https://www.halvorsen.blog

Week Assignment

Software Documentation

Hans-Petter Halvorsen

Week Assignment

1. Create **System Documentation**
2. Create **User Manual**(s)s
   - We shall Make **Video**(s)
3. Make sure to Update Existing Documents (SDP, SRD, STP, STD)

+ Continue with Implementation, Testing and Bug Fixing!! – **Iteration #3**

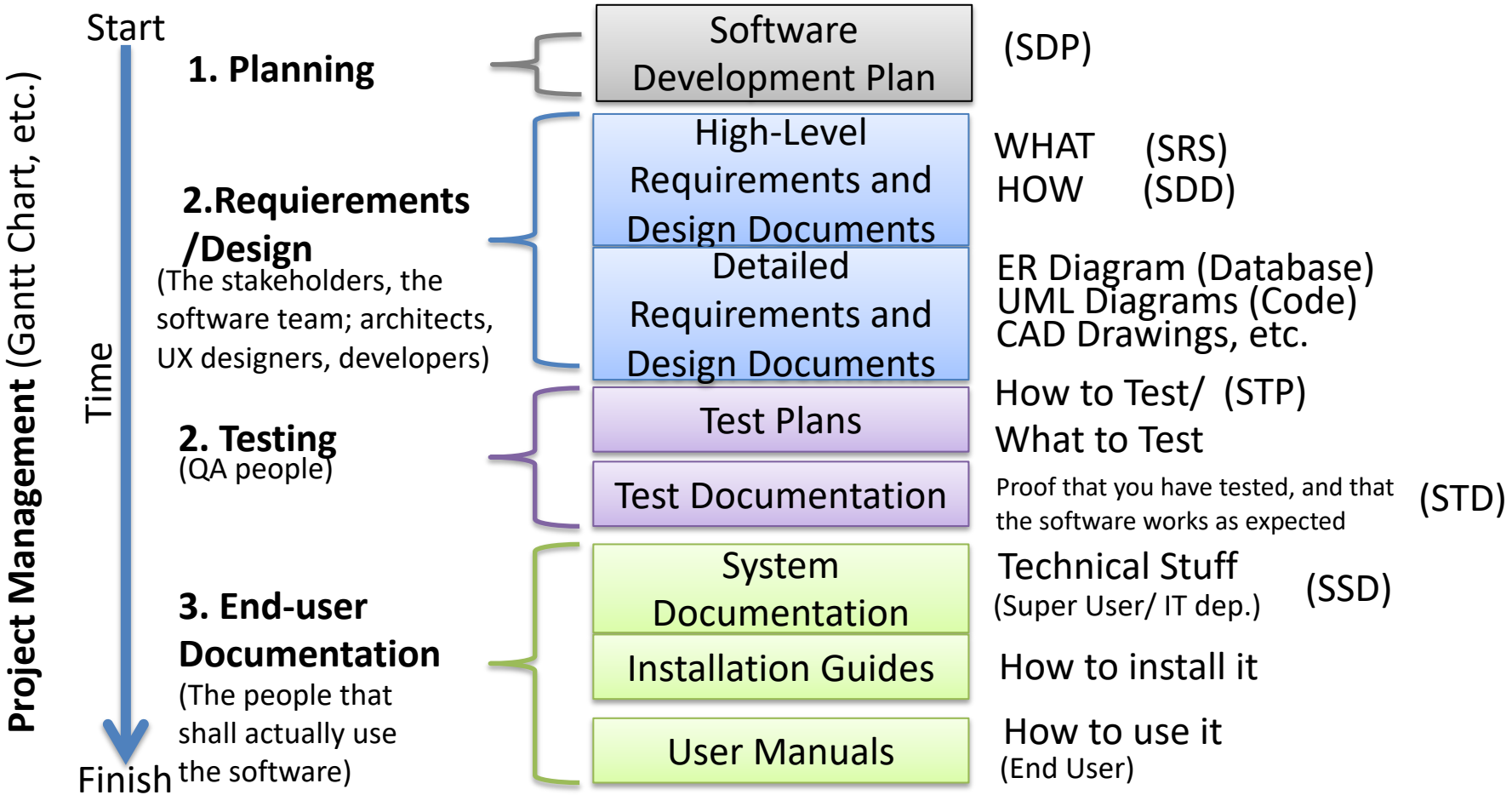
Next Week: Installation (Deployment) and Installation Guide(s), etc.

