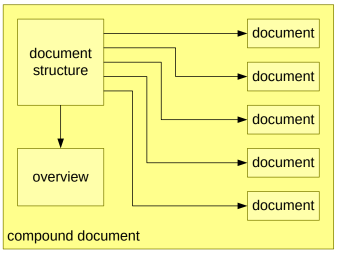
Granularity of Documentation

-



Gerrit Muller
USN-SE
Hasbergsvei 36 P.O. Box 235, NO-3603 Kongsberg Norway
gaudisite@gmail.com

This paper has been integrated in the book "Systems Architecting: A Business Perspective", `http://www.gaudisite.nl/SABP.html`*, published by CRC Press in 2011.*

Abstract

The design of documentation is discussed, with emphasis on the requirements, the need for decomposition, the measures needed to maintain overview and criteria for granularity.

version: 1.2   status: concept   March 27, 2021

1 Introduction

Documentation is an important communication means in the Product Creation Process. The whole documentation set is written by multiple authors with different competencies. System architects contribute to the structure of the documentation, and write a small subset of the documentation themselves. The size of the units within the documentation structure is called the granularity of the documentation.

The right level of granularity improves the effectiveness of the documentation. We discuss criteria to design the documentation structure, the documentation granularity, and the documentation processes.

2 Stakeholders

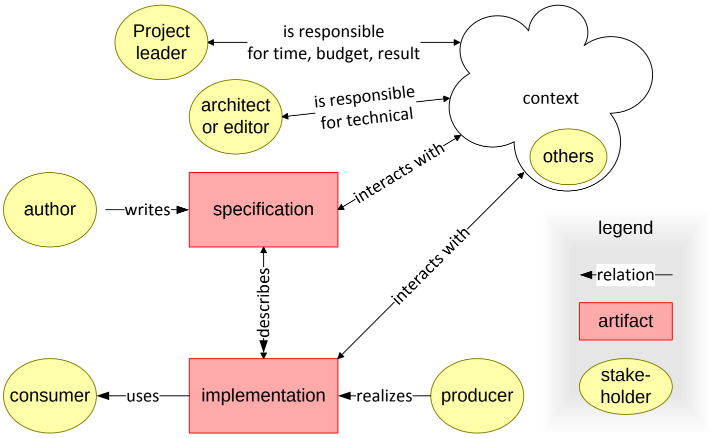


Figure 1: The stakeholders of a single document

Figure 1 shows the stakeholders of a document. The document is a description of some function or component that has to be realized by means of an implementation. The producers and the consumers of the function or component are the main stakeholders of the document. The author is also an important stakeholder. The function or component is always realized and used within a broader context. This context interacts with the function or component, so the persons responsible for the context are also stakeholder of the document. In the context there will be other stakeholders as well; people who do have some involvement with the function or component.

2.1 Example digital flat screen TV

An electronics designer writes a specification for a Printed Circuit Board (PCB) to be used in a digital flat screen TV. A digital designer and a layout engineer realize the design, hence they are the producers. A software engineer will write the software making use of the functionality of the board, he is one of the consumers. The product (the digital flat screen TV) is the context for this PCB. The designer of the power supply might be a stakeholder, especially if the PCB has specific power requirements. The industrial designer responsible for the packaging is another stakeholder. The final product will have a project leader, responsible for the schedules, costs et cetera and is stakeholder with respect to these issues. The architect at last is responsible for a balanced and consistent product design, where the PCB should fit in.

3 Requirements

The documentation of a product need to be decomposed in smaller units, with the smallest units being atomic documents. We will discuss the requirements for the entire documentation structure, the documents itself, and the underlying process.

The criteria for the entire documentation structure and process are:

Accessibility for the readers ; the information should be understandable and readable for the intended audience. The signal-to-noise ratio in the document must be high; information should not be hidden in a sea of words.

Low threshold for the readers ; No hurdles such as many pages of meta information, cumbersome security provisions, or complicated tools should dissuade readers from actually reading the document

Low threshold for the authors ; authors have to be encouraged to write. Hurdles, such as poor tools or cumbersome procedures, provide an excuse to delay writing.

Completeness of important information. Note that real completeness is an illusion, there are always more details that can be documented. All crucial aspects have to be covered by the entire documentation set.

