Writing and Reviewing Technical Documents

Andreas Rüping

sd&m software design & management GmbH & Co.KG

Thomas-Dehler-Straße 27
D-81737 München, Germany

e-mail: rueping@acm.org

Introduction

Writing a good document takes time. First of all, it takes quite some time on the author's part. But for a document to become a good document, it needs to be reviewed just like software needs to be tested. And therefore, a good document takes time on the reviewers' part as well.

Guidelines for the Readers

This paper presents a collection of patterns that are meant to improve the process of writing and to make the review of technical documents a positive experience both for the author and for the reviewers. This includes all forms of reviews, such as meeting reviews or distribution reviews or both. The patterns address organisational issues as well as social or communication issues.

Although these patterns were observed in software engineering projects, they may equally apply to the documentation process in other technical domains.

Various pattern languages have already been published that deal with related issues, such as review culture, communication issues, or teamwork. It therefore comes as no surprise that many patterns in this paper overlap with other people's work. Jim Coplien's pattern language for writers' workshops [3] deserves special attention since the review principles he describes underlie several patterns in this paper.

This paper builds upon a number of existing patterns and pattern languages. Some patterns here are variations of patterns that have been described before. These patterns are included in this paper nonetheless for the following reasons:

- The collection of patterns in this paper would be incomplete if some very important patterns were missing from it.
- The patterns presented here differ slightly from earlier variants in other papers since the focus has shifted; therefore they contain some added value.
- The inclusion of these patterns in this collection gives evidence that they have a broader significance than originally stated, since they are useful in a wider context than the pattern language in which they were first included.

Copyright © 1998 by Andreas Rüping. Permission is granted to EuroPLoP '98 to make copies for conference use.

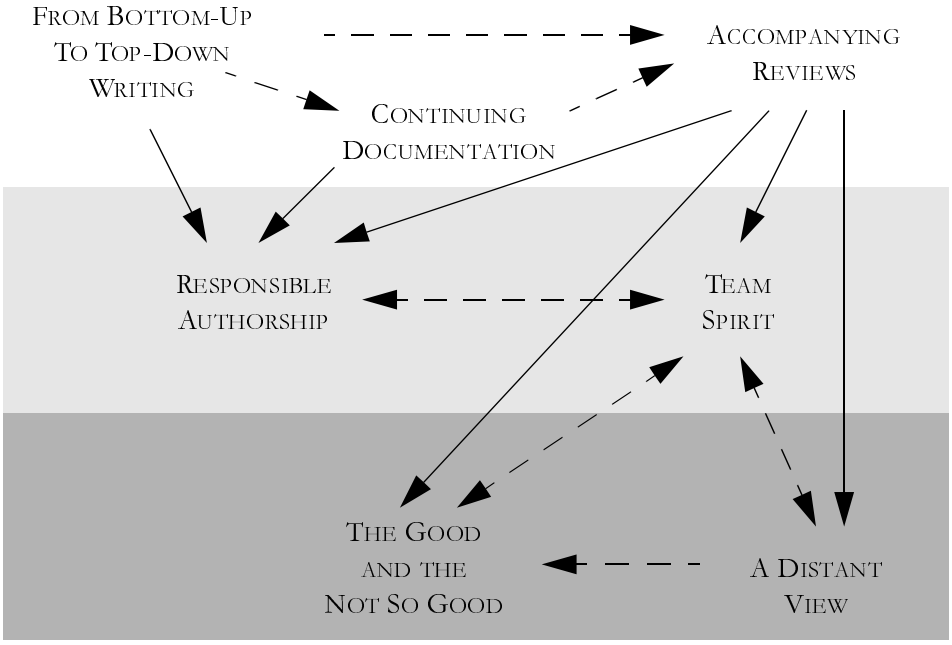
Because this paper focuses on the underlying principles of the writing and reviewing process, rules concerning the actual structure and the layout of technical documents are beyond its scope, but are dealt with in a separate paper [7]. This paper builds a bridge between the patterns in that paper and the existing pattern languages on teamwork and review culture.

