard user manual may contain too much information and details for an advanced user. Too much time is spent thumbing through the 50-page document when all the advanced user really wants to know is a basic function that involves five or six steps (Johnson, "Quick Reference Guides: The Poetry of Technical Writing").

The secondary audience is users who do not care to read or know all the details of the software. Many software end users learn best when they can click through the program and learn the functionality themselves. But even the best clickers get stuck from time to time. The quick reference guide may burn their pride, but will help them in the long run.

## The Scope

Determining the scope of a quick reference guide can easily become one of the most difficult tasks in the writing process. Because the document length is two pages or less, focusing on what not to include may be an easier task than determining what to include. If starting from a blank document, the best plan of attack is to write more than you anticipate including in the final product. If starting from an existing manual, begin by removing all irrelevant and detailed information. Then, think about what information is essential and what can be included in either another quick reference guide or perhaps a full-length manual.

The scope for a single quick reference guide is core tasks. If the quick reference guide is for an entire software program, write about the core features and functions of the product at a high level. If the quick reference guide is for a specific feature or function, limit the scope to only that core task.

If space allows, include a list of resources where more information can be obtained, such as the full-length user manual or another quick reference guide.

## Helpful Hints When Writing Quick Reference Guides

- Be aware of space and layout. Since the goal is to fit information into less than two pages, space is precious. Do not be wasteful.
- Distinguish between essential tasks and information, and non-essential tasks and information.
- Understand that a quick reference guide is not the solution for all documentation projects.
- Teach in five minutes with a quick refernce guide what would take fifty minutes to teach in a full-length user manual.

## Chapter Summary

The following key points have been covered in this chapter:

- The basic deliverables of technical writers working for a computer software company are user manuals, product descriptions, release notes, training manuals, and quick reference guides.
- The primary goal of any technical writing deliverable is to communicate to the intended audience.
- Each deliverable has a distinct scope and intended audience that must be analyzed by the writer before drafting.

# Chapter 5: Technical Writing Process

## Chapter Contents

## Chapter Objectives

The following are objectives of this chapter:

- To evaluate the similarities and differences between technical and creative/academic writing.
- To understand the primary phases of the writing process for technical documents.
- To comprehend the basic elements of planning a document.
- To evaluate the essential elements of achieving a clear focus.
- To direct a document towards multiple audiences.
- To know when to apply the different levels of editing as well as the importance of each one.
- To understand the invaluable importance of document testing.
- To explain how to increase the usability of a document.
- To recognize the various ways to produce a document.

## Overview of the Writing Process

Let us start by covering the basics: What is a "writing process"? A writing process is the stages a writer progresses through to generate a document (in its most simple definition). Since many technical writers stem from technical backgrounds versus writing backgrounds, the writing process (which is taught within writing programs)  is often over-looked or even unknown to the writers. The writing process varies from writer to writer and even from project to project. The writing process for technical writing is similar to – but at the same time greatly differs from – the academic or creative writing processes.

## Technical versus Academic or Creating Writing Process

The main similarity between the two classifications of processes is that both are (arguably) circular processes rather than linear. Both processes emcompass prewriting, drafting, revising, and editing, and each phase is revisited throughout the cycle, thus making the writing process circular.

While the basic steps of the academic writing process are embedded within the tehcnical writing process, there are vast differences the writing processes for these generes of writing that are shaped by the writing environment, writing mechanics, and writing style.

### *Time Constraints*

While technical, academic, and creative writing often have deadlines, technical writing must frequently be written quickly and efficiently with little notice. Deadlines impact how to plan a document and how much time is allotted for each phase of the writing process. During a tight deadline, stages must be combined and performed in tandem, specifically editing and revising. With a time constraint, there is often very little room for editing, so this phase must be performed while drafting.

### *Planning*

Other key differences between the various disciplines of writing are planning and brainstorming. Many creative and academic writers compose at their leisure. Research is often conducted, but the most time-saturated phases of the academic writing process are composing and editing. While composing a technical piece, research is conducted throughout the entire writing process.

### *Collaboration*

A primary difference between the technical writing and the creative or the academic writing processes is often the need for collaboration. Creative writers often view collaboration as stealing ownership of the written piece. If the opinions and thoughts of outside writers are used or taken into consideration while composing, creative writers are reluctant to see the piece as solely theirs. Some writers in an academic setting, such as scholars, often collaborate. However, depending on the nature of the piece, many choose to work alone. Technical writers usually do not enter a writing process with this frame of mind. Collaboration, not only among other writers but with developers and with the product manager as well, is absolutely essential for the accuracy, clarity, and credibility of the final product.

Typically, technical writers are not the subject matter experts on the topic on which they are writing, but they have to sound like they have a wealth of knowledge. Collaboration may often take effect during the planning, revising, or editing stages of the writing process. Subject matter experts (often the developer, product manager, or quality assurance tester) are called upon to verify the content is accurate.

## The Technical Writing Process

Like the creative or academic writing processes, the process of technical writing is circular, rather than linear (arguably speaking). However, unlike the academic writing process, the technical writing process usually includes many people and consists of more collaborative phases. As previously mentioned, technical writers more often than not lack expert knowledge about the subject about which they are writing, nor are technical writers necessarily editors. Therefore, the process for technical writing can be far more convoluted, complex, and involve more individuals than that of creative or academic writing.

This writing process can be broken down into four main phases:

1. **Plan** – Involves great research and formulation of an outline.
2. **Draft** – Creates the rough draft, which is often the most collaborative phase.
3. **Revise and Edit** – Pays attention to the style, organization, and simplicity of the writing.
4. **Production** – Finalizes the “look and feel” of a document and packages it for delivery.

These steps often overlap and vary in time consumption and intricacy, depending on the writing assignment. For example, due to the nature of a training manual, a great deal of emphasis is placed on the Revise and Edit phase because there is so much detail that has to be perfect. Each step of every exercise has to precisely coincide with the software in order for students to grasp the concept and not to be confused while learning new material. However, while some phases are more elaborate and occur more frequently throughout the composing process, all phases are essential to produce a successful and complete product.

## A Model for the Writing Process

The writing process is highly theorized in various ways, and, therefore, there is no one model that represents the writing process. Figure 13 provides a diagram of a model for the writing process.

