

Software Requirements and Design (SRD)

Vi må ha en standard tittelside (Side 1) på alle dokumenter.
I tillegg til tittel, kan vi ha med firmanavn, logo, m.m.
Innholdsfortegnelse bør også være med på side 2.

1 Generelt om dokumenter

Nedenfor ser vi typiske dokumenter i et software-utviklingsprosjekt (Figure 1-1).

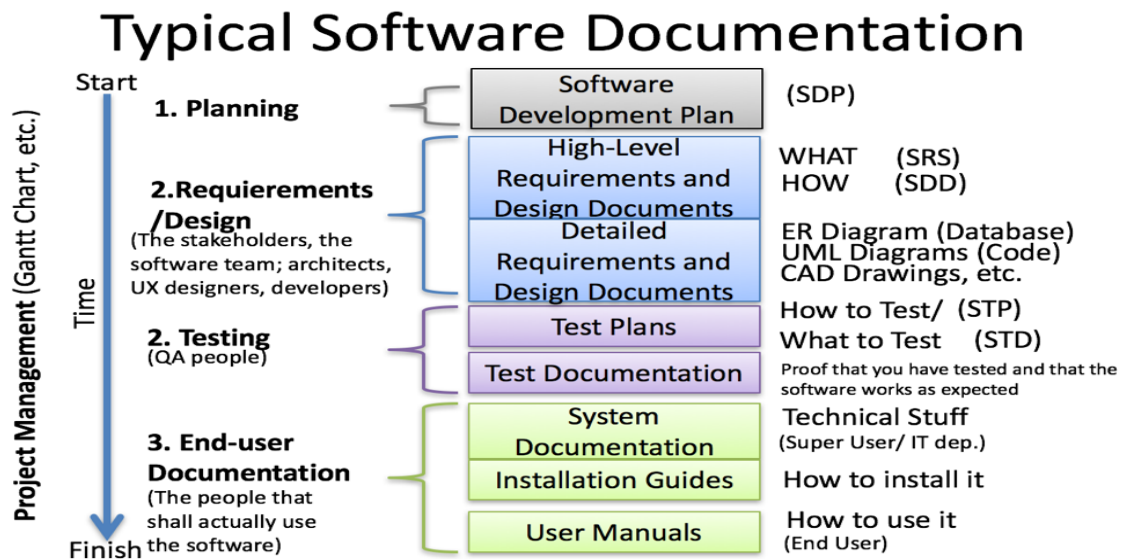


Figure 1-1: Typisk Softwaredokumentasjon

Alle disse dokumentene skal vi lage utover i faget.

1.1 OneDrive

Det kan være lurt å bruke f.eks. Microsoft OneDrive for deling og samhandling ifm utarbeidelse av dokumentene. Da har man muligheten for at flere personer samtidig kan endre på samme dokument på hver sin PC. OneDrive er også innebygd i Windows 10, så det kreves ikke noe ekstra for å ta det i bruk. Man kan lagre opptil 5Gb med data gratis, som jo burde være nok i vårt tilfelle.

For best utbytte er det viktig at alle bruker Word 2016, evt. Word Online.

1.2 Dokumentkategorier

Dokumenter kan deles inn i ulike kategorier (Figure 1-2).