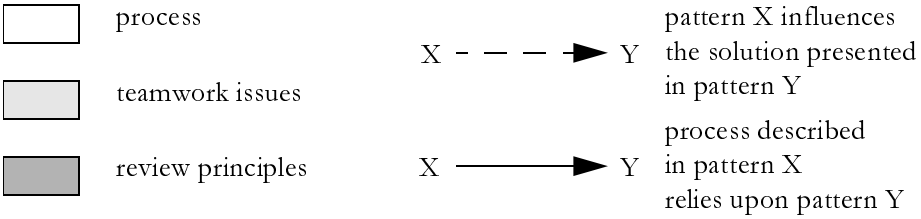
Each pattern in this paper consists of four parts. The problem section presents a question concerning the writing or reviewing process. The forces section describes the driving forces behind possible solutions. The solution section explains how the forces can be resolved. The discussion section gives additional information, and in particular elaborates on the relationships to other patterns.

Together, the problem section and the first paragraph from the solution section form so-called pattlets — thumbnail sketches that give a quick overview of each pattern. The pattlets are printed in bold face. The other sections are necessary to fully understand the rationale behind the patterns and the relationships to other patterns.

The patterns presented in this paper can be grouped into process patterns, patterns addressing teamwork issues, and patterns on review principles. The following diagram gives an overview.

Overview





1 From Bottom-Up to Top-Down Writing

Problem

How can the process of writing technical documents be organised?

Forces

It is well known that software development doesn't start with programming. Rather, software developers begin with a requirements analysis and then make a design. Likewise, the development of a document doesn't start with the actual writing.

When starting a document, authors often have only vague ideas of what they want to write or what they are supposed to write. For example, in the early stages of a software engineering project, the project's technical details may still be unclear and so cannot be documented yet.

Setting up the overall structure of a document too early can hinder the flow of ideas. Once the sections and subsections of a document are defined, the risk of missing aspects which fall outside the tentative structure is high.

Finished documents need to be well-structured and well-organised. A document structure cannot be set up impromptu, however, but needs to develop and to go through iterations as the text matures.

Good ideas take time. Creative processes must extend over several days to allow ideas to develop and mature.

Solution

The process of writing technical documents should go through several stages in which material is first collected in a bottom-up style, and then the evolving document is refined in a top-down style. Throughout all stages, creative activities must be given enough time.

The individual stages of the writing process are the following:

- In a brainstorming stage the author collects material that might belong in the document. This can be done with hand-written notes or on a computer.
- The following structuring stage represents the bridge from bottom-up to top-down writing. In the structuring stage the author sets up an initial document structure, considers material and assigns it to the individual sections, and maybe discards unnecessary material.
- In the writing stage, both the material and the initial structure are continually refined; what used to be mere ideas in the brainstorming stage are now worked out to become elaborate textual building blocks.
- Finally, a document must be polished with respect to language, formatting etc.

The brainstorming stage and the writing stage in particular involve a lot of creativity on the author's part. It is therefore important that both stages extend over at least a couple of days to allow ideas to develop in the back of the author's mind.

Discussion

In some cases, when the purpose of a document is clear from the start, collecting material in the brainstorming stage, and pinning it down, appears to be a very easy task, taking maybe only a few hours. Still, the time span for the brainstorming and writing stages should not fall short of a couple of days, even if only one hour per day is spent on the document.

Documents can profit a lot from reviews. The switch between bottom-up-writing and top-down-writing, and the completion of the final polishing stage, are important milestones for reviews (see ACCOMPANYING REVIEWS).

2 Continuing Documentation

Problem

When should documentation be written?

Forces

The process of writing is an important means of validation. While writing a document, an author will notice things that are imperfect or incomplete about the material. Writing documentation provides feedback on what is being documented, like, for instance, a software design.

Typically, technical documents are written in projects where deadlines need to be met. If the documentation is started only a few days before the deadline, the person responsible for the documentation will probably run out of time. In any case, there will not be enough time for several iterations for improving the documentation, particularly if a major revision is necessary.

Documentation doesn't come for free or happen automatically. It is important to understand that high-quality documentation requires time and effort.

Solution

Documentation should start while the work being described is still in progress, and should continue as the work goes on.

Project documentation should be written in parallel to all other project activities. The documentation should reflect the actual state of a project:

- The documentation must be continued as the project goes on.
- The documentation must be updated whenever the subject of documentation changes.
- While writing, the author should check the documentation for problems and inconsistencies. The observation that parts of the documentation are problematic may suggest that it is the subject of documentation that really is problematic.

To make continuing documentation possible, documentation must be considered an integral part of a project that requires personnel capacity and a budget just like all other project activities.