Software Documentation

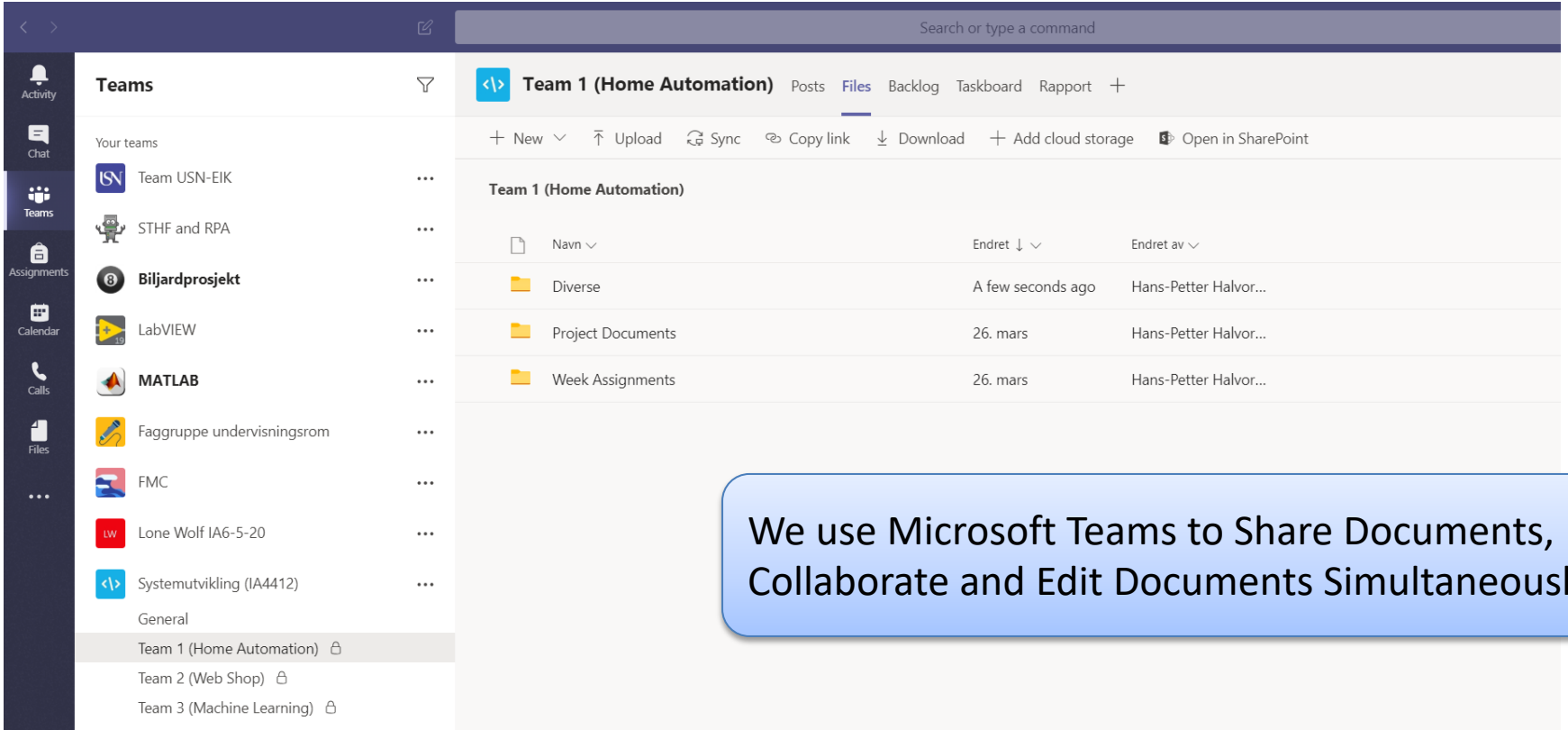
Hans-Petter Halvorsen

Table of Contents

Typical Software Documentation

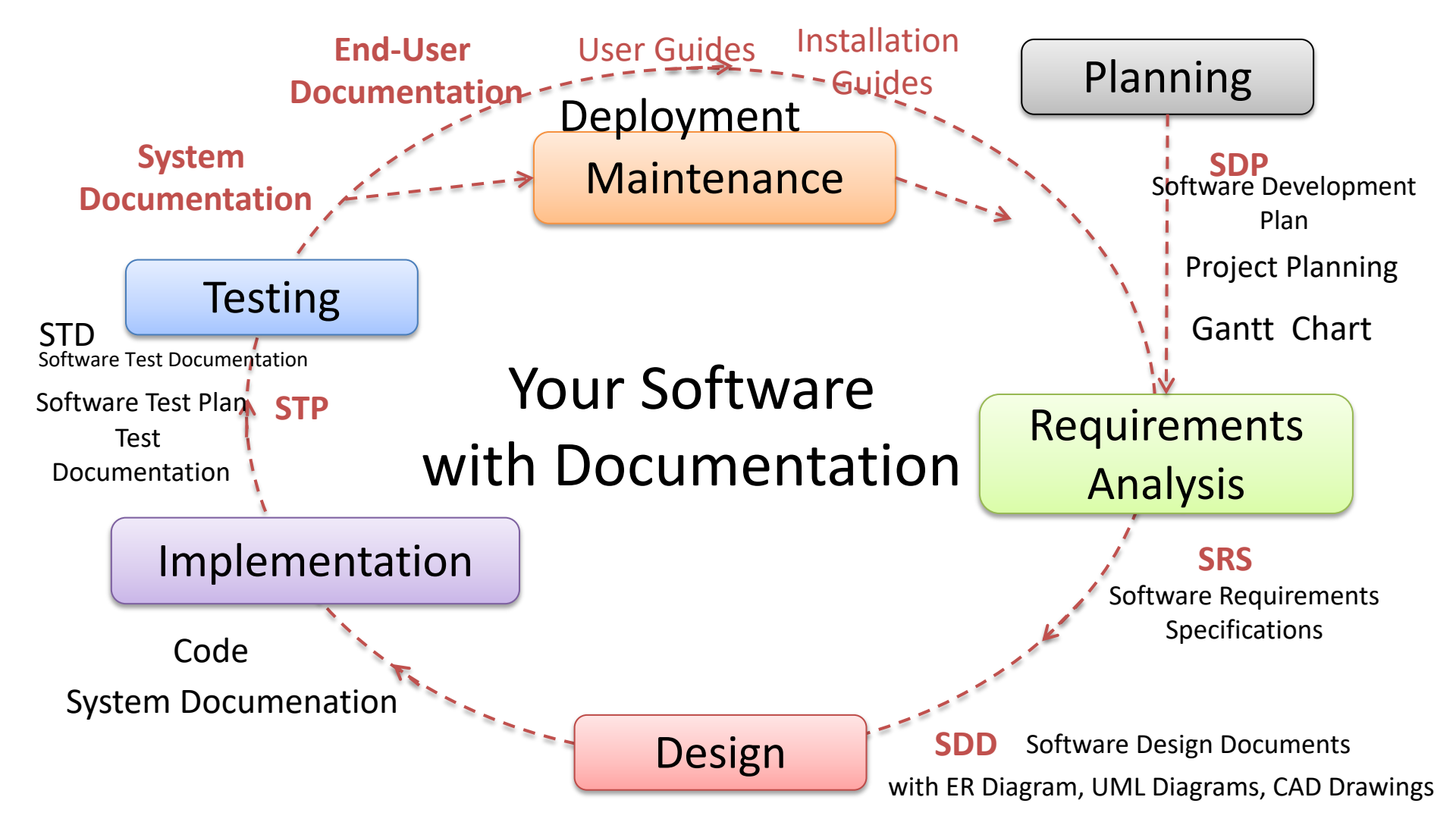


Sharing Documents with Teams



Document Location?

- We will use **Azure DevOps** to store and share Project Planning and Source Code
- While Working documents should be stored in **Microsoft Teams** to make it easy for the Team to work on the same documents simultaneously in real time.
- PDF documents should also be uploaded to your HTML Web Site (Your “Final Report”)
- We should also share Release/Final Documents (Word files, Excel files, Visio files, etc.) in Azure DevOps (In case you need to update a specific document for a specific release)