Software Documentation Categories

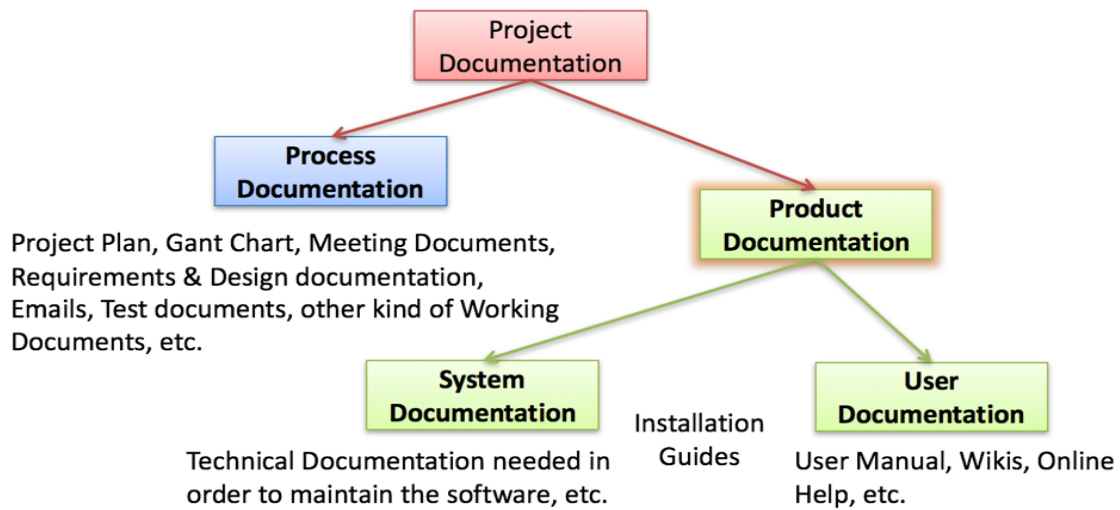


Figure 1-2: Dokumentkategorier

Figure 1-3 viser forskjellen mellom et tradisjonelt skoleprosjekt og et virkelig software-prosjekt.

“Traditional School Project” vs. “Real Software Development Project”

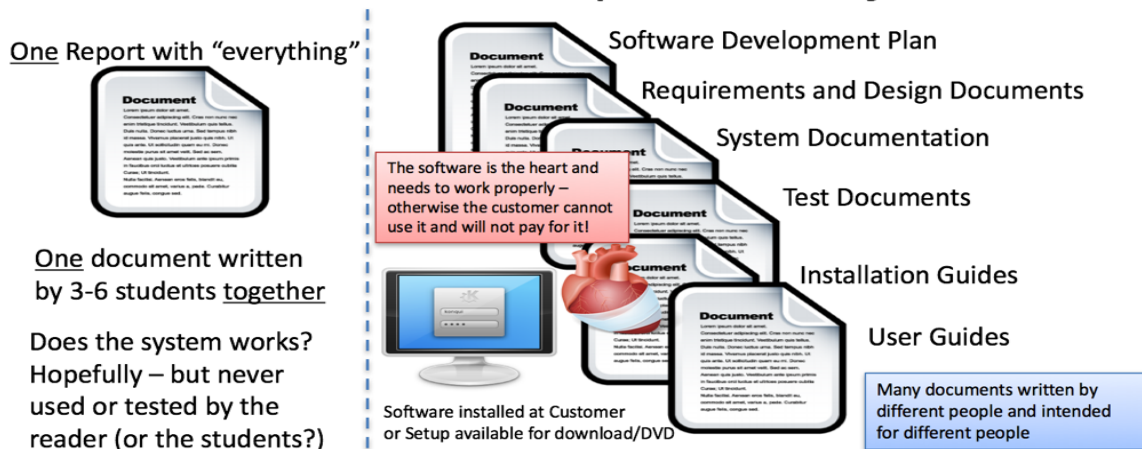


Figure 1-3: Skoleprosjekt vs. virkelig prosjekt

Dvs. vi skal ikke lage et dokument i form av en rapport i dette faget (som jo er vanlig i noen prosjektfag i skolesammenheng), men et sett med dokumenter som til sammen vil inngå som en del av totalleveransen. Dette for å gjøre det mest mulig virkelighetsnært i forhold til hvordan et virkelig software-prosjekt blir dokumentert.

Det er ingen fasit mtp utarbeidelse av dokumentasjon, det er mange varianter ute å år. Det er en del grunnleggende ting som går igjen i alle former for dokumenter.

Alt dere har lært om rapportskrivning tidligere gjelder også her! – Den eneste forskjellen at her er det flere dokumenter istedenfor en rapport. Produktwebsiden blir på en måte selve rapporten, mens SDP, SRD, STP, osv. blir på en måte vedlegg.