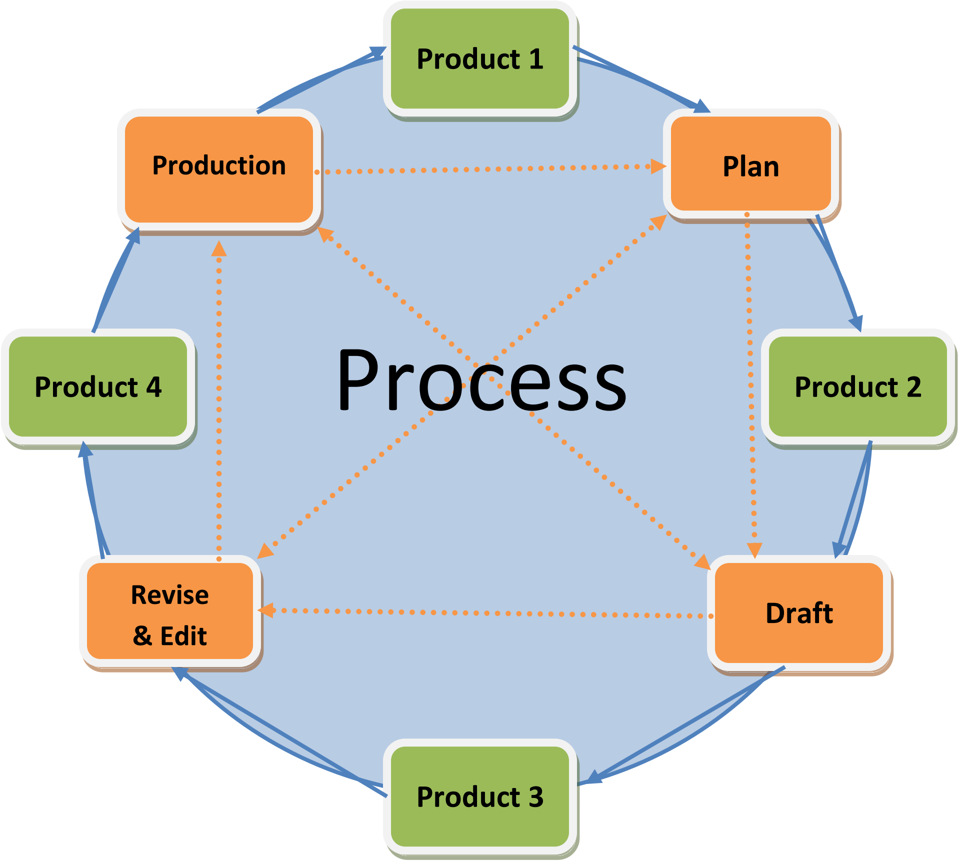


Figure 13 – A Model for the Writing Process

As previously discussed, the writing process is non-linear. The process is a continuous cycle that does not have an ending point, just as there is not an ending point to a circle. A document can be finished and published; however, it will be updated for the next software release to contain the current functionality. The product is not one ultimate end result, but instead can be viewed as multiple milestones within the process. The overall progression of the writing process is circular and flows through each phase. In this writing model, the straight lines represent the overall progression of the writing process that moves between each milestone (also known as the "product") and each phase. Notice how the Production phase links back to Plan. The Production phase links to the Plan phase because a piece of technical writing is never finished. Documentation continuously changes as new versions of software are released. The dotted lines signify the cross reference and collaboration within the writing process.

## Plan

The best piece of advice for any technical writer is to think before writing. As trivial as it sounds, take a moment, before applying pen to paper, to think about what the piece will entail. This simple task is often hard with deadlines quickly approaching, but as in any process, haste makes waste.

Planning a document includes defining the purpose, analyzing the audience, and defining the scope. The goal of a technical document is to communicate information to the reader. Therefore, the success of a document is measured by how well the audience is able to extract information from the document. The document must be easy to read, easy to understand, and easy to follow correctly. The intended audience is reading a document to quickly learn the information in order to perform a function, such as how to use the software or to decide whether or not to purchase the software.

When creating a deliverable, the most important element is determining the focus. The focus is achived by defining the purpose, the audience, and the scope. If these elements are not determined during the planning phase of the writing process the deliverable will not have a definitive direction. With a deadline quickly approaching, writers often run before they walk and, consequently, hit a wall part way though the writing process. Having a clear focus before composing makes the writing process streamlined.

### Define Purpose

The purpose of a document must be one of the first elements to consider when planning a deliverable – what do you want the audience to do with the document? Is the document to persuade the audience, or perhaps to instruct? If a clear purpose is not defined, there is no direction or goal of the document.

For example, the purpose of a product description is to persuade a potential client to buy the software program, while release notes is intended to persuade an existing client to upgrade to the latest version of the program. User manuals and training manuals are designed to instruct the audience by walking the users, step-by-step, through functions and tasks. The ultimate purpose of a document is to communicate information to the targeted audience.

The purpose of a document also determines how the end product is formatted. Since release notes persuade the intended audience, bullet points and short descriptions are often used to enable the audience to read the document quickly in order to make an informed decision. Training manuals, however, are designed to instruct and be used as reference materials. Therefore, they include a great amount of detail with enumerated steps to navigate the user through a function or a task.

## Audience Analysis

If you do not know for whom you are writing, how and where do you begin? A deliverable is written to cater to a specific audience. Defining the audience is especially important when writing for a software company due to the wide range of audiences and skill levels. The usability and organization of a document is determined by how well it reaches the needs of the target audience.

The typical audiences of documents created for a software development company include the following:

- Clients
- Potential clients
- Developers
- Sales and marketing departments

Do not start planning a document without knowing the audience. If the audience is unknown or unclear during the planning phase of the writing process, assume the audience has no prior knowledge of the topic. If many details are included in the document and then it is discovered that the audience is highly technical, removing content is not a huge effort. If the opposite is true, where not enough detail is included and the audience turns out to be non-technical, adding additional content is a much harder task. Always underestimate how much the audience knows.

### Determine Technical Level

With a wide range of audiences comes an equally wide range of experience, knowledge, and technical backgrounds. Most of the deliverables outlined in this textbook are developed for semi-technical to non-technical audiences. Assume the audience has no or very little experience with the topic. Therefore, a lot of details, definitions, and explanations are included. Software documentation geared towards a technical audience tends to use more jargon and fewer descriptions, and it assumes the audience has a high level of knowledge of the product. Very little documentation produced by technical writers in the IT industry is targeted towards technical audiences. Technical Writers are usually hired to produce documentation for clients because unlike documentation for technical audiences (i.e. developers), end user documentation is a profitable deliverable.