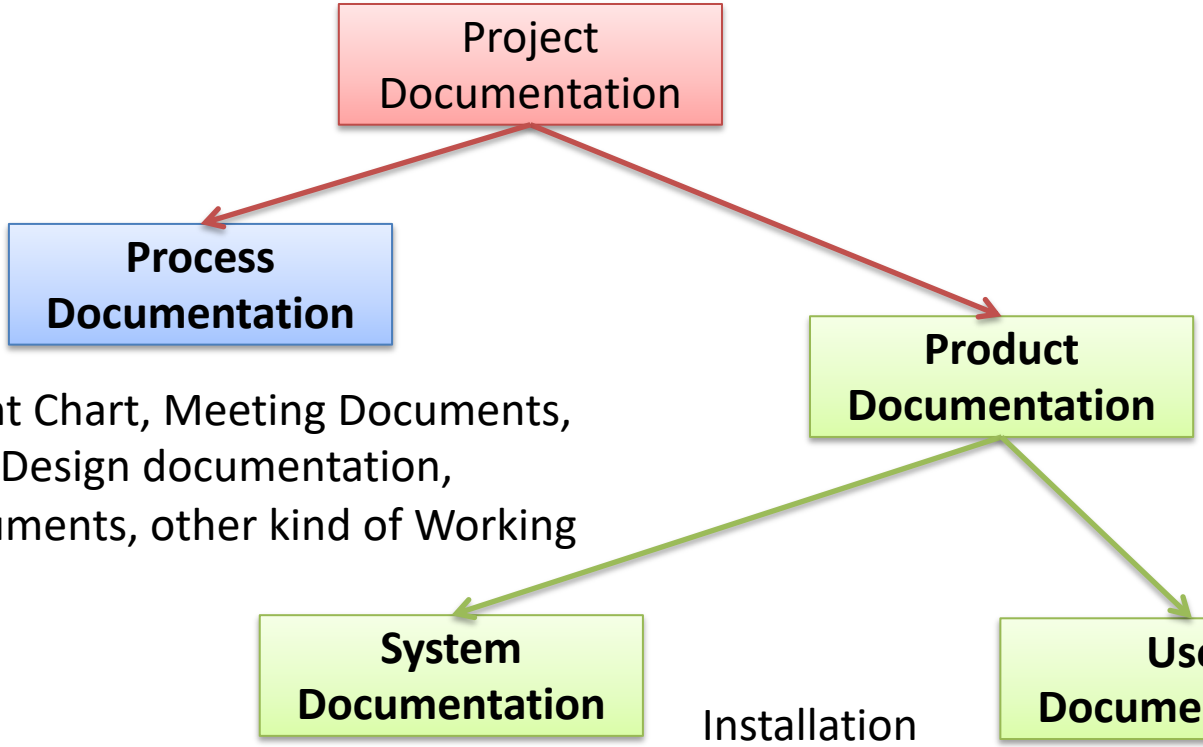

End-User Documentation
User Guides
Installation Guides
Planning
Deployment
Maintenance
System Documentation
SDP
Software Development Plan
Project Planning
Gantt Chart
Testing
STD
Software Test Documentation
Software Test Plan
Test Documentation
STP
Your Software with Documentation
Requirements Analysis
SRS
Software Requirements Specifications
Implementation
Code
System Documenation
Design
SDD
Software Design Documents
with ER Diagram, UML Diagrams, CAD Drawings

Software Project Documentation

Documentation produced during a software Project can be divided into **2 Categories**:

- **Process Documentation**
  - These documents record the process of development and maintenance, e.g., Plans (Software Development Plan, Software Test Plan, ...), Schedules (e.g., Gantt Charts), etc.
- **Product Documentation**
  - These documents describe the product that is being developed. Can be divided into 2 sub categories:
    - **System Documentation**
      - Used by engineers developing and maintaining the system
    - **User Documentation**
      - Used by the people that is using the system

Software Documentation Categories



Project Plan, Gant Chart, Meeting Documents, Requirements & Design documentation, Emails, Test documents, other kind of Working Documents, etc.

Technical Documentation needed in order to maintain the software, etc.

Installation Guides

User Manual, Wikis, Online Help, etc.