Dvs. grunnleggende “rapporttekniske” ting som Tabell og figurnummerering, m.m. må selvfølgelig være med!

Typisk struktur: Forside/Tittelside, innholdsfortegnelse, Innledning, en eller flere kapitler med innhold, + evt vedlegg.

Bruke nummerering i forbindelse med kapitler og underkapitler er vanlig.

Sidenummerering må selvfølgelig være med.

Bruke samme "Word Template" på alle dokumentene! – dvs bruk samme fonter, fargebruk, m.m. i de ulike dokumentene.

Bruk av referanser der det er bruk kilder, ressurser fra andre.

2 Hva er SRD?

SRD – Software Requirements and Design

Requirements (WHAT):

- WHAT the system should do
- Describes what the system should do with Words and Figures, etc.
- SRS – Software Requirements Specification Document

Software Design (HOW):

- HOW it should do it
- Examples: GUI Design, UML, ER diagram, CAD, etc.
- SDD – Software Design Document

Note! Many don't separate SRS and SDD documents, but include everything in a Requirements & Design Document (SRD). In practice, requirements and design are inseparable.

SRS – Software Requirements Specifications

SDD – Software Design Document

SRD – Software Requirements and Design

Figure 2-1 viser typisk innhold i et SRD document.

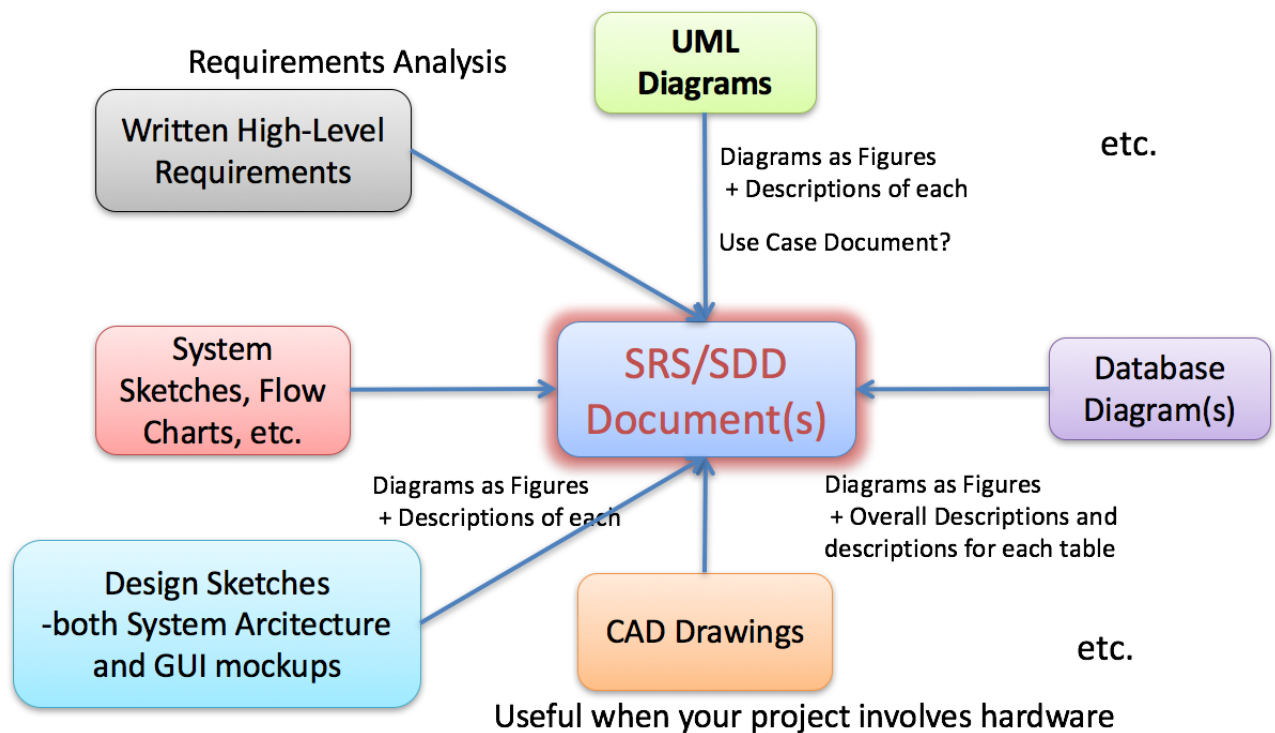


Figure 2-1: SRD Contents

2.1 Hensikten med dokumentet

Et løpende dokument som brukes til å gi en oversikt over hva som skal lages (Requirements)/utvikles og hvordan det skal lages/utvikles (Design).

Dokumentet brukes som en "kontrakt" mellom utviklingsfirma og kunde, samt brukes kontinuerlig av utviklere (når de skal lage/utvikle løsningen), testere (når de skal teste løsningen) og kunde (slike at løsningen blir slik de ønsker) underveis.

Mye av innholdet lages i oppstarten av prosjektet (eller i forkant av prosjektet), men innholdet må oppdateres kontinuerlig når endringer oppstår, særlig med med smidig utvikling vil dette være tilfellet.

3 Eksempler

3.1 Eksempel A

Typisk innhold i SRS dokumentet:

Følgende temaer bør være med [1]:

- System Overview (brief description of what the software system will do)
- Technical Requirements (Functional requirement, Non-functional requirements, User-interface specification, User task flow, Input/output and other data specifications, Interface specifications to other systems)

- Acceptance Criteria/Interaction Scenarios
- Validation/Verification
- Requirements Considerations (Assumption made about the software, End users, Existing systems, Environment, Limitations)
- Other Information...

