Software Documentation

HEI! HVA ER DETTE?
HAR VI BYTTEL
DOKUMENTSYSTEM?

STEMMER!

HVFOR NÅ?! NÅR
ALT FUNKER OG VI ENDELIG
KLÆRER OSS SELV?!
DU HAR JO
KURSET OSS ALLE FRAM TIL
EN VØNSKET DOKTORGRAD I
DOKUMENTNUMMERERING!

DITT OPPDRAG
ER FERDIG!

VENT, LITT...

NYTT
GRUNNKURS
STARTER PÅ
MÅNEDAG!


Hans-Petter Halvorsen, M.Sc.
Your Software with Documentation

Planning
- SDP: Software Development Plan
- Project Planning
- Gantt Chart

Requirements Analysis
- SRS: Software Requirements Specifications

Design
- SDD: Software Design Documents with ER Diagram, UML Diagrams, CAD Drawings

Implementation
- Code
- System Documentation

Testing
- Test Plan
- Test Documentation

Deployment
- User Guides
- Installation Guides

Maintenance
- End-User Documentation

System Documentation
- STD: Software Test Documentation

Diagram:
- Your Software with Documentation
- Planning
- Requirements Analysis
- Design
- Implementation
- Testing
- Deployment
- Maintenance

Connections:
- Planning to Requirements Analysis
- Requirements Analysis to Design
- Design to Implementation
- Implementation to Testing
- Testing to Deployment
- Deployment to Maintenance
- Maintenance to Planning

Documentation:
- SRS
- SDD
- STD
- User Guides
- Installation Guides
- End-User Documentation
- Project Planning
- Gantt Chart
- Test Plan
- Test Documentation
- Code
- System Documentation
- ER Diagram
- UML Diagrams
- CAD Drawings
**Project Documentation**

**Project Planning**
- High-Level Requirements and Design Documents
- Detailed Requirements and Design Documents

**Software Development Plan (SDP)**
- HOW
  - ER Diagram
  - UML Diagrams
  - CAD Drawings

**Test Plans**
- How to Test/
- What to Test
- Proof that you have tested and that the software works as expected

**Test Documentation**

**System Documentation**
- Technical Stuff
  (Super User/ IT dep.)
- How to install it

**Installation Guides**
- How to use it
  (End User)

**User Manuals**

**Final Report**

---

**Time**

**1. Planning**
(stakeholders, the software team; architects, UX designers, developers)

**2. Testing**
(QA people)

**3. End-user Documentation**
(The people that shall actually use the software)

**4. Handover**
(Business people)
Software Project Documentation Categories

Project Documentation

Process Documentation
- Project Plan, Gant Chart, Meeting Documents,
- Requirements & Design documentation,
- Emails, other kind of Workin Documents, etc.

Product Documentation

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

User Documentation
- 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 Documents (STD)
Software Requirements & Design

Requirements (WHAT):

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

Software Design (HOW):

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

Many don't separate SRS and SDD documents, but include everything in a Requirements document.

In practice, requirements and design are inseparable.
The Software Requirements Document

Software Requirements Specifications (SRS)

• The software requirements document is the official statement of what is required of the system developers.

• Should include both a definition of user requirements and a specification of the system requirements.

• It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it.

# The Structure of the SRS Document

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>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.</td>
</tr>
<tr>
<td>Introduction</td>
<td>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.</td>
</tr>
</tbody>
</table>
Software Documentation Requirements

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

=> Satisfying these requirements requires different types of documents from informal working documents through professionally produced User Manuals
If the documentation is not useful – dont make it!!
Software Documentation

• System/Technical Documentation
  – Class Diagrams
  – State Diagrams
  – Sequence Diagrams
  – Code Comments

• User Documentation
  – User Manual
  – Installation Guide
  – Wiki
  – Online Documentation and Help
Software Documentation

• For large projects, it is usually the case that documentation starts being generated well before the development process begins.

• The set of documents that you have to produce for any system depends
  – on the contract with the client for the system (the customer)
  – the type of system being developed
  – its expected lifetime
  – the company culture
  – the size of the company developing the system
  – the development schedule
Programmers don't document 😞

**SIMPLY EXPLAINED**

LET ME TAKE A WILD GUESS:
int doSomething()
DOES SOMETHING

WHAT ELSE?

SELF DOCUMENTING CODE

AND THIS IS JIM, OUR NEW DEVELOPER

GREAT! DOES HE ALREADY KNOW ANYTHING ABOUT OUR SYSTEM?

