REQUIREMENTS FOR WRITTEN REPORTS

Introduction to reports in Project

The ability to prepare high quality documentation is a very important skill that each engineer has to demonstrate throughout his/her career. The reports prepared in Project will lay the foundation for this.

> You have to be proud of your Project reports.
>
> Consider this: your final report (at the end of the year) is the *only* permanent record of your work. Even if you did excellent technical work (either in Project or further on in your career), a poor report will always create a bad impression of your work.

▶ Reports of *very* high quality are *expected* in Project; this is the norm, not the exception.

Consider buying a good textbook[1] on technical writing, and also concentrate in class, where the Project Proposal and other project reports will be discussed in some detail.

You will find that different employers require technical reports in different formats in your future career. Details on the required *content and format* of the project reports in this module appear in the appendices that follow. You have to adhere to this format strictly. Note that the requirements differ for the various reports.

Incomplete or unsatisfactory reports WILL be penalised HEAVILY and will have to be revised.

Instructions for the various reports appear in Appendices 2, 3 and 4 of the Project EPR400/EPR402 study guide.

---

[1] E.g., Beer,D. & McMurrey,D., "A Guide to writing as an engineer", John Wiley and Sons, New York, 1997

APPENDIX 2 TO THE STUDY GUIDE

REQUIREMENTS FOR THE PROJECT PROPOSAL

(Rev 0 of 2022)

INTRODUCTION

The project assignment you received from your study leader is a *concept* proposal, which you have to develop into a full project proposal. The document generated is called the Project Proposal.

Once approved (through the Project Proposal submission and approval process), this proposal will then be your agreement with your study leader and the Project EPR400/402 lecturer (Prof JJ Hanekom) on the deliverables of your project.

► The *project definition phase* is only complete when the Project Proposal has been approved by the Project lecturer.

Note that the final, *approved* Project Proposal must be bound into the final report at the end of the year. Examiners will use this as a reference against which your outputs will be evaluated.

Content and layout requirements

It is ***compulsory*** that the Project Proposal should contain *exactly* the layout and contents described below.

► Use the provided template.

The template is a pdf form which can be completed using Foxit Reader or Adobe Reader.

More detail on what is required will be discussed during the lectures. Examples of previous Project Proposals will appear on the Project website.

Formatting

Formatting is to a large extent fixed by the pdf form. Kindly don't create your own pdf form, but use the template provided.

The template limits the number of characters in each section. This is how much space you have. Part of the challenge in writing a high quality document is to choose your phrasing carefully. The template that limits the amount of space forces the student to do so, and to distinguish between more relevant and less relevant aspects.

► An important writing rule to take note of is that *each sentence in the document needs to be a full sentence*. Even where you use bullet lists, each bullet has to contain proper, grammatically correct, full sentences.

TITLE AND APPROVAL BLOCK

The first page (see the template) contains a title block that also serves as an *approval block* that contains declarations by your language editor, your study leader and yourself.

Project number
This is the number appearing on the Project website, e.g. TH8 or LZ5.

Revision no
The first official submission (not those to your study leader in earlier iterations) will be Rev 0. If a revision is required, you will submit Rev 1, and so on.

Project title
Use the project title that appears on the project concept note. This is the title to be used throughout all of your reports, unless changed with written permission from the Project Lecturer (Prof JJ Hanekom). You may not change your project title from report to report.

Study leader
The study leader (whose name appears on this page) is your assigned study leader as indicated on the Project website**.**
By signing, the study leader confirms that he/she has had enough time to read and comment on the document and approves of the content.

If the study leader is not available to sign the document, write "see attachment to the Project Proposal" in the signature block, and attach an email message from the study leader confirming that he/she has had enough time to read and comment on the document and approves of the content. Attach this after the last page of the Project Proposal.

Language editor
The language editor, as explained in class, may be yourself or a different party that you trust to do this *properly*, taking note of the fact that very high quality documents are required. By signing, the language editor confirms that he/she has had enough time to edit the document and that he/she has edited the entire document.