Consistency of the information throughout the documentation. The writers strive for consistency, but we have to realize that in the complex world with many stakeholders some inconsistencies can be present. Inconsistencies that have significant impact on the result have to be removed.

Maintainability of the entire documentation, both during product creation as well as during the rest of the product life cycle.

Scalability of the documentation structure to later project phases, where many more engineers can be involved. The following measures help for scalability:

- well defined documentation structure
- explicit overview specifications at higher aggregation levels
- recursive application of structure and overview documents
- distribution of the review process

Evolvability of the documentation over time. Most documentation is re-used in successive projects.

Process to ensure the quality of the information . The quality of the content of the information is core to good results. Documentation that has been made only to satisfy the procedure is a waste of effort and time.

From reader point of view this translates in the requirements for the document infrastructure: it must be fast and easy to *view* and to *print* documents, and *searching* in the documentation also has to be fast and easy. Searching must be possible in a structured, e.g. hierarchical, way, and also via free text “a la Google”. Any part of the documentation must be reachable within a limited number of steps, so no excessively deep document hierarchies.

The criteria for the documents within the documentation structure are:

High cohesion within the document. The information in a document has to “belong” together. If information is not connected to the rest of the document, then this information might belong in another document.

Low coupling with other documents. Some coupling will be present, since the parts together will form the system. If the coupling is high, then the document decomposition is suspect and might need improvement.

Accessibility for the readers, as for the entire documentation.

Low threshold for the reader, as for the entire documentation.

Low threshold for the author, as for the entire documentation.

Manageable steps to create, review, and change the document. Documents in product creation are reviewed and updated frequently. Hence these operations should take limited effort and time. The consequence is that single documents should not be large.

Clear responsibilities, especially for the content of the document. Documents with multiple authors are suspect, responsibility for the content can be diffuse.

Worse are documents where an anonymous team or committee is "the author". If a document needs multiple authors, then it is often a symptom of bad decomposition. Also the reviewers responsibility must be clear, hence we recommend to limit the number of reviewers. When many reviewers are needed, then the decomposition is again suspect.

Clear position and relation with the context documents only make sense in the intended context. On purpose the information is captured in multiple documents. Therefor for every individual document it should be clear in what context it belongs and how it relates to other documents.

Well-defined status of the information. Documents are used and most valuable in the period when they are created. The content can be quite preliminary or draft. The document must clearly indicate what the status is of its content, so that readers can use it with proper precautions.

Timely availability of the document. When documents are too late available we do not harvest the value. Authors have to balance quality, completeness,and consistency against the required effort and time.

A very important function of documentation is communication. Communication requires that the information is accessible for all stakeholders, and that the threshold to produce documentation or to use documentation should be low[1].

4 Documentation Structure

The standard way to cope with large amounts of information is to decompose the information in smaller parts. The decomposition of the large amount of information results in a set of smaller documents. The structure of such a decomposition is made explicit in the "documentation structure", fulfilling the requirement to have a *well defined documentation structure*. The documentation structure is managed as a normal document. An overview document is required to keep the overview accessible, addressing the requirement to have *overview specifications at higher aggregation levels*. Overviews help the readers, especially when the more detailed information gets scattered in smaller documents.

This decomposition is applied recursively, see Figure 3. In this way the granularity supports the realization of the requirements as described in the 3. For instance, the principle of *recursion* is a good answer to the requirements related to *scalability*

[1] Quite often organizations focus on the documentation procedures, and documentation management, forgetting the main drivers mentioned in this subsection. The result can be tremendous thresholds, causing either apathy or bypasses. It cannot be stressed enough that procedures and tools are the **means** to solve a problem and not a goal in itself

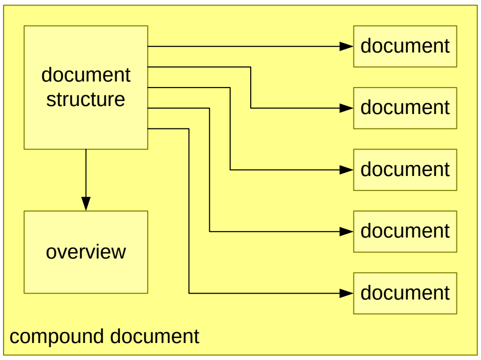


Figure 2: Large documents are decomposed in smaller documents, supported by a document structure and overview

of the entire documentation. Creating explicit structure and overview documents and allocating creation and maintenance to authors supports *maintainability*.