I READ THE WHOLE DOCUMENTATION

NO

But they should 😊!!!
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, 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
Process Documentation
Process Documentation

Purpose:

• Process Documentation is produced so that the development of the system can be managed

• It is an essential component of plan-driven approaches (e.g., Waterfall)

• Agile Approaches: The Goal is to minimize the amount of Process Documentation
Categories:

1. **Plans, estimates and schedules.** These are documents produced by managers which are used to predict and to control the software process.

2. **Reports.** These are documents which report how resources were used during the process of development.

3. **Standards.** These are documents which set out how the process is to be implemented. These may be developed from organizational, national or international standards.

4. **Working papers.** These are often the principal technical communication documents in a project. They record the ideas and thoughts of the engineers working on the project, are interim versions of product documentation, describe implementation strategies and set out problems which have been identified. They often, implicitly, record the rationale for design decisions.

5. **E-mail messages, wikis, etc.** These record the details of everyday communications between managers and development engineers.
Software Development Plan (SDP)

An SDP normally include the following sections:

1. **Introduction**: This briefly describes the objectives of the project and set out the constraints (e.g., budget, time, etc.) that affects the management of the project.

2. **Project Organization** (Team Description) This section describes how the development team is organized, the people involved and their roles in the team. Software Process Model Description (Scrum, XP, Waterfall, ...), etc.

3. **Risk Analysis**

4. **Hardware and Software Resource Requirements**

5. **Work Breakdown** (WBS, Work Breakdown Structure): Break down the project into activities and identifies milestones.

6. **Project Schedule**: Shows dependencies between activities, the estimated time required to reach each milestone, allocation of people to activities. (5) and (6) is typically done in a Gantt Chart (created in e.g. Microsoft Project).

7. **Monitoring and Reporting Mechanisms**: Definition of the Management Report that should be produced, when theses should be produced, etc.

8. Tools that you are using
Gantt Chart

Microsoft Project - gantt chart example.mpp

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine a budget</td>
<td>1 day</td>
<td>Tue 10/9/07</td>
<td>Tue 10/9/07</td>
<td></td>
</tr>
<tr>
<td>Research Technologies</td>
<td>5 days</td>
<td>Wed 10/10/07</td>
<td>Tue 10/9/07</td>
<td>1</td>
</tr>
<tr>
<td>Select Technology</td>
<td>1 day</td>
<td>Wed 10/17/07</td>
<td>Wed 10/17/07</td>
<td>2</td>
</tr>
<tr>
<td>Research Brands</td>
<td>3 days</td>
<td>Thu 10/18/07</td>
<td>Mon 10/22/07</td>
<td>3</td>
</tr>
<tr>
<td>Select Brand and Model</td>
<td>1 day</td>
<td>Tue 10/23/07</td>
<td>Tue 10/23/07</td>
<td>4</td>
</tr>
<tr>
<td>Check On-line Stores</td>
<td>2 days</td>
<td>Wed 10/24/07</td>
<td>Thu 10/25/07</td>
<td>5</td>
</tr>
<tr>
<td>Check Local Outlets</td>
<td>5 days</td>
<td>Wed 10/24/07</td>
<td>Tue 10/30/07</td>
<td>6</td>
</tr>
<tr>
<td>Select Retailer</td>
<td>1 day</td>
<td>Wed 10/31/07</td>
<td>Wed 10/31/07</td>
<td>7,8</td>
</tr>
<tr>
<td>Purchase</td>
<td>1 day</td>
<td>Thu 11/1/07</td>
<td>Thu 11/1/07</td>
<td>8</td>
</tr>
</tbody>
</table>
Test Documentation

Software Test Plan (STP)

Planning Tests → Perform Tests → Document Test Results

Software Design Document (SDD)
Software Requirements Specifications (SRS)

Test Logs

Software Test Documentation (STD)

These documents will be the foundation for all Testing

- Functional & Non-Functional Requirements
- User & System Requirements
Product Documentation
Product Documentation

Purpose:

• Describing the delivered software product
• Unlike most process documentation, it has a relatively long life.
• It must Evolve in step with the product that it describes.
• Product documentation includes
  – User documentation, which tells users how to use the software product,
  – System Documentation, which is principally intended for maintenance engineers.
To cater for these different classes of user and different levels of user expertise, several documents (or perhaps chapters in a single document) should be delivered with the software system.
Product Documentation

User Documentation
User Documentation

THE GEEK’S HANDBOOK

CHAPTER 2: HOW TO BE A GOOD COLLEAGUE

I’m looking for someone to do the user documentation. Who do you think is better in documenting: you or... STEVE!!

