Lab Session 01

Explore the usage of any documentation tool in Software Development Life Cycle (SDLC)

Software documentation

All large software development projects, irrespective of application, generate a large amount of associated documentation. A high proportion of software process costs is incurred in producing this documentation. Furthermore, documentation errors and omissions can lead to errors by endusers and consequent system failures with their associated costs and disruption. Therefore, managers and software engineers should pay as much attention to documentation and its associated costs as to the development of the software itself. The documents associated with a software project and the system being developed, have a number of associated requirements:

1. They should act as a communication medium between members of the development team.
2. They should be a system information repository to be used by maintenance engineers.
3. They should provide information for management to help them plan, budget and schedule the software development process.
4. Some of the documents should tell users how to use and administer the system.

Types of Software Documentation

Generally, the software project documentation produced falls into two classes:

1. ***Process documentation:*** These documents record the process of development and maintenance. *Plans, schedules, process quality documents and organizational and project standards are process documentation*. The major characteristic of process documentation is that most of it becomes outdated. Plans may be drawn up on a weekly, fortnightly or monthly basis. Progress will normally be reported weekly. Although of interest to software historians, much of this process information is of little real use after it has gone out of date and there is not normally a need to preserve it after the system has been delivered.

2. ***Product documentation:*** This documentation describes the product that is being developed. *System documentation describes the product from the point of view of the engineers developing and maintaining the system; user documentation provides a product description that is oriented towards system users.* Unlike most process documentation, it has a relatively long life. It must evolve in step with the product, which it describes. Product documentation includes user documentation, which tells users how to use the software product and system documentation, which is principally intended for maintenance engineers. See Figure 2.1 for different types of User Documents.

System documentation consists of the following:

- Software Requirement Specification (SRS)

- Design and Architecture
- Source Code Documents
- Testing Documents
- Formal Technical Reviews

User documentation includes the following:

- End-users Documentation
- System-admin Documentation

Document structure

The document structure is the way in which the material in the document is organized into chapters and, within these chapters, into sections and subsections. Document structure has a major impact on readability and usability. As with software systems, you should design document structures so that the different parts are as independent as possible. The IEEE standard for user documentation proposes that the structure of a document should include the components shown in Figure 2.2.

Component	**Description**
Identification data	Data such as a title and identifier that uniquely identifies the document.
Table of contents	Chapter/section names and page numbers.
List of illustrations	Figure numbers and titles
Introduction	Defines the purpose of the document and a brief summary of the contents
Information for use of the documentation	Suggestions for different readers on how to use the documentation effectively.
Concept of operations	An explanation of the conceptual background to the use of the software.

Procedures	Directions on how to use the software to complete the tasks that it is designed to support.
Information on software commands	A description of each of the commands supported by the software.
Error messages and problem resolution	A description of the errors that can be reported and how to recover from these errors.
Glossary	Definitions of specialized terms used.
Related information sources	References or links to other documents that provide additional information
Navigational features	Features that allow readers to find their current location and move around the document.
Index	A list of key terms and the pages where these terms are referenced.
Search capability	In electronic documentation, a way of finding specific terms in the document.

Fig 1.1: Suggested components in a software user document

Document Preparation

Document preparation is the process of creating a document and formatting it for publication. Figure 2.3 shows the document preparation process as being split into 3 stages namely document creation, polishing and production. The three phases of preparation and associated support facilities are explained in the figure below:

