

Planning user documentation - a guide for software project managers

By Carol Johnston.

Introduction

A Guide to the Project Management Body of Knowledge (PMBOK® Guide)–2000 Edition is the main sourcebook in the project management field. Whilst it covers Project Communications Management, it doesn't extend to user documentation.

This article seeks to provide guidance for project managers as to how the user documentation process fits in with the overall project planning. It examines:

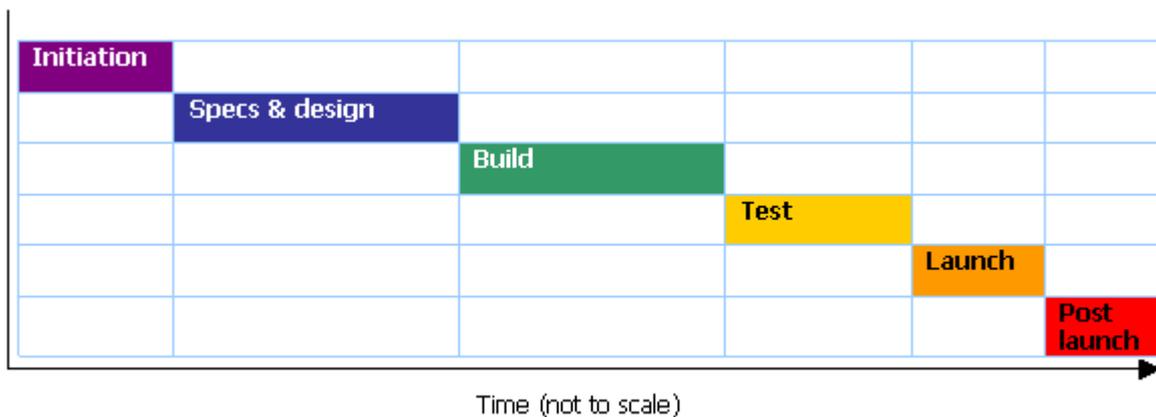
1. the traditional way documentation is approached and how it impinges on project planning
2. the effects of making changes to this traditional approach.

What do we mean by documentation?

When talking about documentation for a software application, most people think of the traditional paper-based or online manual. In this article, we will be looking at ways of taking this traditional approach forward. The term ‘user assistance’ then becomes more useful, implying ‘anything that makes the user’s life a bit easier’. More on this later...

Your project plan

A typical plan for a software project might look like this:



A Traditional documentation project plan

A **traditional** documentation project comprises the following stages:



The vertical grey bars on the above diagram represent reviews of the work done. These bars are not to scale. Reviewing can take a lot longer than you might think and must be done by somebody with expert knowledge of the software.

Prototype

The prototype comprises a structured set of empty topics (pages of information) with a prototype look-and-feel. For each different type of topic, one is populated with content as a sample.

The review is to determine completeness and suitability of the structure together with suitability of the look-and-feel, writing style and writing conventions.

First Draft

During the first draft stage, the author writes the content for all the topics. This is the longest stage and takes the longest to review.

The review is to check for technical accuracy of all the content.

Second Draft

The second draft stage is for the author to make the necessary changes, reported from the first draft review.

The review is to confirm that this has been done and to request any minor tweaks still outstanding.

Final Draft

This final stage is to follow up on the minor changes picked up at second draft review. The result is the final deliverable.

So where does the documentation fit in with my project plan?

Traditionally, the documentation comes in towards the end of the **testing stage** of your project. The software is more or less stable and you're close to launch. Having seen the stages involved in the documentation, you may be thinking that it's a lot to cram into this stage. You'd be right! This is why many project managers see documentation as that necessary nuisance that takes up valuable manpower to create and review just when deadlines are looming.

So it is also vital to consider the effects of the documentation at the **initial planning stage**. In this first stage of your project, your planning will include such things as:

- Timescales and resources
- Roles and responsibilities

Timescales and Resources

When thinking about timescales, don't forget the time required for the documentation! It may seem obvious, but effective documentation takes time to develop. As seen in the typical documentation project plan above, the elapsed time can be much longer than just the writing time due to the review process.

Depending on the scale and complexity of your software, you may have to consider splitting the documentation between different authors. Authors working in parallel can speed up the process, but be warned – depending on the tool(s) used to create the documentation together with the associated workflow this may not be possible. In addition, there is the extra issue of keeping the documentation consistent across the authoring team.