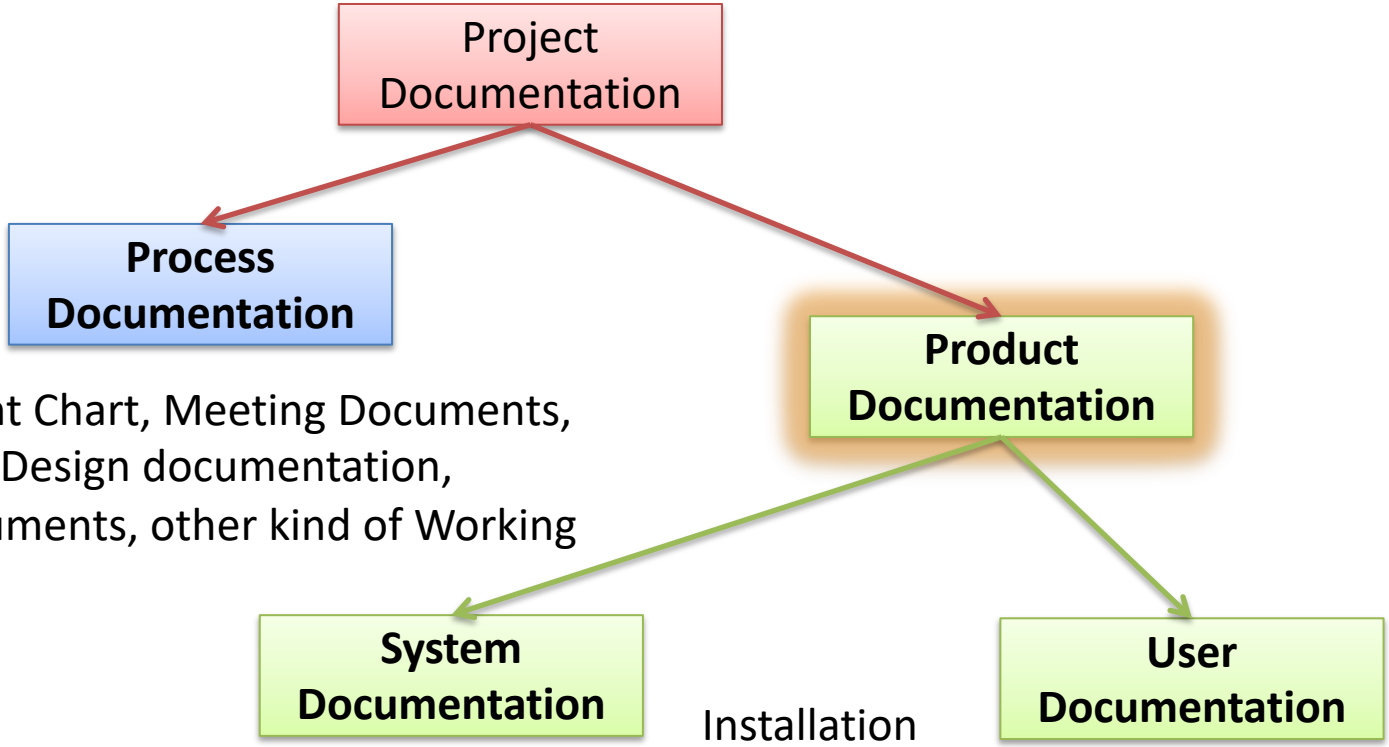
Software **Process** Documentation

1. Software Development Plan (**SDP**)
2. Software Requirements Specifications (**SRS**)
3. Software Design Documents (**SDD**)
4. Software Test Plan (**STP**)/Software Test Documents (**STD**)

Those we are “Finished” with

– Next Step is to create the **Product** Documentation

Software Documentation Categories



Project Plan, Gant Chart, Meeting Documents, Requirements & Design documentation, Emails, Test documents, other kind of Working Documents, etc.

Technical Documentation needed in order to maintain the software, etc.

Installation Guides

User Manual, Wikis, Online Help, etc.

Software Documentation Requirements

- Should act as a communication medium between members of the Development Team (Process Documentation)
- Information respository used by Maintenance Engineers (Product Documentation)
- Information for Management to help them Plan, Budget and Schedule the Software Development Process (Process Documentation)
- Some of the documents should tell users how to use and administer the system (Product Documentation)
- Documents for Quality Control, System Certification, etc. (Process/Product Documentation)

=> Satisfying these requirements requires different types of documents from informal working documents through professionally produced User Manuals

User Documentation **Readers**

- Users of a system are not all the same.
- The producer of documentation must structure it to cater for different user tasks and different levels of expertise and experience.
- It is particularly important to distinguish between end-users and system administrators:

1. **End-users** use the software to assist with some task.
   - This may be flying an aircraft, managing insurance policies, writing a book, etc. They want to know how the software can help them. They are not interested in computer or administration details.
2. **System Administrators** are responsible for managing the software used by end-users.
   - This may involve acting as an operator if the system is a large mainframe system, as a network manager is the system involves a network of workstations or as a technical guru who fixes end-users software problems and who liaises between users and the software supplier.

Document Checklist

Document Checklist

When writing documents, they should have a minimum standard when it comes to formatting, layout, numbering, the way you write, etc. Make sure that your documents fulfill all the items in the list below before you deliver it.

#	Item	OK
1	I have included a separate **Title page** with a Title (that makes sense for the reader of the document) and your Name. Typically, a Date is also appropriate to include.	
2	My **Headings/Chapters** are using **numbering**, e.g., "1 Heading1", "1.1 Heading2", etc. In that way it is so much easier to find a specific chapter and to see the structure of the document	
3	I also use the built-in Styles "Heading1", "Heading2" and "Heading3" included in MS Word. I can of course select "Modify" to adjust them to my needs (unless you are using a Template that should not be changed)	
4	I have made a **System Sketch** typically in the Introduction (or the Problem Description) chapter. A system sketch gives the user an overview of the system and the relationship between different parts of the system using basic squares, rounds, arrows, etc. I can use any tool I prefer for this, but with PowerPoint you can make such a sketch in a minute or two and then directly copy it into my document.	
5	I always start each **Chapter** and **Subchapter** with a short introduction text before I present any Figures, Tables, a list of bullet points, etc.	