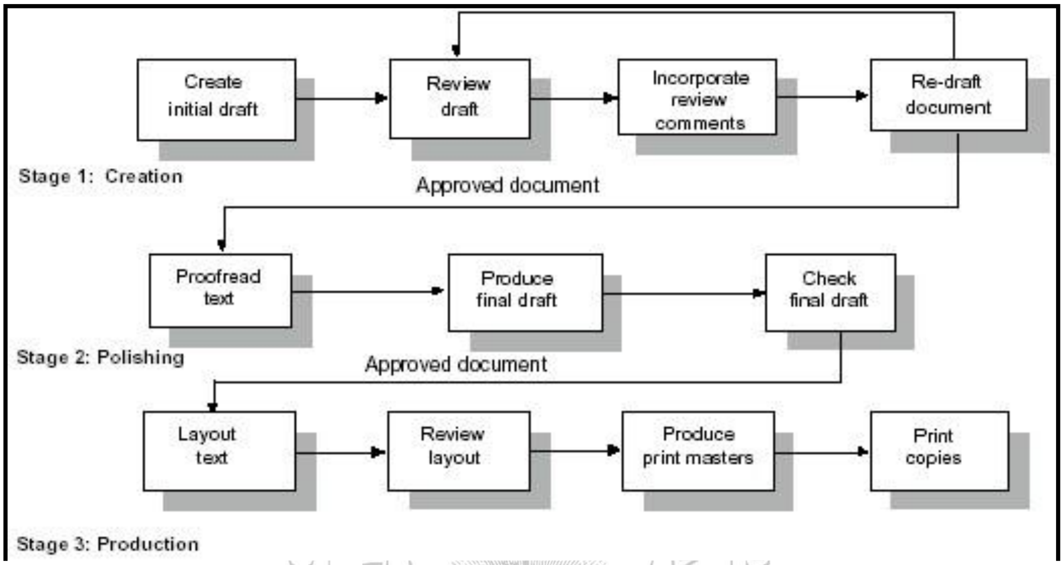

Fig 1.2: Stages of document preparation

LaTeX

LaTeX is a high -quality document preparation system; it includes features designed for the production of technical and scientific documentation. It allows users to very quickly tackle the more complicated parts of typesetting, such as inputting mathematics, creating tables of contents, referencing and creating bibliographies, and having a consistent layout across all sections. It is based on the WYSIWYM (what you see is what you mean) idea, meaning you only have focus on the contents of your document and the computer will take care of the formatting.

LaTeX is available as free software. To set up a basic TeX/LaTeX system, download and run the Basic MiKTeX Installer. TeXworks is used as editor for writing commands. We start off with a basic .tex file which contains LaTeX code. LaTeX uses control statements, which define how your content should be formatted. LaTeX compiler takes .tex file as input and convert it into a .pdf file.

First Piece of LaTeX

```
\documentclass{article}

\begin{document}
First document. This is a simple example, with no  extra
parameters or packages included.
\end{document}
```

The first line of code declares the type of document, known as the *class*. The class controls the overall appearance of the document. In this case, the class is article, the simplest and most common LATEX class. Other types of documents you may be working on may require different classes such as **book** or **report**. Then the contents of our document are written, enclosed inside the **\begin{document}** and **\end{document}** tags. This is known as the *body* of the document. You can start writing here and make changes to the text if you wish. To see the result of these changes in the PDF you have to compile the document.

The Preamble of a document

In the previous example the text was entered after the **\begin{document}** command. Everything in your **.tex** file *before* this point is called the **preamble**. In the preamble you define the type of document you are writing, the language you are writing in, the *packages* you would like to use (more on this later) and several other elements. For instance, a normal document preamble would look like this:

```
\documentclass[12pt, letterpaper]{article}
```

Below a detailed description of \documentclass:

\documentclass[12pt, letterpaper]{article}

As said before, this defines the type of document. Some additional parameters included in the square brackets brackets can be passed to the command. These parameters must be comma-separated. In the example, the extra parameters set the font size (**12pt**) and the paper size (**letterpaper**). Of course other font sizes (**9pt**, **11pt**, **12pt**) can be used, but if none is specified, the default size is **10pt**. As for the paper size other possible values are **a4paper** and **legalpaper**.

Adding a title, author and date

To add a title, author and date to our document, you must add three lines to the **preamble** (NOT the main body of the document). These lines are:

\title{First document}

\author{Hubert Farnsworth}

\thanks{funded by the Overleaf team}

This can be added after the name of the author, inside the braces of the **title** command. It will add a superscript and a footnote with the text inside the braces.

