

Title:

Successful Documentation Projects - Part 2 of 3 - 'Specifying'

Word Count:

1803

Summary:

So you're responsible for managing a documentation project. You know who your audience is, what they're trying to achieve, how the product enables them to achieve it, and what the audience requires of the help. Now it's time to spec out your intentions.

Keywords:

documentation projects

Article Body:

So you're responsible for managing a documentation project. You know who your audience is, what they're trying to achieve, how the product enables them to achieve it, and what the audience requires of the help. Now it's time to spec out your intentions.

NOTE: This is the second in a series of three articles outlining the key elements of a good user documentation process. (To read the first and third articles in this series, go to <http://www.divinewrite.com/docoprocess1.htm> and <http://www.divinewrite.com/docoprocess3.htm>.)

State your goals

Generically speaking, your goal statement should indicate that you hope to create a suite of documentation products that will satisfy audience requirements. Specifically, you'll have a number of sub-goals. (TIP: It may help to remember that the goals you set here will need to be used to measure the success of your product through your own in-house testing as well as through evaluative user research.) Such sub-goals may include:

- Ease of use
- Accessibility
- Helpfulness
- Accuracy
- Relevance
- Comprehensiveness

- Adherence to style guidelines
- Correct spelling and punctuation

Write your Concept Specifications

Your goals set, you can start to contemplate what you're going to produce. The first step is to create some concept specifications. Simply put, concepts specs are very high level overviews of what you're proposing to produce. For example, your concept spec for the online help might state that you will be producing a product that allows the user to access information using a TOC, an Index, and a Find. It might suggest some possible GUI features of these elements, but it will not lay down requirements; just possibilities. The concept spec for your manuals might state that they will be professional looking, will contain many professionally drawn pictures, will have adequate white space, will be stylish, will be divided into chapters to match the task oriented nature of the online help, etc.

Generally, the product you're proposing could be implemented in a number of different ways. You should write one or more concept spec(s) for:

- what components the documentation suite will consist of (online help, printed manuals, tutorials, overviews, etc.) - "Documentation Products Concept Specification"
- the types of information your documentation will contain (e.g., the structure of the TOC, are you going to follow minimalism practices?) - "Documentation Content Concept Specification"
- the functionality and user interface of your documentation suite (e.g., how it will work and how the audience will interact with it) - "Online Help User Interface Concept Specification", "Printed Documentation User Interface Concept Specification", etc.
- the delivery method (how you will deliver the help to users and how you'll update it)
- what languages the documentation will be produced in

Design some possible implementations

Now that you've decided roughly what you'd like to produce, you can design some possible implementations of it. Your designs will be very high level and they may not actually work (they may actually be just paper prototypes).

With most other considerations already finalised through your user requirements research, these implementations should only differ as a result of:

- the technologies behind them

- the tools used to create them
- the overall look and feel

You need to learn as much as possible about these things, in order to determine what is actually possible, successful, effective, etc. You should be aware of current trends, literature, white papers, etc. This information can be obtained from a variety of sources. Some good places to start include:

- List servers
- Conferences
- Books
- Other publications
- Other writers
- Other products

Conduct usability testing on your prototypes

Model (prototype) your designs for the decision makers and audience samples. This allows you to pick the best features from each design (and to determine priorities for them). Select a design (or merge multiple designs) that you believe best satisfies user requirements. This process may be iterative. At the end of this stage, you should know enough to detail exactly what you'll be producing (including what help platform and tool you'll be using).

TIP: For details on possible research methods, take a look at Managing Your Documentation Projects by Hackos (1994) esp. pp.446-447, User and Task Analysis for Interface Design by Hackos & Redish (1998), Social Marketing: New Imperative for Public Health by Manoff (1985), Designing Qualitative Research 2nd Edition by Marshall & Rossman (1995), and "Conducting Focus Groups - A Guide for First-Time Users", in Marketing Intelligence and Planning by Tynan & Drayton (1988).

Write your Requirements Specifications

Requirements specifications detail exactly what you must end up with. These specifications should contain as much detail as possible about the features and functionality of the documentation product (not how you'll go about building it).