The Table 3 provides a high-level overview of the technical knowledge levels of the audience types. Details are provided in subsequent sections.

| Technical Level of Audience | Description | Users | Writing Style |
|---|---|---|---|
| **Technical** | Very experienced with the product and technical concepts. An advanced user. | • Developers<br>• System Administrators<br>• Quality Assurance Testers | • Lacks descriptions and definitions.<br>• Omits step-by-step instructions |
| **Semi-technical** | Some experience with the product. An intermediate user. | • Existing clients<br>• Sales Department<br>• Developers | • Contains facts and figures.<br>• Explanations and definitions are included, but brief. |
| **Non-technical** | No experience with the product and may not have previous knowledge of the concepts. A novice user. | • Potential clients<br>• Sales Department<br>• New employees | • Contains lots of definitions, step-by-step instructions, and details.<br>• Assumes the audience has no prior knowledge on the topic. |

Table 3 – Technical Levels of Audience

### *Technical*

Audiences with great knowledge of the software are considered *technical*. In the computer software industry, technical audiences can be developers (not just of the product being documented, but any developer), individuals who install, implement, or support the software (often consultants), or clients who are experienced with the product. For this audience, use technical jargon, acronyms, and abbreviations.

Documents written for a technical audience contain facts and figures. Since this audience is very familiar with the product, there is no need for a lot of details and definitions.

The ultimate goal of any document is to convey information to the audience so the desired knowledge can be quickly obtained in order to complete the task at hand. If a technical audience is overwhelmed with definitions and thorough explanations, the reader becomes annoyed and frustrated with the document, which lowers the document's usability.

### *Semi-technical*

A *semi-technical* audience has some experience with the product, but not as much as a technical audience. Documents written for semi-technical audiences contain facts and figures that are explained with little detail.

Sales and marketing departments and existing clients are semi-technical audiences, depending on an individual's experience with the company and with the technical concepts (since not all salesmen of computer software actually have a technical background).

When writing for a semi-technical audience, minimize the use of corporate or software jargon and spell out acronyms and abbreviations.

### *Non-technical*

*Non-technical* audiences, such as new or potential clients, are not familiar with the product. Therefore, documents written for this class of audiences require facts and figures to contain many details and explanations. Details are outlined in the simplest terms without using corporate or software development jargon. The layperson is dependent on the technical writer to "translate" information provided by the developer (a technical audience) so content is comprehendible. Information must be presented in simple and digestible terms. A non-technical audience easily becomes frustrated if the information is only understood and digested after reading the document several times. There is no such thing as too much information or too many details for non-technical audiences.

When writing for a non-technical audience, do not use jargon and spell out all acronyms and abbreviations. Always remember that non-technical audiences are not familiar with the program or possibly the company, so write simply, yet be descriptive and clear. If highly technical terms are unavoidable, provide the non-technical audience with a clear definition. This practice prevents the terminology from delaying the audience in reading the document.

A non-technical audience is often the most difficult to write for. The more you know about a product, the more you assume the audience knows, and you forget how much knowledge you have acquired. Take a step back and remind yourself how little this audience knows about the software program.

## Writing for Multiple Audiences

When producing a deliverable, the ideal situation is to have a single audience compiled of individuals with the same technical background and knowledge of the product. Rarely is this the case. Even within an audience class, such as clients, there are various ranges of technical abilities. Be prepared to write individual documents targeted towards multiple audiences as well as different technical levels. To help determine which audience to focus the document towards, decide which is the *primary* audience and which is the *secondary* audience.

- **Primary Audience** – The *primary audience* is the target reader, is often decision maker, and is audience who requested to read the document.

- **Secondary Audience** – The *secondary audience* may come in contact with the document, but is not the intended audience.

For example, the audiences of release notes are generally the sales department of the software company and the clients looking to upgrade to the most recent version of the software. The sales department is often a non-technical audience because members of this department are more familiar with the bigger picture of the software than individual functions. Clients looking to upgrade, however, are often a semi-technical audience because they have been using the software and they know what elements in the release notes help decide if upgrading is beneficial to their business' needs. In this scenario, the client is the *primary audience* and the sales department is the *secondary audience*.

As a rule of thumb, when the document is short (less than two pages), tailor it to both the primary and the secondary audiences. If the document is long (more than two pages), tailor it to the primary audience and include sections for the secondary audience, such as appendices. Writing for multiple audiences is a big enough challenge; narrowing the audience for shorter documents facilitates the process.

## Define Scope

The scope is the amount of detail and information included in a document.  Defining boundaries of the scope enables you to decide what information to include in the document and, equally as important, what information to exclude.

How much information does your audience need to know? The purpose and audience of the document greatly influence the scope. If the purpose of the document is to persuade the audience, then instructional steps are out of scope. Or, if the audience is technical, providing a glossary of terms and definitions is too much detail and unnecessary information for the audience.

## Research

Conduct research to learn the product. The goal of researching is to learn every function applicable to the end user, such as the main features, the purpose of each page, and the use of each field.

Keep in mind that the research process usually continues throughout the writing process. As you become more familiar with the product and you are deep into the drafting phase, additional questions to ask the developer may surface.

## Sources

Recognize available and relevant resources before researching information on the product. Existing documentation, developers, and the product itself are often the most valuable resources when documenting the product (Bremer 111).

Using a variety of sources is important because one source never provides you with all the information to document a product accurately.

### *Existing Documentation*

Existing documentation is a great place to start the research process. Leveraging the work completed by others reduces your time spent on researching and drafting and in turn, expedites the writing process. The first step to researching is to find any existing documentation: technical specifications, marketing material, proposals, or presentations.

A specification document is like the proposal of the product – it justifies the need for the product, indicates how it fits into the toolset (if it is part of a toolset) or why it can stand alone as a separate product, walks through each page (of the prototype), and provides a high-level description of the product’s primary features. Specification documents can be very useful when learning about a new product, but since they are created during the design phase of the product, these documents do not always contain much detail. Also, be wary that these documents are subject to change throughout the software development process.

Stay in contact with the marketing department because you both write about the product to the same audience. If the marketing department has produced literature by the time you begin documenting the product, it could be very useful, especially when writing a product description.