\date{September 2019}

Enter the date manually or use the command **\today** so the date will be updated automatically.

```
\documentclass[12pt, letterpaper, twoside] {article}

\title{First document}
\author{Hubert Farnsworth \thanks{funded by the Overleaf team}}
\date{September 2019}
```

Now a title, author and date are provided to the document, we can print this information on the document with the **\maketitle** command. This should be included in the **body** of the document at the place you want the title to be printed.

```
\begin{document}
```

```
\maketitle
We have now added a title, author and date to our first \LaTeX{}
document! \end{document}
```

Adding Comments

To make a comment in LATEX, simply write a **%** symbol at the beginning of the line as shown below:

```
\begin{document}

\maketitle
\pagenumbering {gobble}
We have now added a title, author and date to our first  \LaTeX{}
document!
 % This line here is a comment. It will not be printed in the
document.
 \end{document}
```

Basic Formatting

- **Abstracts**

In scientific documents it's a common practice to include a brief overview of the main subject of the paper. In LATEX there's the **abstract** environment for this. The **abstract** environment will put the text in a special format at the top of your document.

```
\begin{document}
\newpage
\begin{abstract}
This is a simple paragraph at the beginning of the
document. A brief introduction about the main subject.
\end{abstract}
\end{document}
```

- **Paragraphs and Newlines**

When writing the contents of your document, if you need to start a new paragraph you must hit the "Enter" key twice (to insert a double blank line). Notice that LATEX automatically indents paragraphs. To start a new line without actually starting a new paragraph insert a break line point, this can be done by \\ (a double backslash as in the example) or the \newline command.

```
\begin{document}
\begin{abstract}
This is a simple paragraph at the beginning of the document. A
brief introduction about the main subject.
\end{abstract}
Now that we have written our abstract, we can begin writing our
first paragraph.
```

```
This line will start a second Paragraph.
\end{document}
```

- **Chapters and Sections**

Commands to organize a document vary depending on the document type, the simplest form of organization is the sectioning, available in all formats.

```
\chapter{First Chapter}

\section{Introduction}

This is the first section.

Lorem  ipsum  dolor  sit  amet,  consectetuer  adipiscing
elit.  Etiam  lobortisfacilisis sem.  Nullam nec mi et  neque
pharetra sollicitudin.  Praesent imperdietmi nec ante.  Donec
ullamcorper, felis non sodales...

\section{Second Section}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.   Etiam
lobortis facilisissem.  Nullam nec mi et neque pharetra
sollicitudin.  Praesent imperdiet mi necante...

\subsection{First Subsection}
Praesent imperdietmi nec ante. Donec ullamcorper, felis non
sodales...

\section*{Unnumbered Section}
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Etiam lobortis facilisissem
```

The command **\section{}** marks the beginning of a new section, inside the braces is set the title. Section numbering is automatic and can be disabled by including a ***** in the section command as **\section*{}**. We can also have **\subsection{}**s, and indeed **\subsubsection{}**s. The basic levels of depth are listed below:

-1	**\part{part}**
0	**\chapter{chapter}**
1	**\section{section}**
2	**\subsection{subsection}**
3	**\subsubsection{subsubsection}**
4	**\paragraph{paragraph}**

5	**\subparagraph{subparagraph}**

Note that \part and \chapter are only available in report and book document classes.

• Bold, Italics and Underlining

Following are some simple text formatting commands.

- **Bold**: Bold text in LaTeX is written with the **\textbf{...}** command.
- *Italics*: Italicised text in LaTeX is written with the **\textit{...}** command.
- Underline: Underlined text in LaTeX is written with the **\underline{...}** command.

```
Some of the \textbf{greatest}
discoveries in \underline{science}  were
made by \textbf{\textit{accident}}.
```

• Unordered lists

Unordered lists are produced by the **itemize** environment. Each entry must be preceded by the control sequence **\item** as shown below. By default the individual entries are indicated with a black dot, so-called bullet. The text in the entries may be of any length.

```
\begin{itemize}
  \item The individual entries are indicated with a black dot, a
so-called bullet.
  \item The text in the entries may be of any length.
\end{itemize}
```

• Ordered lists

Ordered list have the same syntax inside a different environment. We make ordered lists using the enumerate environment:

```
\begin{enumerate}
  \item This is the first entry in our list
  \item The list numbers increase with each entry we add
\end{enumerate}
```

EXERCISES

1. Construct a Software Requirement Specifications (SRS) document for a Software System of your choice using LaTeX. Attach the printout of the .tex file and .pdf file.

2. Develop a brief document describing the basic functionality of an IoT based Health Monitoring System. Also include any block diagram/ image for explaining the overall functionality of the system. Use LaTeX. Attach the printout of the .tex and .pdf files. (Feel free to you use the other available formatting options in LaTeX).