If the language editor is not available to sign the document, write "see attachment to the Project Proposal" in the signature block, and attach an email message from the language editor confirming that the task was completed. If revisions are required, all changes and updates need to be language edited as well.

Student declaration
Ensure that you do understand exactly what plagiarism is and what it would entail in the context of Project before you sign this.

CONTENT OF THE PROJECT PROPOSAL

The descriptions below consider each section of the Project Proposal in the order that they appear in the Project Proposal.

1. Project description

This is an overview section that briefly introduces the reader to the concept of the project, before jumping into more technical details in the sections to follow. This will probably be very similar to the project concept note that appears on the project website. You may use an edited version of the concept note, and would probably expand on this to ensure that you answer the questions below.

> This section answers these questions: what is the project about? What does the system have to do? What is the problem to be solved?

The focus in this section will be on the problem to be solved and perhaps the context thereof, and will only briefly touch on the proposed solution. You may commence with a phrase like "The problem addressed in this project is...". A paragraph commencing with a sentence like "The idea with this project is to develop a system that solves this problem by…" may be an appropriate introduction to a brief description of what solution you have in mind. Further sections will expand on the latter in more technical terms.

The following is an example (example text in blue).

> The problem addressed in this project is the growing rate of crime in South Africa. Existing video surveillance of sensitive areas does not appear to be an adequate deterrent any more. Thus, the problem addressed in this project is to develop a new kind of multi-technology surveillance system based on audio and video data to act as stronger crime deterrent.
>
> The concept is to try to detect a potential crime before it happens by analysing stress patterns in voices of speakers, and by analysing movement patterns (like stealthy walking patterns as opposed to normal walking patterns). To achieve this, a non-PC-based processing platform will be developed that does all of the necessary processing at the point where the sensors (video and audio) are installed.

2. Technical challenges in the project

2.1 Primary design challenges
2.2 Primary implementation challenges

By the very nature of the module, Projects intends to provide appropriate technical challenges.

Some challenges will be on the design side (e.g. difficult theory). Some will be on the implementation side (e.g., while the circuit equations for very high currents would look similar to those for small currents, real world implementation issues make the former far more difficult to work with).

The idea is that you separate foreseen and intended design challenges and foreseen implementation challenges into two subsections.

The challenges in some projects will be mainly in the "design challenge" category, and in other projects these will mainly be in the "implementation challenge" category.

The objective in these two subsections is to identify the technical engineering challenges *explicitly*.

Note: you may use bullet list format or paragraph format, but always write in complete sentences.

The engineering design challenge is that aspect (or those aspects) of the project that would make the project a technical challenge for a typical final year engineering student. The interest here is not in the scope of the project, but rather in the parts that will be *technically complex* or *difficult to achieve*. Tell the reader what you foresee those to be.

An example of a *technically complex* aspect may be to use estimation and detection theory methods (which can be mathematically complex) to extract a weak signal hidden in a large amount of noise. These are not aspects covered in the undergraduate curriculum, and will provide a challenge to the student.

An example of an aspect that may be *difficult to achieve* may be where a particular specification is rather strict. It may be that the underlying theory is not that complex, but that it is particularly difficult to achieve the strict specification because of the use of (for example) high currents and the inherent design challenges of working with these, so that it would only be possible to achieve the specification through very careful design.

A project may also provide challenges because of the *technical limitations* or *restrictions* that the student (or engineer) has to contend with. Therefore, consider these as well when writing this section. For example (example text in blue): Limitations to contend with include the restricted available bandwidth. Because limited bandwidth is available, video data will have to be compressed before transmission.

If you discuss limitations to contend with, discuss *technical* limitations only, and not any others. E.g., *don't* talk about time limitations (this fixed by the nature of the module), and *exclude* personal limitations (e.g. own technical ability, or own ability to write).

3. Functional analysis

This section answers the question, HOW will the problem be solved?

This is a description of your approach to solving the problem in the form of a functional analysis, in which you have to show a first systems-level design in the form of a block diagram. The intention is to show clearly how you visualise the design *at functional level*.