As well as providing the writing resources, it is just as important to provide adequate resources for the reviewing and a documentation project manager to coordinate writers and reviewers. All this must be factored into the cost.

With so much to do in the testing stage of your project plan, the inclusion of the documentation at this stage and the drain on expert staff for reviewing presents a real project risk if not properly planned and resourced.

Roles and Responsibilities

Who should write the documentation?

If your organisation has a specialist documentation department then your solution is simple. So what are your options if this is not the case?

The programmers themselves

Seems logical. They certainly know the product inside out. But beware. The programmers are experts in programming – not necessarily in communicating! They may well assume a level of knowledge that the end-user does not have. Not only that but their enthusiasm for their creation often gives them an understandable tendency to explain every last technical feature in detail – focussing on product functionality rather than answering users' questions.

A contracted technical author

This is often a good solution, though it very much depends on the rest of your project plan. In the **build** stage of your project plan, you may be setting aside long stretches of time on the coding of the software between tests. Unless there are other projects to work on, a technical author can be left idle and compromise your budget.

Outsourcing

Contracting an external organisation to project manage and develop your documentation means that you will only get charged for the days spent managing and writing. This solves the problem of idle time, but does have a couple of disadvantages. One is that the author spends most of the actual writing time off-site so direct communication may be more difficult. The other disadvantage is that it may not be possible to guarantee that one particular author will be available if timetables and deadlines change.

Who should review my documentation?

You may want or need more than one reviewer. As already mentioned above, all the documentation at every stage must be reviewed by at least one expert in the software. This ensures fitness for purpose and technical accuracy. In addition, it is often useful to get other relevant people to review parts of the documentation. One example is if you have access to some helpful end users. Their comments can be very valuable. Another example is if there is to be a training programme for the software. The training staff can provide useful feedback.

Who should manage my documentation?

As stated previously, the documentation project manager is necessary to co-ordinate the writing and the reviewing process. Unless your programmers are writing your documentation, many questions will arise on the author's part about the details of how the software is to be used. So part of the project manager's role includes co-ordinating these questions and their answers between the author and the programmers.

If you are outsourcing, the external company manages the delivery of the drafts, the receipt of review points and the author's questions for the programmers. However, there should still be a person named as the documentation project manager in your organisation to provide a single point of contact for the external document project manager. Without this role, review points get fed to the external company in dribs and drabs and sometimes contradict one another.

Spreading the authoring work more evenly

The traditional close-to-launch timing of the documentation presents, as mentioned already, a project risk. To reduce the amount of time and resource required at this busy time, there are a number of tasks that your technical author can do **earlier** on in the project to reduce the project risk.

Look and feel of the Help system

These days, authoring tools for technical writers are making it possible to create Help systems with a customised look-and-feel. Software houses are now getting more interested in their Help systems looking better than the rather grey-looking standard Help viewers (for example WinHelp and HTML Help) and more in line with the software itself. Creating a customised look-and-feel takes time. But this is one of tasks your technical author can be doing in the **design** stage, as soon as the software look-and-feel is agreed.

