

# Technical Documentation

Synnøva O’Gorman

15.10.2023

# 1 Lesson 1

Today we will be talking about:

- Introducing ourselves
- Content of the course
- Course structure
- Expectations
- Technical documentation in testing
  
- Five-step process for writing a document
- Planning a document
- Audience analysis
- Defining objectives and scope
- Selecting tool and techniques

## 1.1 Introductions

My name is Synnøva O'Gorman and I'm from Dublin, Ireland. I have lived in Graz for 30 years now. My original training is in translation and interpreting, and I have a post-graduate degree in Technical Communication. I've been working as a technical writer at a software company for 25 years and for the last seven years, I've also been teaching technical writing at the FH Joanneum (Masters in Technical Documentation). I was also a member of the development team that designed the curriculum for that course.

## 1.2 Content of the Course

In this course, I want to teach you the basic principles of technical writing. We will cover such subjects as writing for specific audiences, planning documents, structuring paragraphs and sections, choosing the correct language, organizing information for easy retrieval, review and publishing strategies for documents, using visual aids, typography and layout, interviewing subject matter experts, text types and online documentation. You can see an outline of the lesson plan in the syllabus on Moodle.

The course is designed to deal with the life cycle of a documentation process, so first we will talk about planning tasks and then we will gradually move towards more granular issues.

I will also be dependent on you for input about the text types to deal with. All of the theory I'll be covering with you can be applied to all different types of text, but my background is very different from yours, and to make the course more tailored to your needs, I'll be looking for input on your particular documentation needs in the area of system test engineering.

## 1.3 Course Structure

Usually, I will deal with whatever the theory is for the lesson and then we'll try to do a practical exercise. Hopefully, there will be practical work each week, but some topics involve a little more theory than others.

There is no exam for this course. The grade is made up of the following elements:

- Planning, structuring and writing project: 30 points
- Language quiz: 10 points
- Graphics and layout project: 20 points

- Presentation: 20 points
- Class work: 20 points

The planning, structuring and writing project will be completed over the course of a few weeks. The language quiz is before Christmas. Class work is ongoing and the other two projects (possibly combined) will be carried out in the new year).

## 1.4 Expectations

Each lesson will have a handout like this one. These handouts are minimal and do not exactly comprise a "script" for the course. I expect you to listen and take notes in your document as I'm talking. All materials will be uploaded to Moodle the day before each class. There is a Moodle topic for each lesson and that's where you'll find any materials I used during the class or anything needed for an assignment. I commit to uploading class materials the night before your lessons.

You will also submit all assignments and class exercises on Moodle.

I like an informal class atmosphere and welcome questions and interruptions at any time. Please call me by my first name (if you can pronounce it!).

I am flexible on submission dates: if you have a high workload in a week when we've arranged a deadline, just let me know and we can change the date.

## 1.5 What is Technical Documentation?

What do you think of when you think of technical documentation? You probably think first of documents and texts that you encounter in your job, or which you maybe have had to create yourself in work. But technical documentation takes in many different types of text.



What kinds of things can be called technical documentation?

## 1.6 Technical Documentation in Testing

Now back to the specifics of technical documentation in test engineering. Here's where you get to teach me.



Think about the typical processes in test engineering, and describe documentation that is required (or helpful) in each process.

## 2 Five Step Planning Process

In your department, you will have your own processes, including perhaps a documentation process. And then you will be part of a bigger set of production processes. This can be a balancing act. For example, at what point in your process do you need input from SMEs? Does that coincide with a time when they are particularly busy? All these things are going to affect the process you implement in your own group, or even affect things on a micro-level for your own work.

### Some Vocabulary

**Stakeholders:** People who have a particular interest in a project or business activity. This can include your manager, project manager, project sponsor, members of project team, representatives of key business functions (risk management, legal counsel, health and safety).