The design shown here is part of the Project Proposal that will be approved, *and as such you need to ensure that it captures the intention of the project clearly*. However, this will not necessarily be exactly the final form of the design; while the core idea will remain the same, small variations are to be expected. Major deviations are subject to further approval.

Note the following.

1. The functional analysis considers the solution from a *functional viewpoint,* and *not* from an *implementation viewpoint.*
2. The functional analysis should focus on your *design* or *product,* and not on your tasks as student. I.e., the functional units on the functional block diagram should *not be your tasks*, but rather *the functional units of the product* that you have to develop. Said in another way: the functional analysis is used to break the final product down into its functional units, and is ***not** a flow diagram of your tasks or the steps in the design*.

If your project contains more than one product (for example, a hardware part and a software part), it would often make sense to have two or more separate functional block diagrams.

The functional analysis section should contain a *functional block diagram* (or more than one; section 3.2) along with a *complete description* of the system functions in narrative format (story format; section 3.1). In the latter you should describe the functional flow in narrative format in one of more paragraphs, as opposed to giving a list of definitions of what each functional block is.

As an example, if you have a functional unit (FU) that does filtering, you may say something like (example text in blue): FU2, the high pass filter, filters out the low frequency components of the signal, and suppresses 50 Hz noise, which would be the *correct* way to write this.
The *incorrect* way would be to give a definition for the task of each functional block. For example (example text in blue) FU2 is a high pass filter is a definition, and not descriptive of the task that you intend this filter to accomplish within the system.

Define the inputs and outputs explicitly. Note that there will be a strong correspondence between the requirements and specifications (given in the next section) and the functional block diagram.

The functional block diagram should be prepared as a figure in a drawing package of your choice (and *may not* be drawn by hand). You will then be able to import this into the pdf form by simply clicking on the open area in section 4.2.

4. System requirements and specifications

At the core of what one would like to communicate with the Project Proposal are the *requirements* and the *specifications*. These have been discussed in some detail in class. In brief, the requirements tell us, in the language of the client, what the main things are that the product should do. The specifications translate these requirements into measurable items in technical language.

This is the core section that defines the project, and is completed in table format. The idea is to define the set of requirements that describes the system as a whole (the mission requirements), and then to add details (the specification corresponding to each requirement, how you will demonstrate that the requirement was met, and so on, as detailed below). The first row describes the requirement (column 1 would be requirement 1, column 2 will contain requirement 2), the second row the specification corresponding to this requirement, the third row a motivation for the specification, and so on.

Row 1: requirements

The right place to start is to visualise the product and what it should do (create a picture in your mind, or on paper, of the physical setup).

Next, describe the set of requirements that describes what the system or product as a whole needs to be able to do (functional and performance requirements).

Refer to the example discussed in class in which we visualised the concept with the project first, then wrote down a set of requirements (around seven in the class example) that captured what we wanted to achieve, and then went on to translate each requirement into a specification.

Step 1 is to develop the set of requirements. These will be captured in the columns of the table in section 4, with the intention being *that requirement 1 should be the most fundamental requirement that captures the core task of your system*. The requirements are given more or less in order of importance, although they may be equally important and should together capture the complete functionality required of the system.

> Requirements capture the required functionality of the system *in the language of the client* and answers the question "WHAT"

Note that you will have to separate the *requirements of the product* and *your own tasks,* i.e. the tasks that you will perform to achieve these requirements. The latter goes into section 5.1.

You should provide a description of what the *product* should do. I.e., describe *all* aspects of the system as visualised, including where it fits in (e.g. part of a larger system with specified inputs and outputs).
However, you should not include any requirements that cannot be classified as mission requirements *of the system* or *of the product* to be developed. For example, some of the deliverables may include simulations that need to be completed before the final product is implemented. These would typically *not* be part of your final product or system and should not be expanded on under "requirements". Rather, describe these in section 5.1 (*your* tasks).