Discussion

Those parts of the documentation that are partially complete can be made available to other team members as the project goes on. This way, team members can give feedback concerning the available documentation. This immediate feedback can become part of the review process that accompanies the writing process (see ACCOMPANYING REVIEWS).

Charles Weir's patterns for designing in teams include a pattern that presents a technique to implement CONTINUING DOCUMENTATION. This pattern suggests that all team members make AD-HOC CORRECTIONS [8] on a printed copy of the documentation in order to to keep the documentation up-to-date.

3 Accompanying Reviews

Problem

How can authors receive the right feedback at the right time?

Forces

Hardly anybody manages to write perfect documents without help from other people. However, readers are easily upset when they have to plough through a document of poor quality.

Different aspects of a document need to be reviewed, including the contents, the structure, the layout, or the spelling. The importance of these different aspects varies depending on the actual state of the document. A review of a document in an early stage need not be concerned with the spelling. When a document is almost finished, it is to probably to late to change its overall structure.

Some authors prefer to incorporate feedback concerning structure, contents and presentation separately.

Problems concerning a document's scope or its overall organisation might make a major revision necessary for which it is probably too late when the document is almost finished. Even if it is not too late, these problems could have been fixed with much less effort, had they been detected earlier.

Solution

The review process should accompany the writing process rather than being an a-posteriori activity. Reviews at different times should look at a document from different angles.

At least two reviews are necessary:

- An early review should consider the general goals as well as the overall structure and organisation of a document.
- A final review should check the contents of the document in detail, as well as the layout and spelling.

As a general rule of thumb, documents should not be given away to uninvolved readers until they have passed the final review.

Discussion

The early review should take place as soon as a document has gone through the structuring stage. The final review should take place when the spell checking and formatting has been done (see FROM BOTTOM-UP TO TOP-DOWN WRITING).

If a project keeps CONTINUING DOCUMENTATION, the team members can evaluate the documentation while they use it as the project continues. This way, the team members can provide the author with immediate feedback while the documents are still being written.

A good example of a reviewing process that accompanies the writing process is what is called shepherding in the patterns' community (see the introductions to the proceedings of the pattern conferences [6]). Each author is assigned a so-called shepherd who provides feedback while the author is completing a document.

4 Team Spirit

Problem

How can authors best profit from team colleagues?

Forces

Technical documents are often written in projects where several people collaborate. Colleagues who collaborate closely know each other well. They know each others' strengths and weaknesses. Colleagues from the same team or colleagues sharing an office room are predestined to help each other avoid typical pitfalls.

A group of people who share a common goal work much more efficiently and enjoy their work more than people working in isolation [4][5].

People are, to varying degrees, afraid of criticism. Authors might be afraid that their work is considered faulty when it is laid out before the critics' beady eyes, or they might be afraid of the extra effort of incorporating feedback into their work. Anyway, authors sometimes feel reluctant to have their material reviewed.

On the other hand, colleagues are sometimes reluctant to provide critical feedback because they are afraid to hurt the author.

Solution

When a document is written in the context of a project on which several people collaborate, the team structures should be employed to create a positive team spirit in which both authors and reviewers can feel comfortable.

- Team members should be willing to discuss material and provide feedback when their expertise is required. They must understand their role as providing a service to the author.
- Colleagues offering feedback and discussions must receive the acknowledgements and credits they deserve.
- Authors must be willing to accept feedback from reviewers, knowing that the feedback will enable them to write a much stronger document.
- The next time around, when another document is written, colleagues can mutually change the roles of author and reviewer.
- Social events or casual out-of-office activities can help colleagues get along with each other well.

Quite generally, colleagues acting as authors and colleagues acting as reviewers must understand that they are all in the same boat.

Discussion

This pattern is closely related to the idea that reviewers should mention both THE GOOD AND THE NOT SO GOOD about a document. On the one hand, acknowledging the good things contributes to a positive team spirit, while on the other hand, this team spirit is the precondition for an open discussion of what is not so good.

Neil Harrison's organizational patterns for teams stress the importance of team spirit for all team activities. A UNITY OF PURPOSE [4] — sharing a common goal — is crucial for a team to work well together .

Several patterns from Jim Coplien's pattern language for writers' workshops address the issue of team spirit with respect to document reviews. Most importantly, authors and reviewers must form a COMMUNITY OF TRUST [3] in which critical comments are useful rather than cause irritation.