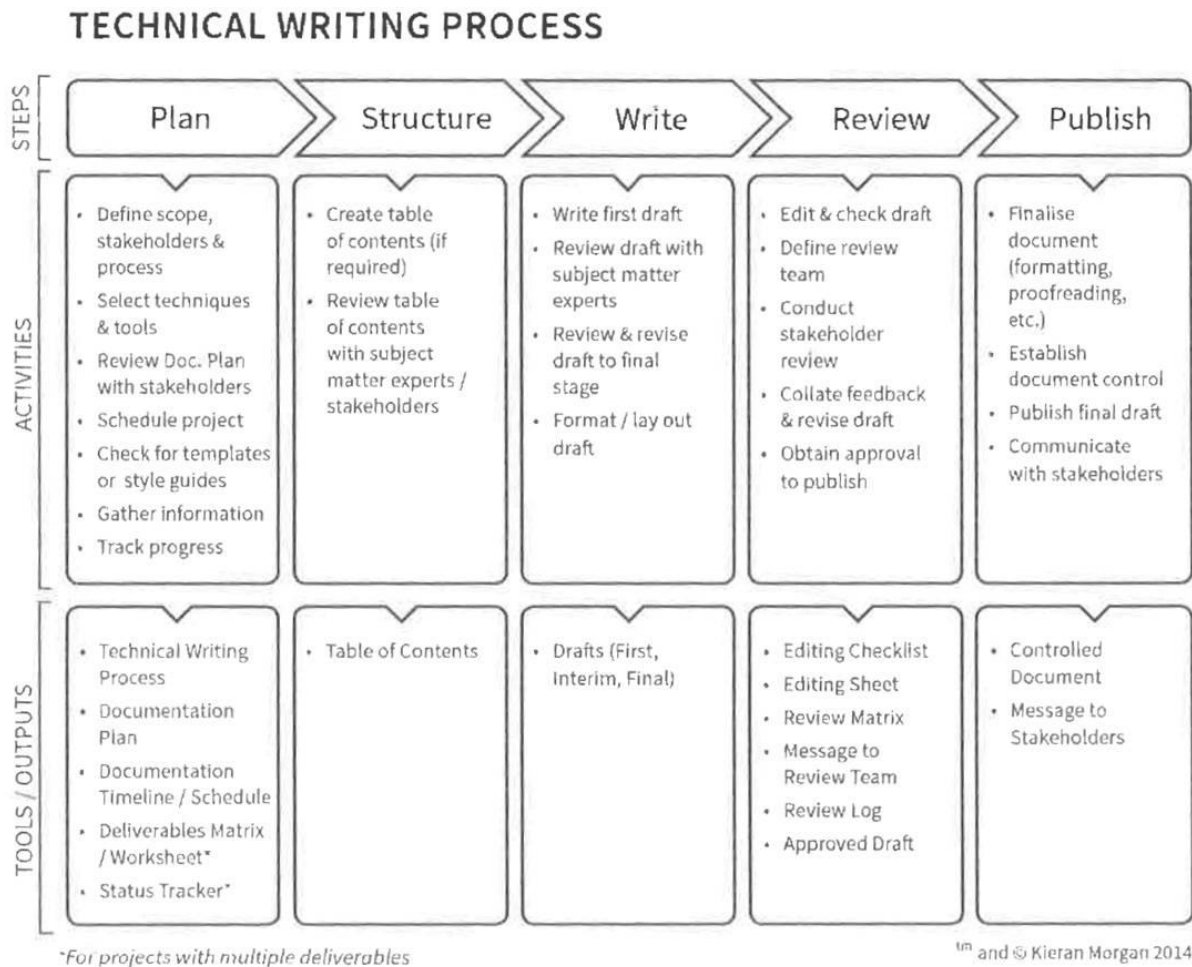
**Deliverable:** A tangible output of a particular process stage, e.g., Document Plan.

**Activity:** A task in the process or project schedule, e.g., define scope.

**Process:** A set of activities performed to accomplish a business objective.

**Scope:** The work that is done in a project. The product scope for a user guide, for example: intended audience, publication format, number of pages, page size, translation formats. etc.