A fine grain structure, e.g. small documents, lower the threshold to make documents and to read the contents, in this way answering document requirements *accessibility for the reader*, *low threshold for the reader* and *low threshold for the author*.

The clarity and the value of the content is the foremost requirement for documentation. Decomposing the documentation is a balancing act in many dimensions, similar to the decomposition of systems. Clarity and value of the content may not suffer from the structure. Dogmatic structuring rules might be conflicting with clear responsibilities (*single author*). When authors write outside their expertise area, then there is a severe quality risk. The decomposition has to result in sufficiently small documents to support the requirement *Manageable steps to create, review, and change*, Large, monolithic documents violate this requirement.

The document granularity is an important design criterion for the documentation structure. The extreme that every *single value* is an entity[2] is not optimal, because the relations between values are even more important than the value itself. In case of *single value* documentation, relations are lost. The other extreme, to put everything in a single document, is conflicting with many of the requirements, such as *manageability*, *clear responsibilities*, *well-defined status* and *timely availability*. The granularity aspect, with the many psychological factors involved, is further discussed in 5.

[2] A common pitfall is to store all values in a database. In this way every value is an entity in itself. Such a database creates the suggestion of completeness and flexibility, but in reality it becomes a big heap, where the designers lose the overview. These databases may help the verification process, but do not fulfill the documentation needs.

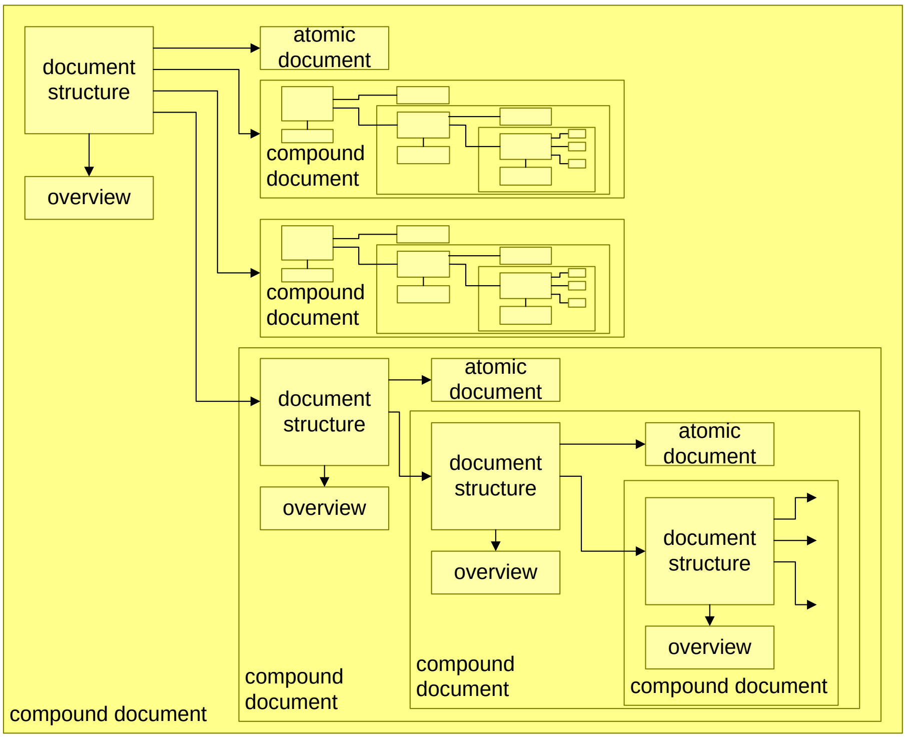


Figure 3: Decomposition is applied recursively until the atomic documents fulfill the requirements in section 3

5 Payload, the ratio between overhead and content

An atomic document must be small enough to be accessible to readers. Thick documents are put on top of the stack of "interesting papers to be read", to be removed when this stack overflows. For most people time is the most scarce resource. Struggling through all kinds of overhead is a waste of their scarce and valuable time. Documentation effectively supports communication if the reader can start directly with reading the relevant information. Figure 4 shows the layout of a good document.