6	I have NOT used any **Figures**, **Tables** or directly copied **Equations** from the resources given by the supervisor since I don't learn anything doing this. I have made my own Figure, Sketches, Tables, etc. where I show how I understands it and, in that way, presenting my work (not others).	
7	The Figures I have inside my document is of **high quality** and I can see all the neces	
8	I ha wr C# ha	
9	**Fi** be do ju au	
10	Fo se of	
11	Al in is	
	regarding the source. I have also obtained necessary permission from the owner in those cases where this is needed.	
12	**I prefer to make and use my own Figures and Sketches because it is important to tailor made Figures and Sketches, so they are in the context of my work.** If I have based my Figures and Sketches on others, I have of course referred to the original Figure in the text and explaining that it is a modified version.	
13	For each **Table** I have added a Table number and Table title **ABOVE** the Table, e.g., "Table 3-4: PID Parameters for selected Tuning methods". I also always do this **immediately** after I have inserted the Table (not later) since it takes just a few seconds to do this. It also looks better when the Table is centered.	
14	For each **Table** I have referred to that Table in the text, e.g., "In Table 3-4 we see the PID parameters for the different tuning methods used in this project, these tuning methods…."	
15	I am using the **Built-in features inside MS Word** when making Figure/Table Captions (right-click on the Figure and select "Insert Caption…") and when referring to those in the text (in the "References" tab select "Cross-reference"). In that way my numbering will always be correct even if I add more Figures or Tables in between later.	
16	The **Equations** are centered and have an Equation number that is right centered, e.g., $y = ax + b$ (2-1)	
26	I am not using any "strong" **colors** except for e.g., alarm handling or other situations that require "strong" colors	
27	I have used **proper names and labeling** for my VIs (NOT like "Form1", "Peters PID Controller"), variables (NOT "Numeric Control" but e.g., "Temperature"), user interface objects (NOT "Waveform Chart" but e.g., "Temperature Chart"), etc.	
28	The results of my work are **discussed**, e.g., "The Skogestad tuning gives better control performance than the Ziegler-Nichols method when used in the ulator…" and/or something like this: "The results from the simulations in Table 3-4 shows that the control system works fine when applying a sponse. The performance is also good when applying noise to the	
	een the "**Big picture**", meaning I have not focused on unnecessary or included very basic stuff, nor am I talking about "Task 1", "Task 2",	
	ces have been included since I use information from other sources e assignment or information provided by the supervisor. In addition to erence list itself, I have inside the report where the source is used to the reference using a number, e.g., [2], e.g., "From [2] we know re are a linear relationship between the voltage and the temperature degrees Celsius."	
	T adding any "manual" space (by hitting Enter button more than side my report. Instead I have used "Modify Style" then selected ", then "Paragraph" and finally setting "Spacing" ("Before" and	
	ead the entire document and I have found no obvious mistakes, **g mistakes**, etc.	
	I have generated the final **PDF** file, I have opened it and read through the entire text and have not been able to find obvious mistakes, spelling mistakes, etc. I have also checked that there are none "Reference not found…", etc.	