## Five-Step Process (Morgan)



- What are the deliverables at each stage?
- Extend process if necessary, e.g., **Translate**.
- This process for *all* projects, but different emphasis.
- Time invested here is time saved later.

This is only an example, there are many different versions of this process. The basic steps are the same in all, just slightly different grouping or emphasis (depending on specialization).

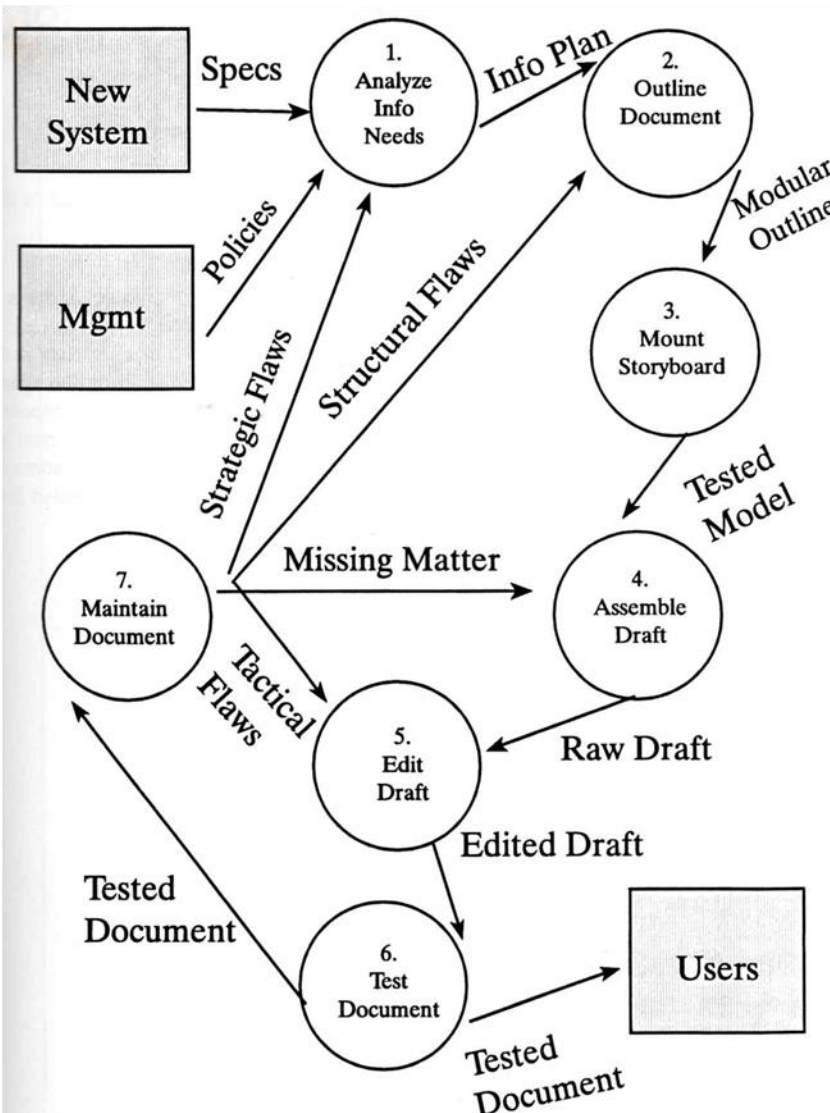
These five steps apply on all different levels of granularity. Essentially you go through this process not just on a document level, but also on a chapter or section level, or even on a paragraph level.

The process may need to be adapted depending on your particular project needs. So, this process diagram would be different if you had a "Translate" loop in there (before "Publish"), or maybe a "Design" loop...

Alternatives...