The front page is used for all relevant meta-information. Meta-information is the information required for the document management, defining the status, responsibilities, context etc. The history and change information on the second page should be a service to the readers, to enable them to quickly see the relevant changes relative to earlier versions they might have read. More extensive change information, required for quality assurance purposes can be present in the document management system, it should not distract the reader from the information itself.

Such a document needs only to be opened to access the contents. Many older

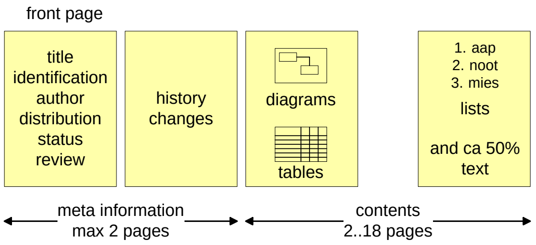


Figure 4: Layout of a good document, heuristic for the number of pages of a good document is $4 \leq nrofpages \leq 20$

organizations tend to make documents with up to 10 pages of overhead information. Many people are interrupted by phone, calendar, e-mail, or person before reaching page three. The overhead de facto inhibits people to read the contents of badly written documents[3].

The contents of a well written document ought to be optimized to get the essential information transferred. The reader community exists of different people, with differing reading and learning styles. To get information across the information must be visualized (diagrams), structured and summarized (tables and lists) and, to a limited extend, explained in text.

Once a document start its life cycle, the next risk is that the document keeps growing Authors have the tendency to transform comments and critiques of readers in explaining text. Unfortunately, large sections of text hide the key information, and violation of the maximum of 20 pages gets probable. It is better to translate the comments and critiques back into an improved diagram, table or list. Authors have to find the root cause of reader comments. For example an unclear diagram gives rise to misunderstanding.

Another frequent occurring trap is the extension of a document with missing context information. For instance, if the higher level specification is missing, parts of that specification are included in the lower level specification. An effective counter measure for this trap is to write the specification structure, showing the context and enabling to write the context later step by step. This strategy results in documents that are more focused, have a better cohesion internally, and have less coupling with other documents.

The heuristic mentioned in Figure 4 is that a good document should have 4 or more pages. This minimum should trigger people with the question if the information in a very small document has a right of existence on its own. The ratio overhead versus payload for very small documents is unbalanced. There are a small documents were the small size is appropriate.

[3] Often the situation is much worse than described here. In name of “standardization” these counterproductive layouts are made mandatory, forcing everyone to create thresholds for readers!

The maximum number of pages for a good document is 20. These documents don't scare people away yet. A 20 page document can be read in less than one hour, and the review can also be done in less than one hour. For many purposes 10 to 15 page documents are optimal. If documents require more than 20 pages the recipe is simple: make it a compound document, so split the content in multiple smaller documents.

In large documents a natural split up is often directly visible.

Large documents often violate a number of the requirements in 3. For instance, the document is edited by a single person but written by multiple authors. Another symptom of requirement violation is a document that is partly finished and partly in draft status (for instance“requirements” sections are written, while the “design” is still in full motion).

6 Acknowledgements

Angelo Hulshout triggered me to fill the the open ends in the requirements section.

References

[1] Gerrit Muller. The system architecture homepage. `http://www.gaudisite.nl/index.html`, 1999.

History

Version: 1.2, date: August 4, 2010 changed by: Gerrit Muller

- textual adaptations
- changed some figures with lists in the article into description lists
- changed status to concept

Version: 1.1, date: June 8, 2010 changed by: Gerrit Muller

- replaced lists by figures

Version: 1.0, date: May 18, 2004 changed by: Gerrit Muller

- Updated layout
- Updated figures
- Added missing text and some more detailed requirements lists
- Added acknowledgements

Version: 0.4, date: August 7, 2002 changed by: Gerrit Muller

- Abstract added

Version: 0.2, date: October 22, 1999 changed by: Gerrit Muller

- Initial Version, no changelog maintained yet.