Row 2: specifications

Once the requirements that capture what the system has to be able to do have been written down, one can attach a specification to each requirement. This is not necessarily easy to do. It is certainly no thumb-suck or guess – it requires **work**, it requires **reading**, and it requires **applying your mind** to arrive at appropriate specifications.

> Specifications capture the required functionality of the system *in technical language, i.e. in measurable terms,* and answer the question "HOW WELL". I.e., how will we know (measure) unambiguously that each requirement has been met?

Line 2 in the tables says "Target specifications", and not simply "Specifications" for a reason. For most projects it is easy enough (after having done the necessary homework) to write down specifications that the system has to meet, even though these may be challenging to achieve. There are examples, however, where one may not know exactly which specifications are achievable or appropriate, but where one will have some target in mind that will solve the problem - therefore the more generic term "target specifications".

The correct and the incorrect way to develop specifications

> ❗ When thinking about specifications, it is very important to understand that specifications exist to solve a particular problem. Specifications are derived *first*, *before* thinking about a solution.

It would be incorrect and poor engineering design to say something like (example text in blue) the load specification is 1 Watt because I'm using power transistor that can supply at most 1 W to the load. Can you see why this is wrong? The design choice should be made to meet the specification. The specification is selected to solve a particular problem. A specification may never be based on a design choice.

Here is a simple example. A farmer has a borehole of 200 meters deep on a farm in the middle of nowhere, and needs to provide solar power to a borehole pump. The engineer needs decide on a borehole pump (given, among others, the required water flow rate) and then needs to design the off-grid solar powered supply. All specifications need to be selected in order to solve the problem; it wouldn't help to guess and simply buy a 1 kW borehole pump, and then to say that the specification for the solar power supply is to supply a load of 1 kW because a 1 kW pump has been bought. If you do this, your chosen specifications *will not solve the problem*. You will have chosen a specification based on a component choice, rather than to select the component to meet the specification.

You can never, ever select specifications "because they would be easy to achieve"! You must always select specifications that, if met, will solve the problem.

Specifications at systems level, vs. detailed specifications

Many details will not be available yet, and you are not expected to complete a design before you develop the specifications. You developed a set of requirements for the system as a whole, and the corresponding specifications will also describe the system as whole. These are the *mission-critical specifications.* I.e., specifications, if met, will solve the particular problem. If not met, the system will not function as required, or will perhaps not function at all. Component level specifications and detailed specifications of each subsystem are *not* required at this stage and will only be developed later on in the project.

For example, when you design a digital communication system, aspects like desired data rate may be important, but details on the power supply may not be. *It is important that the student demonstrates that he/she can distinguish between important system functionality and secondary tasks* (data rate vs. power supply in the present example).

Formulation of the specification

The specification should be given in measurable terms (usually in numbers, e.g. 12 Volt, 100 kbit/s, or 1 kW). Specifications should always be in measurable terms, whether measured with an instrument (e.g. an oscilloscope, a Wattmeter, a lux meter, a sound pressure level meter or another instrument) or measured against some standard or yardstick.

Note: a specification *may not be a description of a task*, but should be a measurable specification.

Specifications may be expressed as *performance specifications* (typically measured with an instrument) and/or *functional (or operational) specifications* that specify functionality or operation. For example, you may have to implement a specific communication protocol. There will still be a yardstick against which you measure in this case, although there may not be a particular single instrument with which to measure whether the specification was achieved. E.g., a student may have to implement the USB 3 protocol correctly, and this will be measured by communicating correctly with a USB device.

Specifications may also be thought of as being *mission-critical* or *design-critical* (e.g., to drive a 1 kW load, you will need at least 1 kW of input power), and if not met, the design fails; or as *target* specifications (e.g., 80% correct identification for a speech recognition system). Note that in the latter example, environmental factors (that should be defined under "field conditions" below) will influence the choice of target specification, and these should also be specified clearly (e.g., the speech recognition should be at least 80% correct when isolated words are spoken in quiet conditions with background noise level smaller than 20 dB SPL).