HomePrize Cloud Storage 2.0 Documentation Plan			
Deliverable	Reviewer	Milestones	Comments
<b>User Guide</b> New content for release. Additional appendix for keyboard shortcuts	Rob, Andrea, Lin	Draft 1: June 3 Draft 2: July 10 Final: July 16	Waiting to see if the Administrator function is being added.
<b>Help</b> Single-source from user guide for output on website, desktop software, and mobile device	Rob, Andrea, Lin	Final: July 13	
<b>Installation Guide</b> New section on upgrading from 1.0, appendix on user permission	Raj, Josh	Draft 1: June 21 Final: July 6	
<b>Release Notes</b> All new	Rob, Lin, Raj	Final: July 18	

Krista van Laan. This alternative is deliverable-driven (rather than activity-driven). It's obviously for a project with a simpler workflow, but works just as well as an overview.



Weiss. This graphic seems hugely chaotic but is just badly drawn (I suspect). It shows the input and output of steps quite nicely.

An alternative process description (Weiss):  
**analyse** needs  
 > **outline** information products  
 > **storyboard** each product  
 > **assemble** text and draft  
 > **edit** for correctness and readability  
 > **test** with representative users  
 > and **maintain** in the face of system changes and revealed errors.

## 2.1 Step 1 – Plan

### Summary:

- Define objectives (based on audience analysis)
- Define scope, stakeholders and process
- Select techniques and tools
- Review documentation plan with stakeholders
- Schedule project
- Check for templates, style guides
- Track progress

### Deliverables:

- Documentation plan
- Documentation schedule
- Deliverables matrix/worksheet (scope, multiple deliverables)
- Status tracker

Morgan also talks about a *deliverables matrix* and a *status tracker* and a few other things, but this depends on how many deliverables the project has in total, I'd say. If the project has multiple deliverables, consider a deliverables matrix - different guides will have different requirements (especially audience).

This might change substantially as the project progresses! Half-way through a project, you might realize you need a separate configuration guide, for example.

### 2.1.1 Audience Analysis

The most important activity in the planning stage is audience analysis.

Your audience determines

- Which documents to write
- What topics to include
- How to organise information
- Level of detail
- Level of examples and graphics

### Audience Analysis

- Describes audience's background
- Describes audience's level of experience
- Draws conclusions about audience and their needs

Who is the audience? Try to capture this in **one sentence** first.



What are some of the characteristics you might list?

Consider different audiences for same document and how you can cater for that. Don't forget audiences that are not immediate end users (e.g., regulatory bodies)

Consider also the same audience going on a journey that will have different requirements along the line.

Ask **Who? Why? When? Where/How? What?**

What about the audience's level of experience?

For simplicity, let's identify three levels: beginners, intermediates and experts (see tables below).

These levels can apply to many categories, so while the user might be an expert in the subject matter you are writing about, they might be novices in the software being used, for example.

Make conclusions about audience and what they need (see above also)

There can be a wide variation in level of experience. What content does each audience need?



## Beginner Users

What the user needs	What this means for content
They don't know how to get started	Getting started information Tour Short tutorials
They want simple, task-related instructions to help them succeed.	Basic procedures Examples Simple exercises
They make common errors that they don't know how to fix.	Troubleshooting procedures
They are not interested in product architecture.	Minimal conceptual information
They want to find information quickly and easily.	Index Clearly labelled procedures Not too many heading levels

## Intermediate Users

What the user needs	What this means for content
They want to understand the relationship between tasks so they can solve their own problems.	Basic conceptual information
They want to troubleshoot on their own.	Advanced procedures
They want procedure details only for tasks they want to do.	
They want to see a more efficient way to complete a task.	Shortcuts
They want advice from expert users.	FAQs Online forums

## Expert Users

What the user needs	What this means for content
They use the product in ways that were never originally intended.	Detailed conceptual information.
They may know more than the original developers.	
They want to exchange information with other users and original system designers. They are a primary source of knowledge about complex tasks and problems.	Online forums KDB User groups Chat systems Tips
They want to look up procedures on tasks they haven't done.	Advanced procedures
They want a reminder of specific steps or functions.	Reference card/guide Shortcuts

### 2.1.2 Defining Objectives

Why is it important? It will guide your structuring and writing and stop you getting into problems along the way. Also, if all stakeholders sign off on objectives and scope, there will be less work to do at review stage.