It is good idea to make (or use) a "Document Checklist" to make sure that you don't deliver documents with basic and obvious mistakes. Use it as a basic Quality Control before you send the document to others (Its almost like Unit Testing but for the different items inside a document instead of your source code). Make sure to update the SDP so the Team Members know where to find the Document Checklist

System Documentation

Hans-Petter Halvorsen

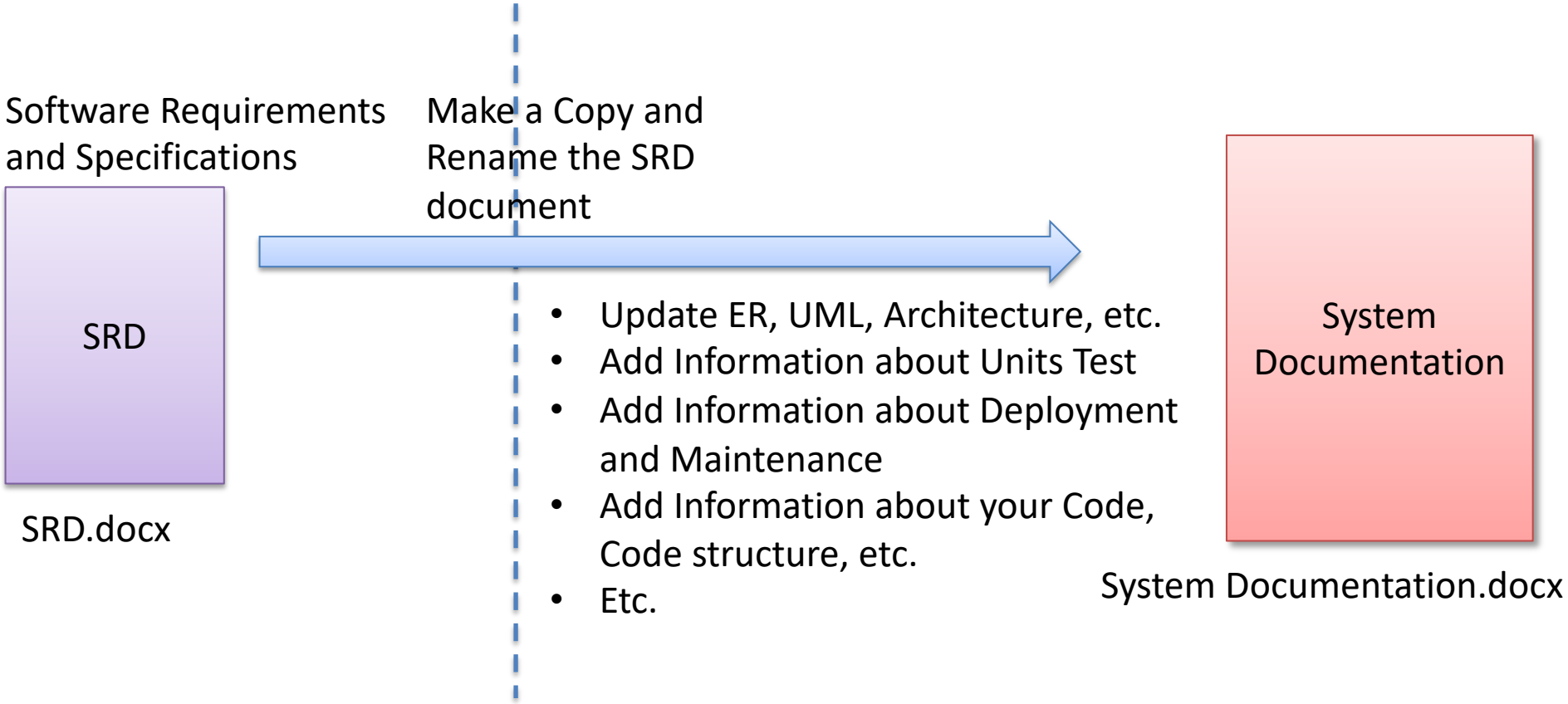
Table of Contents

System Documentation

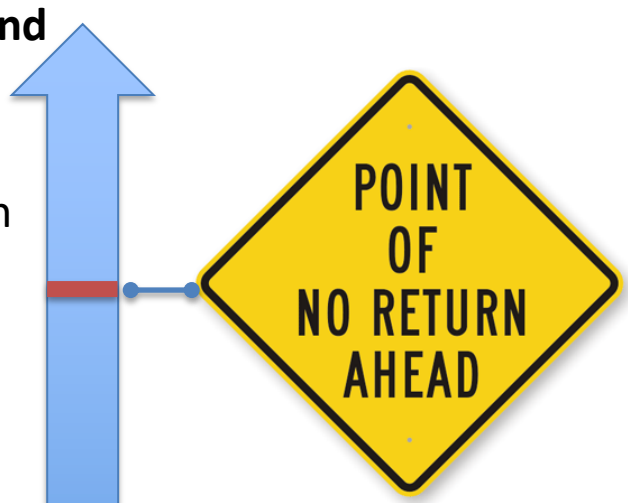
- Create System Documentation for your Systems
- It can be one or more documents
- **Tip**: Make a copy of your SRS/SDD (->SRD) and take it from there (Rename, Add, Delete, Change contents, etc.).

See Next Slides for more details...

System Documentation



Point of no Return



From this point you should not update the “SRD” document. You only update the “System Documentation”, i.e., the focus is writing and finishing the “Product Documentation” and not the “Process Documentation”