Requirements specs are basically an evolution of your concept specs. Once you begin work on your requirements specs, the concept specs are effectively frozen. You should write one or more concept spec(s) for:

- what components the documentation suite will consist of (online help,

printed manuals, tutorials, overviews, etc.) - "Documentation Products Requirements Specification"

- the types of information your documentation will contain (e.g., the structure of the TOC, are you going to follow minimalism practices?) - "Documentation Content Requirements Specification"
- the functionality and user interface of your documentation suite (e.g., how it will work and how the audience will interact with it) - "Online Help User Interface Requirements Specification", "Printed Documentation User Interface Requirements Specification", etc.
- the delivery method (how you will deliver the help to users and how you'll update it)
- what languages the documentation will be produced in

Estimate Project Duration & Resources

Once you've completed the requirements spec stage, you should know enough to accurately estimate the duration and resource requirements for the remainder of the project. You should also update the "Documentation Project Plan" document with this information.

Estimating is always a difficult process, and there's not really any sure-fire way of getting it right. Mostly it depends on the job and your experience. However, following are some guidelines that might help you.

If you have records from previous projects, you might simply be able to estimate project duration based on these. You should try to compare the old subject material and topics with the new to make sure that the old times will be applicable to the new project. On p.174 of *Managing Your Documentation Projects* (1994), Hackos provides some potentially useful guidelines for comparing the complexity of various documentation projects.

If, on the other hand, the project is entirely new, you will have no records to use as a guide (unless you have managed a similar project in the past). In this situation, project estimates will be very difficult to make.

One possible method for estimating is:

1. Compile a list of tasks, and record how many there are in your list.
2. Compile a list of concepts that must be documented, and record how many there are in your list.
3. From your list of tasks, select 10 that are representative of the rest

(in terms of complexity, expected length, status of the relevant development, etc.), and of the same granularity (e.g., you can write a single topic for each).

4. From your list of concepts, select 3 that are representative of the rest, and of the same granularity (e.g., you can write a single topic for each).
5. Estimate the number of pages per topic.
6. Document these tasks and concepts as a trial, ensuring that you track:
 - the total time taken to complete each topic.
 - the portion of this time that was due to product change or indecision.
 - the number of pages per topic.
 - the number of extra, unexpected, but necessary, topics you became aware of as a result of the documentation. Keep a separate record of the number for both task and conceptual topics.

TIP: Make the most of your trial doco. Even though you've chosen a design through design prototyping, you can use your documentation sample to test the usability of your documentation approach. By presenting the sample to an audience sample, you can determine whether you're heading in the right direction with your doco (i.e. whether you have interpreted and implemented your user research results correctly).

7. Determine the average time taken per page for task and for conceptual topics.
8. Apply this average to the rest of the topics in the project. (Topics written early in the project normally take longer due to lack of information and a higher number of technical issues. This means topics written later in the process will probably take less than the average calculated here. However, this will normally be offset by the extra time product changes will incur during the project life-cycle.)
9. Estimate the time per subject area based on the average time per topic.
10. Estimate the number of extra, unexpected, topics that will likely become necessary during the course of the rest of the project.
11. Allow for training, work prac maintenance, holidays, sick days, meetings, usability testing, production (approx 6 weeks turnaround time for printing a 1000 page manual, including proofing), evaluation, and evaluative

testing. Each of these elements will vary according to the nature of the project, and they will tend to take far less time than the actual writing. That is why specific guidelines are not provided as they are for writing.

Figure out how long you actually have to do it, then how many writers you'll need to get it done during this time. Draw up a project schedule using something like Microsoft Project, identifying useful milestones and project deadlines. Some of your milestones might include:

- Prototype Testing Complete
- Work Pracs Written
- Design Specs Written
- First Draft Complete
- Second Draft Complete
- Localisation of Second Draft Complete
- Final Draft Complete
- Localisation Complete
- Documentation Ready for Release
- Production Complete
- Project Evaluation Complete
- Post-release Usability Testing Complete

It is important to note that you will have milestones before this point, but because they occur prior to the formal scheduling stage, they don't need to be included in this schedule.

Write Work Pracs & Design Specs

Along with user research, work pracs and design specs are perhaps the easiest project elements to overlook, especially for a small team. However, even within small teams, it is helpful to maintain both.

Work pracs are for ongoing things, that affect the day to day working environment of the team (e.g., How to use your documentation tool, How to release your help, a style guide, etc.). Design specs are for documenting one-off things like how we actually plan to go about this thing. This will include such information as what tools we'll be using, what each will do, and the mechanics of how it all fits together. e.g., How the VSS project will work, how everything should be managed, multi-user issues, how it will be localised, etc.

To be continued... See part 3 of this article
(<http://www.divinewrite.com/docoprocess3.htm>) for information on writing your user documentation.