5 The Good and the Not So Good

Problem

What sort of feedback should reviewers provide?

Forces

Sometimes authors are unsure about whether to keep or to replace certain material. They need to be helped in their decision; it is the purpose of a review to provide this help. Suggestions in either direction — keep or replace — can be helpful. A review that leaves the author in the dark is not helpful at all, however.

Most people can take criticism much more easily if it is clear that the criticism is intended to help them, and not to put them down [3]. Once they have received positive feedback, people can better accept negative feedback.

Reviewers often tend to say what's wrong with a document or what they dislike about it, while the good things go unnoticed.

Solution

Reviewers must come up with clear comments concerning the quality of the material reviewed.

Reviewers must force themselves to be honest. They must mention both:

- what is good about the document and should be kept,
- what is not so good, and needs improvement.

Along with all critical comments, reviewers should make concrete suggestions for improvement whenever possible.

Discussion

A positive TEAM SPIRIT is essential for this pattern that to allow open discussion of the not-so-good things about a document.

In his pattern language for writers' workshops, Jim Coplien emphasizes the importance of having POSITIVE FEEDBACK FIRST [3]. For psychological reasons, it is time for critical comments only after positive feedback has been provided. Maybe this order cannot be fully established when reviewers provide their feedback in writing, since reviewers will take notes on the paper where appropriate, but even then it is important that a section with general review comments starts with the positive aspects up front.

For criticism to be practically useful, Jim Coplien asks that concrete SUGGESTIONS FOR IMPROVEMENT [3] be offered with each critical remark.

However, a reviewer who is not familiar with the material, but who reviews a document from A DISTANT VIEW, may find that something needs improvement but may not be able to make a concrete suggestion. Though not entirely constructive, this feedback can still be helpful since it describes the impression that non-experts get from a document.

6 A Distant View

Problem

How can authors obtain unbiased feedback?

Forces

In a museum, when people look at a painting, they often step back and look from a little distance. This way, they can take in the entire painting. If they stood right in front of the painting, they would be able to see the details but the overall impression would be lost.

By analogy, reviewers who stand too close to what is described in the document under review may fail to come up with a reliable overall assessment. Reviewers too familiar with the subject of documentation are likely to take a lot of things for granted which would be questioned by an unbiased reviewer.

Technical documents are often reviewed by people with a technical background. These reviewers often focus on the technical contents of a document, but overlook presentation issues. However, the quality of presentation also deserves attention.

Solution

It is wise to choose reviewers who are interested in the document and who are generally knowledgeable in the field, but who are not involved in the actual work described in a document.

For a document written in the context of a project, good candidates for reviewers are:

- a person from outside the team who is familiar with application domain,
- a reviewer, from inside the team with a different educational background.

Discussion

This pattern is not opposed to the idea of TEAM SPIRIT, though it may seem so at first. Rather, both patterns complement each other. For reviews to have their full benefit, feedback is needed from reviewers familiar and unfamiliar with the material presented in a document.

In his generative development-process pattern language, Jim Coplien suggests hiring a MERCENARY ANALYST [2] from outside the team who is an expert in the domain and who can provide feedback on the project documentation.

In the CREATOR-REVIEWER [8] pattern from the patterns for designing in teams, Charles Weir explains that detailed knowledge of the application domain as well as uninvolvement with the work under review are important requirements for such a reviewer.

This pattern corresponds to an observation described in Neil Harrison's organizational patterns for teams. DIVERSITY OF MEMBERSHIP [4] states that mixed teams — teams with people of different age and different gender, as well as with different educational and cultural backgrounds — generally perform better than more homogenous teams, since the team members complement each other. As far as the review process is concerned, the best feedback can be obtained from reviewers with different backgrounds and with backgrounds different to the author's background.

7 Responsible Authorship

Problem

How many people should be responsible for a document?

Forces

If a large number of people are responsible for one task, it is likely that the task will not be done at all since everybody will think someone else is in charge. Responsibilities are clearer if only one person, or at most a small team, is responsible for any task.

In large projects a lot of documentation needs to be written. Maybe the project can't spare one person for long enough to write the documentation or no one person has enough knowledge to do so.

Different people use different writing styles with respect to language, diction, and lines of argument.

Different reviewers can have conflicting opinions, and authors sometimes disagree with the reviewers' opinion. Decisions have to be made about what feedback is incorporated and how it is incorporated.

