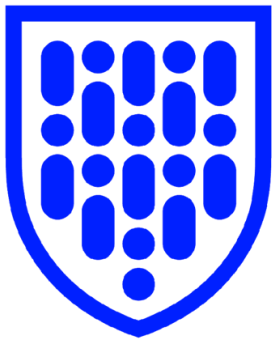

Institute of Data

2022

Software Engineering

Documentation

Institute of Data

Why Document things?

- People easily forget what was said or agreed which can lead to misunderstanding, confusion and wasted time
- Trying to achieve an objective without documentation creates a very high risk of failure, job loss and reputational damage
- The quality of your thinking is demonstrated by the quality of your written documentation
- Other people will duplicate your work in the future if there is no documentation to rely on
- Great documentation results in project approval, promotion and remains an artefact of your success for years to come

Institute of Data

Remembering Your Objective

DEFINE, DESIGN, DELIVER

- **You are the solution**
- **Define** the issue (problem or opportunity?)
- **Design** the solution
- **Deliver** the results

3Ds mapping to Documentation

Typical Document Sections	3xDs stage
Background & Objectives	Define
Assumptions	Define
User Stories and Requirements	Define
Open Questions/Out of Scope	Define
Designs, Mockups, Flow Charts etc	Design
Tasks	Deliver
Results and Conclusion	Define

Writing Style - SAW!

Short

- Sentences - use full stops. Avoid using “and”
- Paragraphs - break them up. Use bold, sub-headings & lists

Active voice

- Active (subject performs action): *“Customers believed their account was suspended when they saw the alert”*
- Passive (subject receives action): *“It was believed by customers that their account was suspended when they saw the alert”*

We focused

- Generally refer to “we” and not “I”

When to Document

Waiting until the end of a project to write things up doesn't work.
So record and document continuously:

- At the start of a project
- When stakeholder feedback has been obtained
- During project meetings
- If a project decision has been taken - record it!
- At the completion of a project
- After the results of a project have been obtained

Institute of Data

What to Document

- Project briefs - capture verbal briefs in writing and email
- **Project plans - continually maintain**
- Meeting minutes - attendees, agenda, decisions, tasks
- Designs - wire frames, flow charts, screen designs, architecture
- **User requirements - [persona, feature, reason]**
- Functional requirements - technical detail
- Tasks - person and due date
- Presentations - beginning, middle & end

Where to Document

- Confluence
- MS Office
- Google Docs

How to Document

- Use logical structure
- Always have a beginning, middle and an end
- Use a project document template
- Remember to use a SAW
- Remember to Define, Design and Deliver.

Institute of Data

Types of Documentation

- System Documentation
  - Requirements, architecture design, source code, validation documents, testing information, help guide
- User Documentation
  - User stories, User flow schemes
- Process Documentation
  - Activities related to product development

Institute of Data

Document Macro

Note: The following are the candidate sections of the document. They are presented here for guidance. Questions in each section could be used as possible aspects to cover. Some questions may not be applied to each project. On the other hand, additional information may be needed.

Introduction

Purpose

- What is the problem or the opportunity that the project is investigating?
- Why is this problem valuable to address?
- What is the current state (e.g. unsatisfied users, lost revenue)?
- What is the desired state?
- Has this problem been addressed by other projects? What were the outcomes?

Industry and domain

Industry/ domain

- What is the industry/ domain?
- What is the current state of this industry? (e.g. challenges from startups)
- What is the overall industry value-chain?
- What are the key concepts in the industry?
- Is the project relevant to other industries?

Stakeholders

- Who are the stakeholders? (be as specific as possible as to who would have access to the software)
- Why do they care about this software?
- What are the stakeholders' expectations?

Requirements - Examples of User Stories

#	User Story Title	User Story Description	Priority	Additional Notes
1	User inactivity	As a user I want to be automatically signed out if there is 5 minutes of inactivity so that my personal details are more secure.	High	
2	API	As a programmer I want access to the app's checkout function so that I can enable it easily in my own future application.	Medium	
…		As a [persona] I want [feature] so that [reason].		

Design

Architecture Diagram

Include a diagram of the building blocks of the design including users and how they interact with the product.

User Flow

Present as a flow diagram the steps a user may make in interacting with the software.

Wireframe Design

Show elements of the user interface, either manually or via a tool such as Figma.

Institute of Data

Non-functional requirements

- What are the key security requirements? (e.g. login, storage of personal details, inactivity timeout, data encryption)
- How many transactions should be enabled at peak time?
- How easy to use does the software need to be?
- How quickly should the application respond to user requests?
- How reliable must the application be? (e.g. mean time between failures)
- Does the software conform to any technical standards to ease maintainability?

Institute of Data

Solution

End-to-end solution

- How well did the software meet its objectives?

Institute of Data

References

References

- Where is the code used in the project? (link to GitHub)
- What are the resources used in the project? (libraries, APIs, databases, tools, etc)

Institute of Data

How to deal with time estimates?

“How long will it take?”

- Answer a question with a question - “When do you need it?”
- If an executive gives you a poorly defined request and demands to know how long it will take. You can respond by saying:
  - “I can design & deliver something that will meet your objectives fast, but I’ll need to define the project in more detail, then I can give you an exact timeframe….
  - “So tell me, what is the primary objective?”
- Projects should never be longer than 6-8 weeks
- If projects get too long then break them into phases
- A 3-month project is a warning that project definition might be poor or ill-conceived.

Fail to plan, plan to fail

Typical Document Sections	3xDs stage
Background & Objectives	Define
Assumptions	Define
User Stories and Requirements	Define
Open Questions/Out of Scope	Define
Designs, Mockups, Flow Charts etc	Design
Tasks	Deliver
Results and Conclusion	Define

Institute of Data

Presentations and Review