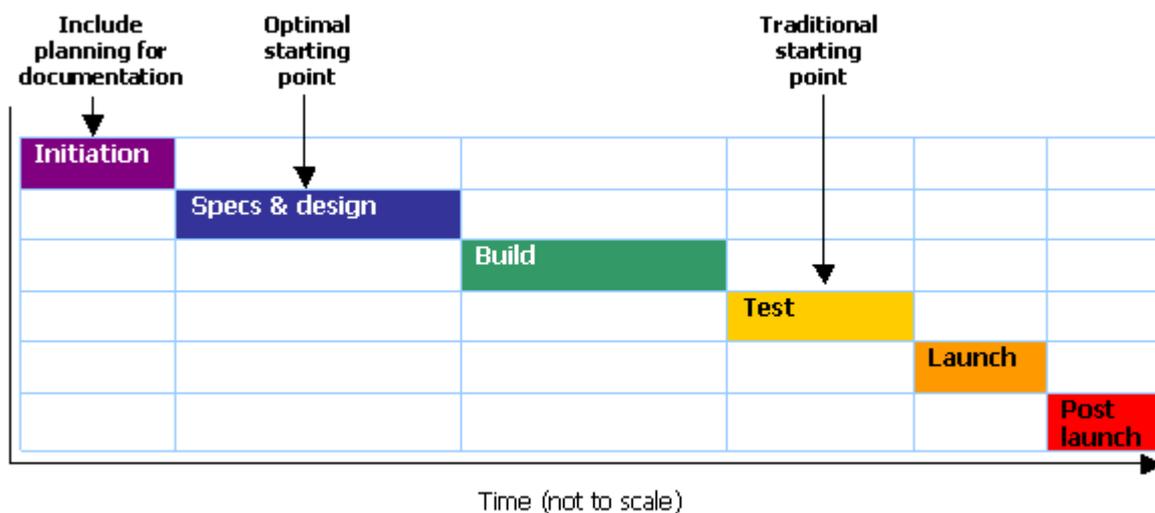
Structuring the Help system

Depending on the complexity of the software, it may be possible for the technical author to sketch out a rough structure for the topics in the Help system after the **design** stage. This would be a flexible work-in-progress structure, but would at least set out a draft of the types of information to be documented.

Conceptual topics

Generally speaking, the majority of Help topics in a Help system are procedural. They tell the user how to do something by giving detailed step-by-step instructions on using the interface so, in the main, are best left until the software is more or less stable. However, there will also be some information for the user that is of a conceptual nature. It may be possible for your author to document some such concepts during and after the **design** stage.

Summary of where your documentation fits into your project plan



Getting more value from your technical author

Specifications and design stage

It may not be something you have considered, but since good technical authors have expertise in designing information, they can also provide **useful input** at the **specifications and design** stage. Getting your technical author on board at this stage can help make the user interface more usable. This reduces your technical support costs and may reduce the time it takes to document the software. It may be something as simple as labelling or ordering the fields on the dialogs in a more helpful way. Moreover, authors with experience in writing for screens will know what sorts of layouts will and won't work. They can provide valuable input for decisions on the structuring and the look-and-feel of the interface. And remember that they are very much user-focussed – providing balance to what may be an over-enthusiastic graphics design team! You can think of the technical author as the representative of your average user as the specifications are decided and the design progresses.

Embedded Help

If your technical author is contributing expertise during the **specification and design** stage of the project, it may well be that the traditional paper manual/online Help 'documentation', created at the very end, is reduced in scope or even unnecessary. In other words the 'necessary nuisance that takes up valuable manpower to create and review just when deadlines are looming' is all but eliminated. The user interface can contain its own 'user assistance'. This kind of approach is known as Embedded Help. Exactly how and where the assistance is provided is part of the specification and design process. For example, one successful technique is to have a dedicated (but shrinkable) part of the interface for displaying context-sensitive assistance.

Build stage

If present at all relevant meetings, the number of questions your technical author will have for the programmers when writing the Help content will be reduced, saving precious time near the end of the project.

Testing stage

If you are planning to perform usability testing, your technical author can participate in designing and giving the tests and in interpreting the results.

Post rollout

If you are outsourcing, the external organisation may provide a service for logging and collating user feedback, reporting it to you and updating the Help where appropriate.

Summary

Providing online assistance for software is important since it can reduce costs of technical support and make life easier for your users. However, documenting software is not a trivial task. It requires time and resources for writing and reviewing that must be included in the overall project plan.

The documentation work is traditionally implemented after the software is reasonably stable – during the testing stage. However, getting your author on board earlier spreads the work more evenly, reducing project risk. Moreover, including your author in design decisions and usability testing can improve the user interface.

Let us know what you think

We welcome your comments and thoughts on this article.

About Cherryleaf

Cherryleaf helps companies increase their sales by having user information that is easy to access, understand and use. We have a unique blend of information design, writing and teaching skills, as well as technological expertise. We combine these skills to develop documents, information systems and affordable content management solutions that bring people, business processes, content and technology together.

For developers of software

Cherryleaf can help you retain your customers by having good user documentation. We can create, develop, resource, manage and maintain your online Help, user manuals, performance support systems and IT Systems documents. We are well regarded in our specialist field, which means we speak at quite a few conferences. Sometimes we're working with people to give them the capabilities to do it themselves - training people in the most popular tools for creating Help file or helping them to recruit the right staff. Other times we're delivering to a fixed price or budget.

For everyone else

We can help you to improve your performance by enabling you to develop better documents in a more efficient way. We deal with procedures, call-centre information, business continuity, training courseware and sales proposals. We can give you the power to re-use existing "chunks" of content when developing new documents, and the ability to guide your staff on how to write a specific type of document (based on your expertise and knowledge). The result: documents that are easier to develop, access and use.