Many people involved in technical project such as software engineers dislike writing documentation but prefer the more technical tasks. The author's motivation for writing is a prerequisite for a high quality document.

Solution

One person must accept responsibility for a technical document.

This person should do the following:

- collect material and arrange brainstorming sessions with other team members,
- set up the document structure and write large parts of the document,
- solicit and integrate contributions from other people if necessary, while ensuring a consistent writing style.
- arrange for document reviews and incorporate feedback from the reviews.

Preferably, this should be a person with reasonable writing skills who also enjoys writing [8].

Discussion

The author should establish a useful writing process (see FROM BOTTOM-UP TO TOP-DOWN WRITING), make sure the document is updated regularly (see CONTINUING DOCUMENTATION), and to arrange ACCOMPANYING REVIEWS.

This pattern does not contradict TEAM SPIRIT. It is fine if many team members contribute to a document; however, one person should integrate all contributions. Sometimes it can work to have several co-authors; however, one of the co-authors must feel responsible for document in its entirety.

Among his patterns of project management risk reduction, Alistair Cockburn stresses that in a project there must be exactly one OWNER PER DELIVERABLE [1], otherwise several people might work on the same thing or important tasks may be neglected. This applies to documentation as much as to all other types of deliverables.

In his pattern language for writers' workshops, Jim Coplien emphasizes that authors are not bound to the verbatim advice of the reviewers, particularly when there are contradictory suggestions for improvement. Authors retain the right and the responsibility to make SELECTIVE CHANGES [3] to a document after the review.

Conclusions

Graeme and Matthew, programmers, and Jane, the project manager, sat around in the cafeteria, drinking espresso and discussing the latest football results. Jane said: "Oh yes, one thing I forgot about at the project meeting — what are we going to do about the documentation?" "Hm, you mean we are supposed to do documentation?" "Well, sure we are." "Ok, but can't we do that when we've completed the next code release? We've got absolutely no time for that now." (Similarities to any real persons are not incidental but simply inevitable.)

When documentation gets that type of attention, it is likely to end up as a bunch of write-only papers. Nobody will read it since nobody bothered to take efforts while writing it, let alone organize the writing process. The patterns in this paper aim to avoid scenarios like this. They invite authors and reviewers involved in the documentation process to improve on their roles and activites.

Acknowledgements

My experience writing and reviewing technical documents has profited a lot from my former and present colleagues who have provided help through discussions or as co-authors or reviewers.

Participating in pattern conferences and attending writers' workshops has also provided me with a lot of insight in how to write and review technical material.

Last but not least, thanks are due to James Noble who, as the EuroPLoP shepherd for this paper, provided several suggestions for improvement including the idea for the conclusion of this paper.

References

[1] Alistair Cockburn. "Patterns of Project Management Risk Reduction", in Frank Buschmann, Dirk Riehle (Eds.), *Proceedings of the European Conference on Pattern Languages of Programming*, Siemens Technical Report 120/SW1/FB, 1997.

[2] James O. Coplien. "A generative development-process pattern language", in: James O. Coplien, Douglas C. Schmidt (Eds.), *Pattern Languages of Program Design*, Addison-Wesley, 1995.

[3] James O. Coplien. "A pattern language for writers' workshops", in Frank Buschmann, Dirk Riehle (Eds.), *Proceedings of the European Conference on Pattern Languages of Programming*, Siemens Technical Report 120/SW1/FB, 1997.

[4] Neil B. Harrison. "Organizational patterns for teams", in: John M. Vlissides, James O. Coplien, Norman L. Kerth (Eds.), *Pattern Languages of Program Design*, Vol. 2, Addison-Wesley, 1996.

[5] Tom DeMarco, Timothy Lister. *Peopleware: Productive projects and teams.* Dorset House, 1987.

[6] *Pattern Languages of Program Design*, Vol. 1-3. Addison Wesley, 1995 - 1997.

[7] Andreas Rüping. "The structure and layout of technical documents", *Proceedings of the European Conference on Pattern Languages of Program Design (EuroPLoP)*, 1998.

[8] Charles Weir. "Patterns for designing in teams: How teams can improve the design process", in: Robert C. Martin, Dirk Riehle, Frank Buschmann (Eds.), *Pattern Languages of Program Design*, Vol. 3, Addison-Wesley, 1997.