The point of no return is the point beyond which one must continue on one's current course of action because turning back is physically impossible, prohibitively expensive, or dangerous

Feature/Code Freeze
Make no new Features - only Finish what you have started

Process Documentation

Product Documentation

SDP SRS SDD STP STD

SRD

System Documentation

User Manual(s)

Installation Guide(s)

SDP - Software Development Plan
SRS – Software Requirements and Specification
SDD – Software Design Document
SRD – Software Requirements and Design
STP – Software Test Plan
STD – Software Test Documentation

System Documentation

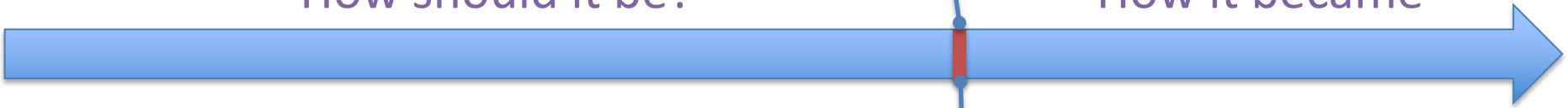
Feature/Code Freeze
Make no new Features - only Finish what you have started

Process Documentation

Product Documentation

How should it be?

How it became



Software Requirements and Design (SRD)

System Documentation

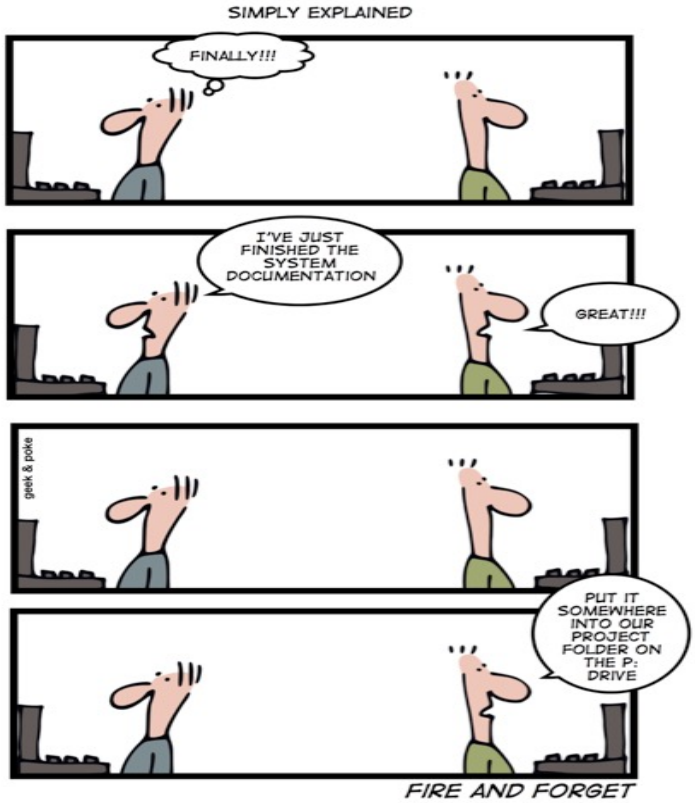
Stop updating SRD

Make a copy of SRD and rename the Document.