Specifications to consider may include: tolerances, signal-to-noise ratios, required quality of the signal, perhaps specifications in terms of test programs, test vectors or test signals to be used. Specifications may be in terms of what has been achieved before as described, for example, in journal articles. This may used to define target specifications, e.g. (example text in blue) The expected outcomes should match those of Jones and Mokoena (2017).

Finally, specifications may sometimes be described as measurable requirements with which the deliverables have to comply. E.g., if one of the deliverables is to perform an experiment at different signal to noise ratios using different algorithms, the specification may read something like (example text in blue) The experiment should be repeated at SNRs of -10 dB, -5 dB, and 0 dB for each of the algorithms A, B and C. The experimental outcomes should be compared for noise suppression and the trade-off with complexity of implementation.

More examples that expand on how to formulate specifications were discussed in class.

Row 3: motivation for the specification

This row should be used to explain how you decided on each particular specification. Read this as meaning "motivate where this specification comes from", or "explain why this specification will solve the problem or ensure that the requirement is met". I.e., if the value in row 2 is 10 A or 1 kW, *how did you decide that this particular value will solve the problem*? This description goes into row 3.

To reiterate this point, the phrase in the previous paragraph is printed in italics, because this is the only valid way to decide on a specification: i.e., decide on the specification that will solve the problem at hand, and *not* on a specification that will be easy to achieve, or that complies with your selected solution. E.g., it would be *invalid* to decide to use a particular microprocessor, and then to base specifications for processing ability on that choice.

Row 4: demonstration at the examination

This is an explanation of how you will confirm and show at the examination that the requirement and the specification in the present column were achieved. Write this in *precise terms* (e.g. (example text in blue) An ammeter and voltmeter will be used to measure RMS current and voltage, and power will calculated from this, rather than Power will be measured.)

Rows 5 and 6: designed and implemented by the student vs. taken off the shelf

The requirements capture the deliverables, but do not explicitly indicate which aspect will be designed and implemented by the student, and which aspects may be off-the-shelf. In these two rows, indicate explicitly whether each deliverable (the deliverable implied by each column) will be designed and implemented by yourself from basic principles, or will be taken "off-the-shelf". The latter means that a complete subsystem is provided by a third party, e.g. bought, downloaded from the internet, or usage of libraries).

5. Field conditions

After having developed specifications, define the field conditions in the present section. The field conditions are *the actual conditions in which your system should be able to operate*. Often, the real engineering challenge is not to make something work in ideal conditions (lab conditions), but to get it to work reliably under more difficult real world conditions.

You will see that, similar to section 4, there are columns for each field condition. Row 1 describes the requirement, while row 2 gives the specification.

Row 1: field condition requirement

Requirements are given in the language of the client. E.g., if you design a device that should operate in a motor vehicle, you will have to mention this as one of the field conditions. Field conditions may be described by requirements like: "the product should work in a moving motor vehicle"; "the product should operate in noisy conditions"; "the product should work outside"; "the product should work under ideal conditions or lab conditions"; "The actual product should operate within a few percentage points of the simulation"; "PV panels will be used in actual sunlight conditions, not with artificial light".

One aspect to clearly define under the "field conditions" heading is the scale of the project. For example, there is a huge difference between a lab test model of an irrigation system, and an irrigation system on a farm. The former would not qualify as a final year project.

Row 2: field condition specification

Now that you have described the field conditions in vague or non-technical terminology in the first row, you need to translate these into measureable conditions. These conditions may be similar to these examples: "The PV panel should produce the specified output and should charge a battery in normal sunlight and partly cloudy conditions typical of those on a Pretoria summer day at 12:00"; "The signal should be extracted from background noise at a SNR of 0 dB or better"; "Image processing should be done on moving images captured from a vehicle moving no slower than 30 km/h"; "Measurements should be taken and processed in real time on an actual athletic track under night conditions with floodlights providing 250 lux".