Be humble.
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.
User Manual

Wow! What is this? I've never seen such a thing.

It's called "manual".

Geek & poke

Geeks
THE HISTORY OF...
GETTING CODERS TO DOCUMENT
...AND YOU CAN NOT ONLY EDIT IN VI, IT ALSO LOOKS LIKE A PERL SCRIPT!!!

Product Documentation

System Documentation
System Documentation

Simply Explained

FINALLY!!!

I'VE JUST FINISHED THE SYSTEM DOCUMENTATION

GREAT!!!

WELL, YOU NEED SOME SKILLS, SOME PATIENCE, SOME PASSION AND...

PUT IT SOMEWHERE INTO OUR PROJECT FOLDER ON THE P: DRIVE

CORRECT!

A LOT OF CTRL-C CTRL-VP
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.
Code Documentation

SIMPLY EXPLAINED

LET ME TAKE A WILD GUESS:
int doSomething()
DOES SOMETHING

WHAT ELSE?

DECADE, CENTURY, MILLENNIUM. AND WHAT COMES NEXT?

//TODO

SELF DOCUMENTING CODE

System Documentation

For large systems that are developed to a customer’s specification, the system documentation should include:

1. The **requirements** document.
2. A document describing the **system architecture**.
3. For each program in the system, a description of the architecture of that program.
4. For each component in the system, a description of its functionality and interfaces.
5. Program **source code** listings, which should be commented where the comments should explain complex sections of code and provide a rationale for the coding method used.
   - If meaningful names are used and a good, structured programming style is used, much of the code should be self documenting without the need for additional comments.
   - This information is now normally maintained electronically rather than on paper with selected information printed on demand from readers.

6. **Validation** documents describing how each program is validated and how the validation information relates to the requirements.
   - These may be required for the quality assurance processes in the organization.

7. **A System Maintenance Guide**, which describes known problems with the system, describes which parts of the system are hardware and software dependent and which describes how evolution of the system has been taken into account in its design.
**CODER - NORMAL**

CODER: "IT'S JUST A QUICK HACK. WE'LL HAVE TO CLEAR UP SOON"

TRANSLATION: "NEVER"

THANK GOD!

LOOK AT THIS CODE!!! NOT ONE COMMENT.

I'M NOT THE ONLY ONE

CODER: "I PUT A // TODO: COMMENT INTO THE CODE"

TRANSLATION: "NEVER EVER"
UML

Use Case Diagrams

Class Diagrams

Sequence Diagrams
Database ER Diagram

Example:

- **SCHOOL**
  - **PK**: SchoolId
  - **SchoolName**: Description, Address, Phone, PostCode, PostAddress

- **CLASS**
  - **PK**: ClassId
  - **SchoolId**: ClassName Description

- **COURSE**
  - **PK**: CourseId
  - **CourseName**: Description

- **GRADE**
  - **PK**: Gradeld
  - **FK1**: StudentId
  - **FK2**: CourseId
  - **Grade**: Comment

- **STUDENT_COURSE**
  - **PK, FK1**: StudentId
  - **FK2**: CourseId

- **STUDENT**
  - **PK**: StudentId
  - **FK1**: ClassId
  - **StudentName**: StudentNumber
  - **TotalGrade**: Address, Phone, Email

- **TEACHER**
  - **PK**: TeacherId
  - **SchoolId**: TeacherName Description

- **TEACHER_COURSE**
  - **PK, FK1**: TeacherId
  - **FK2**: CourseId
I just finished coding this feature. Before I start with the next, I will do the documentation.

There is something wrong here.
VELDIG BRA RAPPORT, KJELL! BARE ET PAR SMÅTTING Å RETTE PÅ!

NEI, NEI! SE HER. ET HELT AVSNITT SOM HAR FÅTT SMILEFEJES!

TULLER DU? ALT HER ER JO RØDT!

...DET ENESTE JEG IKKE HAR SKREVET SELV. REN AVSKRIFT.

...MEN TALLENE, DA?

FRI FANTASI! KONKLUSJONEN OGSÅ.

DU LEVERER PÅ ET UTROLIG STABILT NIVÅ, KJELL. LAVT, MEN STABILT.

TAKK!
References

Hans-Petter Halvorsen, M.Sc.

University College of Southeast Norway

www.usn.no

E-mail: hans.p.halvorsen@hit.no
Blog: http://home.hit.no/~hansha/