Your document plan is a contract for what you're going to deliver (high level planning and descriptions)!

Describe the final result you desire. What do you want to accomplish with your piece of writing?

Keep this reader-centred, so not "I want to describe the procedure..." but "I want new users to learn quickly how to perform ...".

The desired outcome is very much dependent on the users' needs.

On a high level, this is the same for all users: all users are looking to locate, understand and use the information and ideas that are of interest to them. But this can vary based on situation. This will inform the *scope* of your documentation.

### Answering User Questions

You can imagine yourself carrying on a conversation with the user while they are trying to perform the task.

### Persuasive Strategies

Not often a case in techdoc, but it might be for you?

### 2.1.3 Define Scope

Also, what *not* to cover

### 2.1.4 Identify Stakeholders

Manager, project manager, members of project team, risk management, legal counsel, health and safety. Don't forget, not all stakeholders are on the same page about the importance of documentation!

Some projects require cost forecast.

TIP: it pays to know who's in the zoo!

TIP: the benefit of consensus. highly consultative, iterative process pays down the way.

### 2.1.5 Select Tools and Techniques

E.g., Single-sourcing or CMS

Structured content with XML

It depends on your output. Unless it is a completely new document project, you will probably have tried-and-tested tools. Techniques will be dictated by your team set-up and what you want to deliver.

Modular (single-sourcing) requires different tools.

Technical writer = content developer.

## **2.2 Step 2 – Structure**

### **Summary:**

- Create ToC
- Think back to UX questions and audience profile
- Different approaches to structuring a document
- Depends on doc type, but also can be mixed!
- Review ToC with stakeholders/SMEs

### **Deliverables:**

- Table of contents

When I say ToC, I mean an internal information architecture, even if you are not producing classic print documentation with a table of contents at the front. The hierarchy of headings, as it were. We will be covering this in your assessment.

I told you we'd be referring back to our audience analysis at every step! Here it is again! What the users want to achieve with your documentation will dictate how you structure your information.

## Structural Approaches

There are different structural approaches (in the next few slides). Most likely, any single piece of documentation (let alone a whole suite of documents) will contain several of these structural approaches.