6. Student tasks

6.1 Design and implementation tasks

The idea is to capture *your* tasks in this subsection, and *not* those of the system that your will design. Use bullet list format (always using full sentences) and list the tasks that you will need to complete in order to meet the product mission requirements. These may for example include reading on the subject, doing a paper design, doing simulations, building, testing and evaluation of the design.

Additionally, this section should be used to briefly describe the experiments that you will set up and carry out to show that the system meets the requirements. Tasks may include the design and development of test setups and experiments, experimentation through simulations, and actual experiments or measurements with or on your system. If you need to develop a test setup to prove that your system meets its specifications, this should be discussed in this section.

6.2 New knowledge to be acquired

This section should briefly describe (in bullet list format, but always using complete sentences) what the theoretical foundation for the project is and which new knowledge you will have to master in order to complete the project. New knowledge should include only aspects that are not covered in any other undergraduate modules.

FORMAT OF THE PROJECT PROPOSAL

The format is largely fixed by the pdf form.
Kindly use this form/template and complete with Adobe Reader or Foxit Reader. Do NOT create your own template.

1. **Language.** The report should be written in English.

2. **Spelling.** Spelling errors are completely *unacceptable*.

3. **Language editing.** Poor language and grammar are *unacceptable*. It is a requirement that your report is language edited by someone who either writes very well or is a professional language editor. The language editor's name and signature must appear on the Project Proposal.

4. **Style.** Write in formal style. You may *never* write in the first person. Do not write in a conversational style. Also do not give a chronological account of what you did. The report should be factual and not bound to you as a person. This is a technical report, not a personal narration of your experiences and ideas. Give the facts in a brief, clear and professional manner.

5. **Write in full sentences.** You may *never* write in any shorthand, SMS style or telegram style. Telegram style is a clipped form of writing where one packs the maximum amount of information into the minimum number of words, often leaving out conjunctions. Do not do this, but rather always write in full sentences.

6. **Bullet list format.** When using bullet lists, make sure you use the correct list format as described here. You need to always write in full sentences, and this is also the case for bullet lists. Your bullet lists will look like this (example text in blue).

   The following aspects constitute the primary mission requirements of the design.

   - The system cost must be limited.
   - Power requirements must be minimized.
   - User safety must be guaranteed under all operating circumstances.

   Each bullet contains a full sentence, commencing with a capital and ending with a full stop.

7. **Appendices.** No appendices are allowed in the Project Proposal.

8. **Footnotes.** No footnotes are allowed in the Project Proposal.

9. **Page layout.** *Use the template provided on the Project website and complete this with Foxit Reader or Adobe Reader.*
   Leave one blank line between paragraphs and do not indent the first word of a paragraph.

10. **Figures.** These should have consistent formatting.

- Figures may not flow from one to the next A4 page, and A4 sized figures are preferred to larger figures. The pdf form template will allow you only one A4-sized figure. If you have more figures (e.g. two separate functional block diagrams), save them as a single figure file before importing this figure into section 3 of the document.
- All figures should have the same orientation as the rest of the text.
- Text in figures should be in English. Use Times Roman font for the text appearing on figures. The minimum text size in tables or figures is 8 point font.
- Number figures separately from number 1.
- Do *not* use frames around figures. (The block into which you will copy the functional block diagram is already framed, so don't be concerned about this).
- Captions appear *below* figures. Figures should each have a short description printed in the caption.
- Print the figure number as well as the caption in bold. Below is an *example* of a caption. Example text appears in blue.

Figure 1. The figure shows the functional block diagram for the proposed system.

- Captions are left-justified and printed in 12 point font. Use Times Roman font for the caption.
- All figures must be referred to in the text of your document.

11. **References.** The Project Proposal does not have a reference list. Where references are needed, you may use in-text citations. Here is an example (example text in blue).

The original way to do this is described by Liu *et al.* (Liu, C. and Eddins, D.A. (2008). Categorical dependence of vowel detection in long-term speech-shaped noise. Journal of the Acoustical Society of America, 123(6): pp. 4539-4546).

* * * * *