Marketing material is written very differently than product documentation. Be aware that you may not be able to use verbatim what the marketing department provides, since these deliverables often have a marketing spin to the prose.

### *Developers*

Interview developers to learn about a product. If a specification document (or any other documentation) is not available, conduct a preliminary interview to learn the basic information about the product. Interviews can either be conducted in person or via email – whichever works best for the writer and for the developer.

Working with developers can be a challenge. It is not an uncommon mentality of a developer to assign a low priority to documentation. Some developers provide technical writers with minimal information on the product because they do not have the time or the patience to document the product while developing. Other developers create specification documents for the product that contain highly detailed information. Susan

Wu, a working technical writer in Shanghai, comments on her experience as a technical writer while working with various developers: “we just have to dig through all the jargon, look for and add missing content, and stitch the pieces together. It’s like working on a 10,000 piece jigsaw puzzle without a clue what the final picture looks like” (Wu). Therefore, it is almost necessary to utilize other researching methods in addition to developers.

### *Program*

The software program itself is a very beneficial source of information that technical writers do not always think to utilize. As a technical writer, you can act as the end user while you teach yourself the product. Learn each feature and take note of what pages you naturally go to first. You are able to view the product with a fresh set of eyes, just like the users for whom you are writing.

While you navigate through the product, take notes. Observe where you go first and try to figure out the overall picture and the purpose of the product. Write a list of questions you have for the developer, such as field name definitions and any part of the product that you do not understand. Chances are if you have questions, the end user will have the same questions. Remember to incorporate your findings in the documentation.

As a fresh set of eyes, you may also be able to identify errors in the program. Look out for functionality that does not work as you imagined it would. The errors may either be a mistake on the developer’s behalf, or they may mean that the program is not as intuitive as the developer (or you) imagined (Bremer 114).

Technical writers are detail oriented by nature. Be on the lookout for inconsistencies in field names, page names, buttons, etc. As in documentation, if the same field is referenced by two different names, the end user assumes the field has different functions.

## Draft

Write an outline of the document before beginning the drafting phase. The more detailed an outline is, the easier the draft is to write. Bear in mind that the draft is a rough draft, where revisions and edits are expected and required. When writing the rough draft, do not stop to edit. The goal of producing the first draft is to get all thoughts on paper.  Having spelling mistakes and writing in incomplete sentences is okay. Stopping to edit is not okay if doing so impedes the drafting process.

### Writing the First Draft

When composing the first draft, writers tend to start with the introduction and draft in a methodical and progressive fashion that ends with the conclusion. When creating a technical document, this plan may not be the most useful. Instead, try to write the easiest section first, i.e. what you have the most knowledge on and what words flow naturally from mind to paper (or mind to computer). There is no rule that states sections must be composed in the order they appear in the final product.

The theme of collaboration carries over from brainstorming, through outlining and into drafting. Depending on the scope of the project, a "divide and conquer" mentality surfaces where sections of the project are assigned to various project team members and then reviewed by remaining team members. The benefit of working with a group of writers is that each writer may have a different technical skill level or understanding of the product. Therefore, writers often have different sections of the manual that they prefer to write.

### Working with Incomplete Sections

During the composition of the first draft, be aware that you may be dependent on other team members to finish their work before you can complete yours. For example, you may have to draft the introduction later in the writing process because the marketing material you want to leverage is not finished. Alternatively, if development is not complete, there may be additional pieces of functionality to include in the documentation later. A good rule of thumb is to wait until the product is in the Quality Assurance (QA) phase (where the product is tested to verify it functions properly) of development before beginning to document. While there are no guarantees that development ceases when the product is being tested, changes are usually kept to a minimum after this phase.

When drafting, make notes in the document either to yourself or to other members of your team. When working simultaneously on multiple projects, forgetting where you had questions or where you need to develop ideas is an easy task.

## Revise and Edit

Revising and editing can be viewed as two separate phases of the writing process, but they can also be viewed as the same. In this textbook, the two phases are treated as one. In a software development environment, the technical writer is commonly the editor, too. While learning editing techniques is beyond the scope of this document, recognizing the editing role in the writing process is important.

### Revising

Revision is continuously performed throughout the entire technical writing process. If possible, revisit the document several hours (or even days, if time permits) between revisions. The amount of time spent revising a software document can be frustrating and overwhelming. A product changes, even if only slightly, throughout the entire development process. Even the renaming of a page or a field in the software could produce changes in the document.

Being aware of this continuous change can make the revision process easier so you are not confounded when change occurs.

### Document Cycle

Commonly known as a *document cycle*, collaborative revision is a powerful tool to perfect the outcome of a final product. The document can be cycled through other members of the project team or subject matter experts assigned to other projects. The goal of a document cycle is to have numerous sets of eyes read the final product for typing mistakes, clarity, content, accuracy, etc. A technical writer is likely to read the document a dozen times, and each time may miss an obvious error that can be caught by participating members of the document cycle. However, if the technical writer is also the developer of the software, chances are the document is only read a few times. Read thoroughly and carefully, and dedicate as much time as possible to perfect the document.

The creation of release notes is an excellent example of how a document cycle contributes to the success of a document. A developer often provides the content, which is then standardized by the technical writer. The document is sent to the project manager to confirm completeness of information. If elements – new features, enhancements, resolved issues, etc. – are missing, the document is returned to the developer so the elements can be added. Once the content is solidified, the writer edits the document and sends it to the director of application development for approval. Once approved by the director, the release notes is considered finished and can be published.

## Editing

Poor editing can lower the reader's confidence; "editing reflects the quality of the document as well as the quality of the product itself" (Corbin 287). Dedicate as much time as you can to editing. There are two primary levels of editing: comprehensive editing and copyediting.

### Comprehensive Editing

Comprehensive editing is the first level of editing that verifies the document is usable, is organized well, and is complete. The goal of comprehensive editing is to analyze the document's purpose, audience, and use, and to verify all relevant information is included. This phase of editing is to review the document as a whole, not to review the prose line-by-line.

When performing a comprehensive edit, consider the following:

- Organization of each section and of the document as a whole
- Flow of information between paragraphs and sections
- Relevance of the content to the target audience
- Completeness of information

- Accuracy of information

## Copyediting

Copyediting, also referred to as *line editing*, reviews the structure and the integrity of individual sentences. Once the content of the document is established (through comprehensive editing), copyediting is conducted to examine the mechanics of the prose.