Mer detaljer om disse punktene du i boka - Appendix B. Denne boka er tilgjengelig fra Safari Books Online.

3.2 Eksempel B

Typisk innhold i SRS dokumentet:

Følgende temaer bør være med [2]:

Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.
System requirements specification	This should describe the functional and nonfunctional requirements in more detail. If necessary, further detail may also be added to the nonfunctional requirements. Interfaces to other systems may be defined.
System models	This might include graphical system models showing the relationships between the system components and the system and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.
Appendices	These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organization of the data used by the system and the relationships between data.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.

Figure 3-1: SRS eksempel

Merk! Dere trenger nødvendigvis ikke følge alternativ A eller B slavisk, dere kan gjerne blande, tilføye eller utelate ting som ikke er relevant for deres prosjekt.

Det er viktig at man har med software-arkitektur og oversikt over software plattformer, m.m. Dvs. skal systemet som utvikles kun virke for Windows? Eller skal det virke på en Mac også. Hvilke nettlesere skal vi støtte?

Gode skisser (Systemoversikt) i forbindelse med dette med ulik detaljeringsgrad for ulike lesere er relevant.

3.3 Eksempel C

Software Requirements and Design (SRD) document:

- System Overview
 - Introduction, Description of the system, Problem Description, Sketches of the system
- Technical Requirements
 - Functional requirement, Non-functional requirements, User-interface specification, User task flow, Input/output and other data specifications, Interface specifications to other systems
- Architecture
 - The technical architecture of the system, system sketches, etc.
- Database
 - Database modelling and detailed descriptions
- UML
 - Use Case Diagrams, Sequence Diagrams, Class Diagrams

etc.

SRD is a mix of the SRS and SDD documents.

Many don't separate SRS and SDD documents, but include everything in a Requirements & Design Document (**SRD** document).

In practice, requirements and design are inseparable.

4 Referanser

- [1] F. Tsui, O. Karam, and B. Bernal, *Essentials of Software Engineering*, 3 ed.: Jones & Barlett Learning, 2014.
- [2] I. Sommerville, *Software Engineering*, 10 ed.: Pearson, 2015.
- [3] E. J. Braude and M. E. Bernstein, *Software Engineering: Modern Approaches*, 2 ed.: Wiley, 2011.