Add Code Documentation, Unit Testing, etc.

System Documentation

- How the System Works (Technical), i.e. use the Requirements & Design as base.
- Requirements & Design Documents is about how it should (planned to) be, while System Documentation is about how it became
- Includes Technical Design and Platform Overview, Database Diagram, UML diagrams, CAD drawings, Code Documentation, Flow Charts, with explanations, etc.
- How to deploy (how to install server-side logic), maintain, etc.
- Code Documentation, Unit Testing Documentation



System Documentation

- System documentation includes all of the documents describing the system itself from the requirements specification to the final acceptance test plan.
- Documents describing the design, implementation and testing of a system are essential if the program is to be understood and maintained.
- Like user documentation, it is important that system documentation is structured, with overviews leading the reader into more formal and detailed descriptions of each aspect of the system.

Implementation and Code

- Documentation of the Code is an important part of the System Documentation
- Unit Testing should also be part of the System Documentation

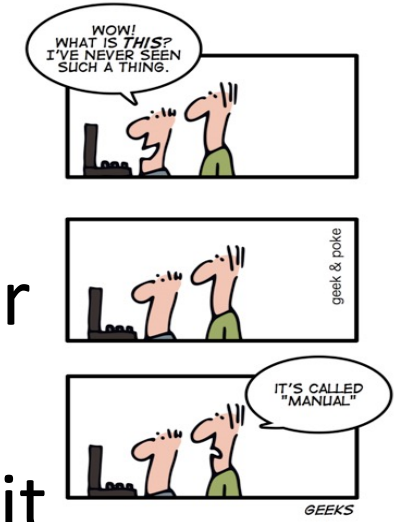
User Manual/Guide

Hans-Petter Halvorsen

Table of Contents

User Manual/Guide

- Create one or more User Manuals for your System
- You typically create one User Manual for each Module
- It can be an ordinary Word/PDF File, or it can be online help (Web, HTML), Video(s), etc.
  - We shall create Video(s)



See Next Slides for more details...

User Manual

Your Software

Video

User Guide.mp4

You

To Do:

- Action: Make one or more Videos where you go through your Software and give an overview and explain how to use your Software.
- Targeted Audience: The End-user of the Software
- Output: MP4 Video File(s)
- Location: Embed your Videos in your HTML Web Site and possibly a link within your software

User Manual/Guide

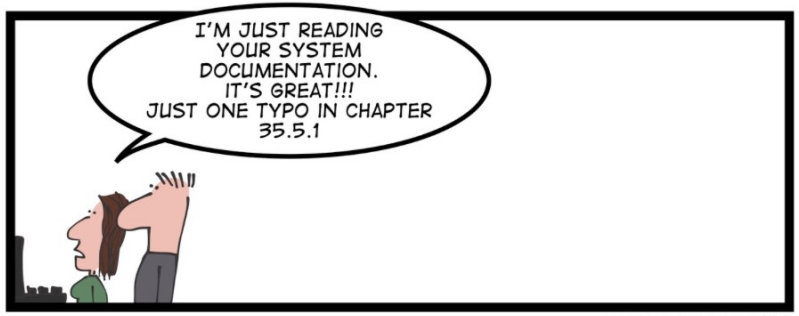
A user guide or user's guide, also commonly known as a manual, is a technical communication document intended to give assistance to people using a particular system. It is usually written by a technical writer, although user guides are written by programmers, product or project managers, or other technical staff, particularly in smaller companies.

The sections of a user manual often include:

- A cover page
- A title page and copyright page
- A preface, containing details of related documents and information on how to navigate the user guide
- A contents page
- **A guide on how to use at least the main functions of the system (Text + Screen Shots)**
- A troubleshooting section detailing possible errors or problems that may occur, along with how to fix them
- A FAQ (Frequently Asked Questions)
- Where to find further help, and contact details
- A glossary and, for larger documents, an index

Our Focus!!

http://en.wikipedia.org/wiki/User_guide



BE AWARE!!!

SOMEBODY MAY ACTUALLY READ IT!

Update Existing Documents

Hans-Petter Halvorsen

Table of Contents

Update Existing Documents

Documents created so far:

- Software Development Plan (SDP)
- Software Requirements and Design document (SRD)
- Software Test Plan (STP)
- Software Test Documentation (STD)

→ Those are living documents that needs to be updated continuously

→ Go through all previous Week Assignments to make sure you have included all necessary information

→ In addition to make sure these documents have the correct contents, they should also have proper formatting (like Figure Numbering, Unified Layout, Intuitive Structure, etc.)

See Next Slides for more details...

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog