Regarding documentation

Document version 1.0

Written by Patrik Hermansson

Contents

Abstract

Documentations is a loose concept. This document is a try to clarify about what documentation is and how to use it from a technologically point of view. Most of the information is pointed towards documentation in the information technology (IT) sector.

Some (but not all) things to answer:

- Different kind of documentation.
- Why do we need documentation?
- Collaborate or not?
- The importance of updating the documentation.

Let's dive in …

Chapter 1 – Different types of documentation

Technical documentation

Examples:

Code in repository (like GitHub or GitLab).

Detailed technical documentation of servers, firewalls or similar.

Electrical schematics.

Recipe in a cookbook.

Descriptive documentation

Descriptive documentation is usually an overview over something. A more top view on how things are working. Little or no technical information is included.

Examples:

A network schematic over a network or setup with several servers.

A flow chart describing on how a program or script is working.

Policy documentation

Policy documentations differ from the other kind of documents described here because the matter of the document is more abstract than in the others. A policy documentation is more of an ide on how to proceed, whereas the other documents show more detailed view on how the world are in a model (Descriptive) or down on a detailed level (technical).

Mix-mode documentation

This is documentation, usually for project planning, that includes both parts of the technical and parts of the descriptive documentations. This kind of documentation is good for planning new setups and changes in current setup, programs or similar.

Chapter 2 - Why do we need documentation?

This might feel like a silly question and maybe the question should be “Why is documentation so important?”.

For my personal opinion, the documentation is very important. It is the foundation to the decision for the program, the setup and / or the function implemented. And it is the foundation for good maintenance of the function or program.

Without good documentation there is a big risk that the program is full of bugs that are not fixed or that the system(s) are not patched or updated. New functions are not implemented and maybe even so bad that the system or program is forgotten.

Another big part is in the case of employees quitting or the implementation is done by consultants or project hired technicians. If there are subpar or no documentation and the person(s) that worked on or with the program/setup is no longer with the company. Then there will be lots of time wasted to enumerate knowledge that could be quickly acquired via documentation. Almost always more time than the time it took do the documentation in the first place. And the documentation needs to be done anyway.

And as many says… Time is money.

The documentation should also include some kind of plan for the future and -/or lifecycle management.

Chapter 3 - For whom is the documentation?

Is it just for the little team or single person or for a bigger group? Mabey the whole company?

On what complexity/technical level should the documentation be written?

How sensitive is the information in the documentation, if it was to 'leak' outside of the company?

Will there be one author or is it a collaboration between several parties involved in the documentation?

For collaboration

What to consider in short:

- Internal or external (cloud) tool.
- Ease of use
- IAM, who will get access to what?

Tools like DokuWiki, SharePoint, Confluence enterprise can be used to make it easier to collaborate in the documentation.

Is the tool ok to be in the 'cloud'? Or is the data so sensitive that it should be a locally placed tool?

Like for all tools the easy to use and level of technical skills of the contributors needs to be assessed. Is the wiki path with all entries done with wiki-markup code? Or is a WYSIWYG (What You See Is What You Get) editor needed?

Ease of use should be one part of the choice of tool.

If the content of the documents is of a sensitive sort. Then a granular restriction function should be implemented. Identity and Access Management (IAM) should always be part of the process of choosing the collaboration tool.

Choosing a tool

A thing to keep in mind is to start with how!

How do you want the documentation and then try to find a tool to work with the how. Not the other way around, I.e., implement a tool and then try to make it work.

The tool should be a help to the documentation not the thing you need to rewrite your policy for to make it work. Also, the tool must be something that everyone works with. There cannot be a college that "goes their own way" and uses other tools for documentation.

Documentation should be easily accessible and easily read for the duration the documentation is valid. This can be a problem if the documentation is written in a prosperity format (or specific program) that is no longer accessible or readable.

Update or not

So should you update the documentation?

The answer is almost always yes. The documentation is a “living” document. That means that constantly updating is needed. And so is the case with the most documentation. There are changes in what the document describes that needs to be updated in the documentation.

Chapter 4 – Contributors

There are different kind of contributors. I have chosen to split them in voluntary- and non voluntary contributors. Description of the two below. There will always be those who want to do it like it always was done. To not change anything if possible. And sure not all change is good but a big part of change is just that.

Voluntary contributors

This is the people that will just do the documentation and "fall in line". Will use the presented tools and do the work. The easy ones.

Non voluntary contributors

This is the people that, buy some way, are against documentation or the tool(s) of documentation. Usually, these people's aviation to the documentation or tool(s) for documentation is based form lack of knowledge. So, the logical way to make them more prone to document is to inform and up the knowledge regarding the importance of documentation.

If there still are someone who do not want to do documentation in the new tool or documentation at all. Then someone with power in the company needs to put down the foot and make them get back in the que.

But this can be a problem if some or several people do not contribute and can be a problem. This needs to be fixed as soon as possible. And via the managers of the organization. But not with only total force if it can be avoided. The best is to inform and educate to resolve the problems there can be. And maybe this will resolve all issues.

Make sure to find those who are sceptic or against the tool/documentation as soon as possible to help them to find the benefits of the situation.

Education

Regardless of what kind of contributors you have, everyone will most probably need some kind of onboarding and education. It is much easier if there is a plan on onboarding and education from the start. Mabey even some how-tos and other documents to help the users start working with the new tool.