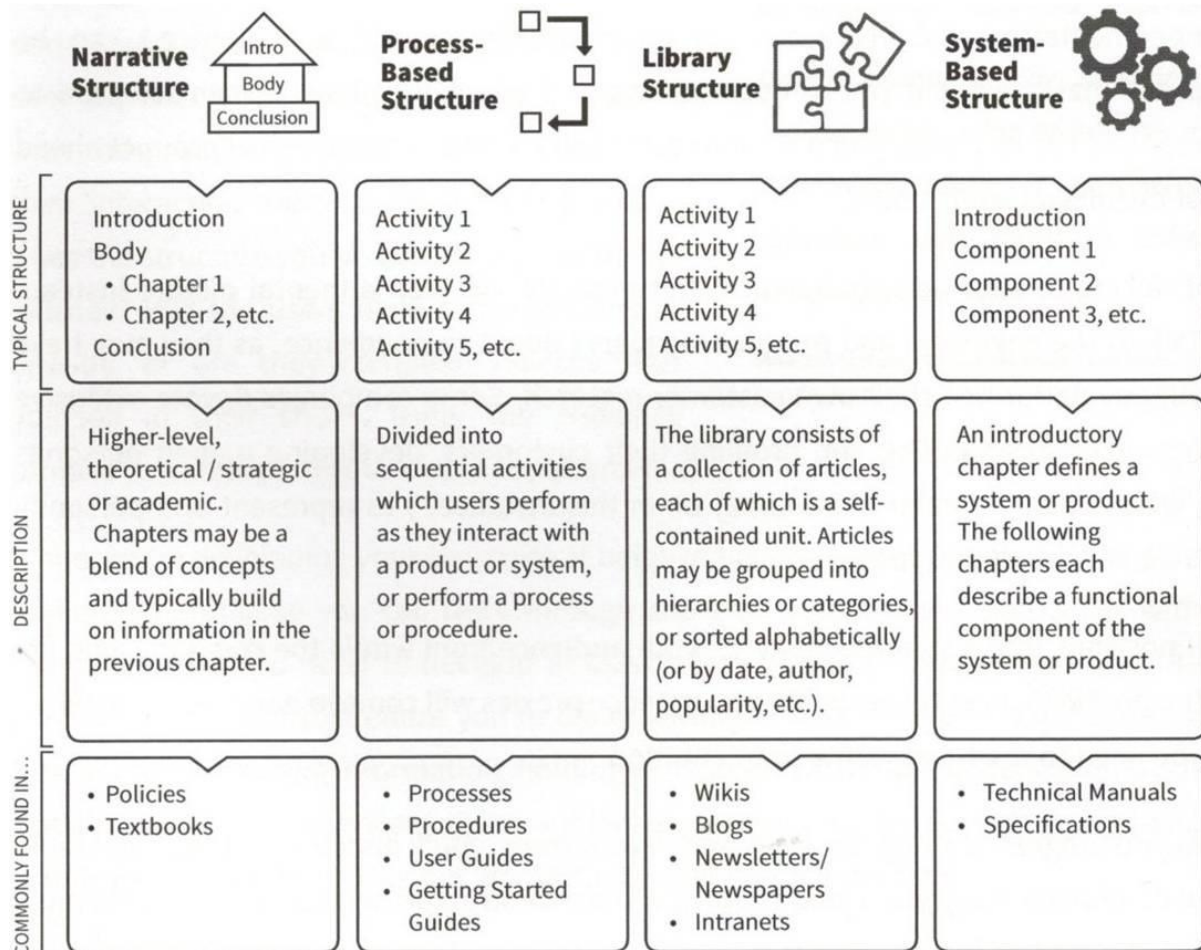


Figure 9: Four common structures in technical documents

### Narrative structure

This begins with an introduction, is followed by the body of text (covering all main topics) and ends with a conclusion. Typically written to be read sequentially.

This structural approach is often used for longer documents that are conceptual: academic papers, reports, essays. In techdoc, it's often only found in introductory and overview sections.

### Process-based structure

Here, the document architecture is mapped to an underlying process, sequence of activities or use cases.

In techdoc, this approach is fundamental for processes, procedures, step-by-step instructions, etc. When using this structural approach, you will typically develop a process model (high-level process, related sub-processes, etc.) which informs the document.

### **Library structure**

Here, content is viewed as a set of independent articles or documents. E.g., wikis, blogs, intranets, reference guides. It usually includes the facility to categorize or groups articles (using metadata). Articles are interlinked. Online newspaper. Company intranet.

### **System-based structure**

Here, the documentation describes a system, e.g., introduction to system > components > detailed description of components. Sequence is not as important.

In techdoc, this approach is used for technical and mechanical specifications, system architecture, etc.

#### **2.2.1 Creating Table of Contents**

Decisions to make for your table of contents:

How many levels does the documentation require?

For process-based structures, the process model will dictate the ToC.

Similarly for system-based structures, where system components will inform how you structure your document.

At this stage, you might find it useful to populate the headings to guide your writing. [We will be talking about headings and their importance in a document in a later class. We'll practice writing headings in English and what you need to pay attention to.]

Don't be surprised if your ToC changes as you complete your writing. When filling in the details, you may realise the relationship between components that you didn't see before, or the importance of one component compared to another. Sometimes an overly hierarchical structure can "hide" important features too far down a doc structure.

Your ToC should reveal the organization of your document to your readers. It will help readers to realise what the relationship is between various segments.

Don't forget to have your ToC signed off on by stakeholders!