Copyediting requires a lot of attention and is a time-consuming task. Allow ample time to copyedit a document, including time to take breaks and to revisit the document at a later date.

Create a style sheet to aid in copyediting.  A style sheet defines abbreviations, capitalizations, spellings, etc. specific to a particular document and verifies the writing style is consistent throughout the document. Elements of writing used throughout all documents (such as trademarks or the use of bold on terms used as terms) are outlined in the company's style guide.

The following is an example of a style sheet used to write the cMat™ 5.0 User Manual (an excerpt of this manual is provided in Chapter 4: Deliverables).

**Spelling, hyphenation, capitalization**

- Business Process ID
- Business Unit
- Collaborative Material Maintenance
- Finish role
- Inspection Plans
- Material Groups
- Materials
- Org Unit
- Plans
- Post role
- Request
- Review role
- Role ID
- Role Type
- Sales Organization
- Sales Organizations
- Scenario ID
- Search function
- Task
- UOM

**Type Style**

- Role ID, Scenario ID, and Business Process ID names – bold and capitalize
- SAP Insert Modes and SAP Update Modes – all capital letters
- SAP table names – all capital letters
- Role Type names – capitalize

### *Mechanics*

Mechanics of a document refer to correct spelling, grammar, and punctuation. Basic writing mechanics and rules must be applied to any technical document. With any literature, the reader quickly loses confidence in the document if mechanical mistakes are discovered. If the reader cannot trust the writer to use correct grammar, why should the writer be trusted with other areas of the document, such as to provide accurate information?

When reviewing the mechanics of a document consider the following:

- Pronoun agreement
- Correct spelling and capitalization
- Verb tense (avoiding future or past tense)
- Complete sentences
- Numerous short sentences instead of a few very long sentences

### *Accuracy*

Accuracy of a document refers to the correctness of the information. This aspect of copyediting is different from comprehensive editing because accuracy does not verify that all information is accounted for. Instead, verifying accuracy in a document examines content errors, such as the following:

- Spelling and trademarks of product names
- Version numbers
- References to field names, button names, page names, etc.
- Cross references to sections within a document
- Links to external websites
- Consistency in abbreviations, names, dates, etc.
- Use of screen shots and illustrations
- Steps in a tutorial

### *Completeness*

Verifying the completeness of a document certifies that it is ready for production. This stage of editing examines the completeness of visual and verbal components. When examining the completeness of a document, verify the following:

- Completeness of the parts of the document, including the title, date, table of contents, and relevant sections
- Formatted screenshots
- Accurate headings
- Correct numbering

## Document Testing

Always test the document before it is released. Testing is best performed right before the software installation package is built to ensure no additional changes will be made in the software. During software testing, page names, navigational steps, or field names can often change (to fix issues discovered during testing).

The purpose of document testing is to verify the product and the document match identically. Inaccurate tutorials in the document are discouraging and frustrating to an end user and lower the usability of a document. While developers may review the document in its final stages, they rarely follow the tutorials step-by-step to verify accuracy. As the writer, march through each tutorial of the document, line-by-line, to verify every step is complete.

To test a document, follow the steps in each tutorial as if you were the end user. Put yourself in the shoes of someone with no or very little knowledge of the tool and examine the following:

- Tutorial steps are correct
- Screenshots match the software
- Text in the document matches screenshots
- Content is clear and logical

### Incorporating Users

If time allows and if the resources are available, have the intended audience participate in the document testing process. The audience can comment and provide useful feedback on the areas that are unclear, need more explanation, or are unnecessary. From usability testing, you can learn what the audience needs and wants out of a document. The more you know about the product, the more you assume the reader also knows and, in turn, the less detail you may include in the document. Consulting the target audience when writing a document can serve a helpful purpose of learning how much or how little knowledge the user has on the product. The exact intended audience may not be available. In such an instance, having a user with similar knowledge and skill level test the document suffices.

## Document Usability

Document usability refers to how easy or difficult it is for the reader to extract the desired information from a document. This is an element of writing that many technical writers overlook or do not consider. The document may be perfect in terms of its mechanics and content. However, if it is not easy to use, the reader either does not read the manual or spends more time searching for information than reading.

As the technical writer, you may have spent weeks composing a single document and, therefore, you have become very familiar – possibly too familiar – with the content. In addition, if highly technical and experienced with the product, the writer can easily forget how much he or she knows. Writers may become so familiar with the locations of buttons and menus their view of the product can become distorted.

There are many ways to increase the usability of a document. When writing and editing, consider the following:

- Search
- Scan
- Clarity
- Simplicity

## Search

Rarely do users read a document from cover to cover. They reference a document for information to complete a task and to spend as little time as possible reading and more time doing. Increase the usability of a document by including helpful search mechanisms, such as an index or a table of contents.

In order to make a document's search useful, you must use terms familiar to the user. Conducting a usability test can help you understand how to communicate with the audience by learning the terms, jargon, and vocabulary used by the audience during the test. It is also beneficial to research terms used by other software development companies to find common industry standards.

### *Index*

An index is located at the back of a document to cross reference key words with the pages in the document in which they appear. Indexing a document is tedious, but a completed index is highly useful to the end user. It is also a very familiar search tool to most readers. When picking up a book or any reference material with a specific topic in mind, readers often proceed directly to the back of the book to refer to the index because they know how to use it as well as realize its usefulness. Indexes are a very common, familiar, and useful means to search through a document.

When creating an index, put yourself in the shoes of the target audience. If new to the product, what would you search for? Index broad topics, such as "security," and specific objects, such as field names.

### *Table of Contents*

Since a table of contents is located at the beginning of a document, it is often the first layer of search performed by the reader. The table of contents is generated based on the document's headings, so it is imperative to write useful and informative headings. Consider the audience when writing headings. What information is the audience looking for? What is important about the product that the audience needs to learn? Writing useful headings to generate the table of contents is a very important and useful practice in order to increase the usability of the document.

### Scan

Readers do not read – they scan. The readers may not know exactly what they are looking for. Therefore, searching through the table of contents or referencing the index may not be the most useful means to find information. Since users often scan through a document, a writer can use various design tactics to draw the reader's attention to specific information. To help users scan a document, consider the following options:

- Make use of headings and sub-headings to break up text.
- Emphasize (by bolding, underlining, and/or italicizing) terms and definitions.
- Avoid using blocks of text. Large paragraphs are not only discouraging to a reader, but they lower the usability of the document because they are not easy to scan.

### Clarity

Clarity is one of the ultimate goals of technical writing. A reader wants to obtain information from a technical document as quickly and effortlessly as possible. If a document is not clearly written, the reader may spend more time deciphering and understanding the information than digesting it in order to perform the task.

### Simplicity

Simplicity is the golden rule of technical writing. While keeping ideas short and sweet, it is important to acknowledge simplified versus simple-minded writing (Shelton 20). The use of short and direct sentences promotes information retention. Using words of short length (i.e. ones that have fewer syllables) is also helpful for the reader to understand the content. The writer must maintain a healthy balance between undermining and helpfully educating the reader. The reader's focus is on digesting the content not deciphering the language.

### Audience

The audience greatly impacts the usability of a document. If the document contains too much detail and explanation of terms, a technical audience will spend more time sifting through the content to find the relevant information than reading the material. On the other hand, if a document does not provide enough detail, a non-technical audience will not understand the document enough to perform the task.

If the user must spend a lot of time reading and comprehending the material while defining terms and jargon, the usability of the document is low.

# Production

In this textbook, *production* encompasses document design and the delivery of the final product. Many software companies do not have resources dedicated to document production, so the design and production responsibilities often fall on the shoulders of the technical writer.

## Document Design

An important element of technical writing is making the final product as aesthetically as it is linguistically appealing. If a document is not visually appealing (too much block text, unreadable font, etc.) a user may become discouraged and may not even read the document. Writers must understand the context in which they are writing before design choices can be made.

Document design greatly contributes to the overall success of a document. It can increase usability and readability by providing a clean look to the page. After a series of usability testing and research projects, Sharon and Steve Gerson, creditable educators of technical communication, conclude “words are not only your concern. What you write is important, but how it looks on the page is equally important” (Gerson 238).

Many technical writers are fluent in document design programs, such as InDesign and Microsoft Publisher, to have the skill set to produce professional-looking documents beyond the scope of word processing programs. But if the only available program is Microsoft Word, for example, then make do with what you have. There are many great features of Microsoft Word that aid in producing beautiful documentation layouts for your company.

Style guides and specific document design concepts are beyond the scope of this textbook. However, there are a few design elements to be aware of when drafting and editing, such as the use white space, the importance of subject headings, and the use of bulleted lists instead of paragraphs.

### White Space

White space is often viewed as a negative design entity to a document. Technical documentation is the perfect place to be generous with white space. Text is very discouraging when there are many words on the page and the reader must exert great focus to read and digest the content in order to extract useful information. Readers are less overwhelmed when reading foreign subjects when text is separated by white space.

### Subject Headings

Another design element to help the reader focus on the content is breaking text into subject headings – much like how this textbook is organized.  Not only do subject headings break up the page to provide the audience with easier readability, but also the headings enable readers to find the subject they are looking for. Rarely does an audience read the entire technical document; readers scan by subject heading to identify useful information.

### Lists versus Paragraphs

Use lists where possible to break up the page and provide the reader with another type of text organization besides block text. If the organization of the text continuously changes, readers are likely to remain engaged in the document because their eyes do not become tired from reading as fast. Introduce some of these techniques whenever possible to keep the reader focused. Using lists instead of a paragraph also increases the usability and readability of the document because lists enable the reader to scan the text for information.

A good rule of thumb is to use lists instead of stringing elements together within a paragraph using commas. However, the over -use of lists can be counterproductive. Too many lists can cause the text to seem like an outline instead of a finished product.

## Producing the Document

Once the document is written, edited, and tested, it is packaged and delivered. There are various ways to package a document, including the following:

- **Electronic** – A document can be distributed electronically as a PDF file, for example. Rarely is an editable file delivered to the audience in order to protect the integrity of the document.

- **Print** – A hard, tangible copy that is physically delivered with the software installation CD.

- **Online** – A copy of the document that is imbedded in the software.

The method used may be greatly controlled by resources available. Electronic documentation (such as a PDF file) is far more cost effective than a printed document. However, a printed document is more professional looking than a PDF file.

A technical writer’s involvement in the production of a document is usually kept to a minimum. Most companies have a designer or production manager that produces the final document. The technical writer’s primary responsibility in the production of a document is to provide the designer with an electronic copy of the deliverable.

## Chapter Summary

The following key points have been covered in this chapter:

- The primary differences between a technical and a creating/academic writing process deal with time constraints, planning, and collaboration.
- Planning, drafting, revising and editing, and producing are the primary phases of the writing process for technical documents.
- The basic elements of planning a document are defining the purpose, analyzing the audience, defining the scope, and researching for content.
- The purpose, audience, and scope of a deliverable are essential to a clear and direct focus.
- No document has only one audience – audiences vary and range in technical skill levels and knowledge of the product.
- Comprehensive editing verifies the usability, organization, and completeness of a document while copy editing examines the document's mechanics and accuracy.
- Document testing certifies the document and the product match identically.
- The qualities of  documents that affect their usability are search, scan, clarity, and simplicity.
- Electronic, print, and online documentation are examples of how to deliver a document to the intended audience.

# Appendix A – Deliverable Checklists

## User Manual

- □ Determined audience(s)
- □ Determine scope
- □ Cover page
  - □ Document title
  - □ Product
  - □ Software and framework version
  - □ Date
  - □ Company address
- □ Table of Contents
- □ Overview
  - □ High-level description of the product
  - □ Prerequisite information
  - □ Explanation of how the product fits within a toolset
- □ Additional resources
- □ Getting started with all configuration steps
- □ Tutorials
  - □ Provide a brief overview of the tutorial
  - □ Use a “To” line to introduce the tutorial

- □ Number steps
    - □ Place steps in chronological order
    - □ Describe each object exactly as it appears in the software
    - □ Include relevant screenshots
    - □ Begin each step with a present tense verb
    - □ Include one step per line
- □ Reference
    - □ Menus, sub-menus, and menu options
    - □ Buttons and/or icons on the toolbar
    - □ Pages with description of function and purpose
    - □ Controls (buttons, fields, icons, etc.)
- □ Appendices
- □ Index

## Release Notes

- □ Title
- □ Product name
- □ Product version
- □ Previous product version
- □ Dependent products with required versions
- □ Consult with marketing department for existing material
- □ New Features
    - □ Name of new object (field, functionality, page, menu, etc)
    - □ Justification for the new feature
- □ Enhancements
    - □ Name of enhanced feature
    - □ Justification for enhancement
- □ Resolved Issues
    - □ Name of the resolved issue
    - □ Solution to the issue
    - □ Known issues from previous release
- □ Known Issue
    - □ Details of why the issue exists
    - □ Severity of issue
    - □ Work around (if applicable/existing)
    - □ When the issue will be fixed
- □ Optional sections
    - □ Installation
    - □ System Requirements
    - □ Frequently Asked Questions

## Product Descriptions

- □ Title
- □ Product name
- □ Product version
- □ Overview
    - □ Brief description of the product
    - □ How product fits into toolset (if applicable)
    - □ Statement that the product stands alone (if applicable)
- □ Features
    - □ Use action verbs
    - □ One entry per line
    - □ List primary features of the product
    - □ Do not use too much detail
- □ Diagrams
    - □ Flow diagram outlining high-level process
    - □ Diagram of how the product fits into a toolset (if applicable)

## Training Manuals

- □ Determined audience(s)
- □ Determine Scope
- □ Cover page
    - □ Document title
    - □ Product
    - □ Software and framework version
    - □ Date
    - □ Company address
- □ Table of Contents
- □ Overview
    - □ High-level description of the product
    - □ Prerequisite information
    - □ Explanation of how the product fits within a toolset
    - □ Purpose of training manual
- □ Intended Audience
    - □ Targeted reader
    - □ Assumed skill/technical level
- □ Prerequisites
    - □ Required prerequisites for the class
    - □ Recommended prerequisites for the class
- □ Objectives
    - □ Listed goals of the training class
    - □ Lead with present tense verb
- □ Exercises
    - □ Determine the "big picture"
    - □ Set mini-milestones
    - □ Provide an overview and purpose of each exercise
    - □ Pre-test for each exercise
    - □ Implementation of each exercise
    - □ Test of each exercise

## Quick Reference Guide

□ Fits to two pages

□ Written for an advanced or technical audience

□ Where to read more information

□ Outlines a single core task

# References

Basu, Anindita. “How to write badly.” Weblog entry. Write Technically. 4 October 2008. 7 September 2009 <http://writing-technical.blogspot.com/2008/10/how-to-write-badly.html.

Basu, Anindita. “The Story of a User Manual.” Weblog entry. Write Technically. 17 October 2007. 7 September 2009 <http://writingtechnical.blogspot.com /2007/10/story-of-user-manual.html >.

Bivins, Thomas H. Public Relations Writing: The Essentials of Style and Format. 6th ed. Boston: McGraw Hill, 2008.

Bonnett, Cara. “Instant Messaging Facilitates Quick Communication, Collaboration.” Duke Today. 9 September 2008. 12 September 2000 <http://news.duke.edu/2008/09/im.html>.

Bremer, Michael. The User Manual Manual: How to Research, Write, Test, Edit and Produce a Software Manual. Concord: UnTechnical Press, 1999.

Bremer, Michael. UnTechnical Writing: How to Write About Technical Subjects and Products So Anyone  Can Understand. Concord: UnTechnical Poress, 1999.

Bush, Don. “Technical Writing Is Unique.” Intercom (December 2001): 38-39,43.

Chatlien, Ruth Hall. "How to Write a Textbook." Weblog entry. Ruth's Visions and Revisions. 8 August 2008. 10 August 2009 <http://rhchatlienblog.blogspot.com/2008/08/how-to-write-textbook.html>.

Corbin, Michelle, Path Moell, and Mike Boyd. "Technical Editing as Quality Assurance: Adding Value to Content." Technical Communication 49:3 (August 2002): 286-300

Friedman, Matthew. "Advances useless without knowledge: As computer software and hardware gets increasingly complex, companies are becoming more aware of the importance of technical communicators." The Vancouver Sun 20 August 1998: E.3.

Gerson, Sharon J., and Steve M. Gerson. Technical Communication: Process and Product. 6th ed. Upper Saddle River: Pearson Longman, 2009.

Giammona, Barbara. "The Future of Technical Communication: How Innovation, Technology, Information Management, and Other Forces Are Shaping the Future of the Profession." Technical Communication 51:3 (August 2004): 349-366.

Grubb, Penny, and Rean Danuta. Writing a Textbook: How to Establish Yourself As an Academic Author. Plymouth, UK: How To Books, Ltd, 1997.

Houp, Kenneth, Pearsall Thomas, Elizabeth Tebeaux, and Sam Dragga. Reporting Technical Information. 11 ed. Oxford: Oxford University Press, 2006.

Jones, Dan, and Karen Lane. Technical Communication: Strategies for College and the Workplace. New York: Pearson Longman, 2008.

Johnson, Tom. “Documentation Usability: A Few Things I’ve Learned From Watching Users.” Weblog entry. I’d Rather Be Writing – a blog about the latest trends in technical communication. 28 April 2009. Wordpress Consulting. 1 August 2009 <http://www.idratherbewriting.com/2009/04/28/documentation-usability-a-few-things-i%e2%80%99ve-learned-from-watching-users/>.

Johnson, Tom. “Quick Reference Guides: The Poetry of Technical Writing.” Weblog entry. I’d Rather be Writing – a blog about the latest trends in technical communication. 6 July 2008. Wordpress Consulting. 1 August 2009 http://www.idratherbewriting.com/2008/07/06/quick-reference-guides-the-poetry-of-technical-writing/>.

Johnson, Tom, and Ben Minson. “Quick Reference Guides: Short and Sweet Documentation.” Weblog entry. I’d Rather Be Writing – a blog about the latest trends in technical communication. 10 April 2009. 1 August 2009 <http://www.idratherbewriting.com/2009/04/10/quick-reference-guides-short-and-sweet-documentation/>.

Kilpatrick, Kelly. “Technical Writing for the Layperson.” Weblog entry. Shanghai Tech Writer. 13 November 2008. 6 September 2009 <http://www.shanghaitechwriter.com/2008/11/13/technical-writing-for-the-layperson/>.

Lannon, John M. Technical Communication. 11th ed. New York: Pearson Longman, 2008

Oxford American Thesaurus. 2nd ed. 2008.

Lepionka, Mary Ellen. Writing and Developing Your College Textbook: A Comprehensive Guide to Textbook Authorship and Higher Education Publishing. Gloucester: Atlantic Path Publishing. 2008.

Levin, Susan. "Webinars Increase Communication Skills." Weblog entry. Market Yourself as a Speaker: Grow your Business and Income Thru Speaking. 10 February 2008. 12 September 2009. <http://www.speakerscommunity.com/blog/2008/02/10/webinars-increase-communication-skills/ >.

Lidwell, A., Holden, K., & Butler, J. (2003). *Universal Principles of Design*. Beverly, MA: Rockport Publishers, Inc.

Lolloway, Brian R. Technical Writing Basics: A Guide to Style and Form. 4th ed. Upper Saddle River:  Pearson Longman, 2007.

Martinez, Diane, Tanya Peterson, Carrie Wells, Carrie Hannigan, and Carolyn Stevenson. Technical Writing. New York: Kaplan Publishing, 2008.

Morris, Randa. "Instant Messenger and the Communication Revolution." Web page. 12 September 2009  <http://www.life123.com/technology/internet/instant-messenger/instant-messenger-and-the-communication-revolution.shtml>.

Postava-Davignon, Christi-Anne, Candice Kamachi, Cory Clarke, Gregory Kushmerek, Mary Beth Rettger, Pete Monchamp, and Rich Ellis. "Incorporating Usability Testing into the Documentation Process." Technical Communication 51:1 (February 2004): 36-44.

Rathbone, Robert R. "Growth of the Technical Writing Profession." Intercom (February 2002): 26-28.

Rude, Carolyn D. Technical Editing. 4th ed. New York: Pearson Longman, 2006.

Merriam-Webster's Collegiate Dictionary. 11th ed. 2007

Sandoval, Leslie. "Technical Writing 101: A Crash Course." Crossroads 14:3 (Spring 2008): 26-27.

Selting, Bonita R. "Conversations with Technical Writing Teachers: Defining a Problem." Technical Communication Quarterly 11:3 (Summer 2002): 251-266.

Shelton, James H. Handbook for Technical Writing. Lincolnwood: NTC Business Books, 1994.

Slattery, Shaun. "Technical Writing as Textual Coordination: An Argument for the Value of Writers' Skills with Information Technology." Technical Communication 52:3 (Aug 2005): 353-360.

Steele, Jeffrey. "The (Technical) Writing's on the Wall for this Group." The Chicago Tribune 30 July 2000: 3.

Take 20: Teaching Writing. Dir. Todd Taylor. Perf. Linda Alder-Kassner, Cheryl E. Ball, Dave Bartholomae, Patricia Bizzell, Bill Condon, Ellen Cushmen, Cheryl Glenn, Brian Huot, Erika Lindemann, Andrea A. Lunsford, Paul Kei Matsuda, Christine McQuade, Don McQuade, Mike Palmquist, Malea Powell, Nedra Reynolds, Mike Rose, Jacqueline Jones Royster, Raul Sanchez, John Schilb, Nancy Sommers, and Howard Tinberg. Bedford/St. Martin's, 2008.

"Writing a training manual – tips and templates." Weblog entry. Helpscribe. February 2009. Helscribe. 30 August 2009 <http://www.helpscribe.com/2009/02/writing-training-manual-tips-and.html>.

U.S. Copyright Office. 5 September 2009. United States Copyright Office. 6 September 2009 <http://www.copyright.gov/ >.

USPTO. 4 September 2009. United States Patent and Trademark Office. 2 September 2009 <http://www.uspto.gov/>.

Varchaver, Nicholas. "The Perils of E-Mail." *Fortune*. 17 February 2003: 96-102.

Weber, Jean Hollis. "Escape From the Grammar Trap." TechWr-l 03 May 2007. 8 August 2009 <http://www.techwr-l.com/node/25/print >.

Wu, Susan. "If Only Technical Writing Was This Simple." Weblog entry. Shanghai Tech Writer. 9 May 2008. 6 September 2009 <http://www.shanghaitechwriter.com/2008/05/09/if-only-technical-writing-was-this-simple/>.

Wu, Susan. "How Well Do You Work With Your SMEs?" Weblog entry. Shanghai Tech Writer. 14 June 2008. 6 September 2009 <http://www.shanghaitechwriter.com/2008/06/14/how-well-do-you-work-with-your-smes/>.

Wu, Susan. "The Challenges of Being a Technical Writer." Weblog entry. Shanghai Tech Writer. 31 January 2008. 6 September 2009 <http://www.shanghaitechwriter.com/2008/01/31/the-challenges-of-being-a-technical-writer/>.

Wu, Susan. "Writing is Like Working on a Jigsaw Puzzle." Weblog entry. Shanghai Tech

Writer. 23 April 2008. 6 September 2009 <http://www.shanghaitechwriter.com/2008/04/23/writing-is-like-working-on-a-jigsaw-puzzle/>.

Zeleznick, Julie M. Technical Writing: What it is and How we do it. New York: Learning Express. 1999.

# ELIZABETH M. WARNKE

2622 Lance Ridge Way, Buford, Georgia 30519 678.977.1560

Elizabeth.Warnke@gmail.com

## EDUCATION

Master of Arts, Kennesaw State University, Kennesaw, GA
**Professional Writing, Applied Writing concentration** **2009**

Bachelor of Arts, Wheaton College, Norton, MA
**Major in Physics, Minor in Mathematics and Spanish** **2005**

## RELATED EXPERIENCE

BackOffice Associates, Norcross, GA
**Product Manager** **February 2009 – Present**

- Managed the release of the internal document web application.
- Developed scope of each release, wrote test scripts, conducted testing, and helped developed the web application for documentation.

BackOffice Associates, Norcross, GA
**Technical Writer** **July 2005 – Present**

- Established documentation standards, documentation process, and documentation policies for company.
- Maintained the document repository, including document control and security reconstruction.
- Created the following document types for all software product releases: user manuals, installation manuals, release notes, and product overviews.

## OTHER EXPERIENCE

Office of Human Resources, Wheaton College, Norton, MA
**Student Assistant** **2001 – 2005**

- Logged and processed incoming resumes.
- Organized data and created lists and visuals for employee workshops.

Department of Theatre Studies & Dance, Wheaton College, Norton, MA
**Box Office Manager** **2003 – 2005**

- Designed and produced tickets and programs for shows.
- Kept records of reservations and coordinate box office workers and ushers.
- Sold tickets before shows

## MEMBERSHIPS

Golden Key International Honour Society **2007 – 2009**