

Instructions for Interim and Final Reports, Their Marking Rubric and Annotated Template

Fabian Gilson

Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand
fabian.gilson@canterbury.ac.nz

CSSE Lecturers

Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand

Abstract—For the last few years, the interim and final reports written by students enrolled in the Final Year Project of the Bachelor of Engineering (with Honours) in Software Engineering are varying in size, completeness, and quality. This situation has caused much frustration both for students and supervisors. Moreover, it appeared from course evaluation surveys and direct feedback that students need more guidance on the expected content and level of details of the aforementioned reports. Therefore, the present template aims at specifying the expected content as well as giving a common document template to harmonise the guidelines across supervisors of those projects. We also clarify the marking rules by providing a clear rubric to which supervisors and students will refer to in order to assess the quality of the submitted reports. We believe that supervisors and students will benefit from such document and the marking process will be more transparent and consistent across projects.

Index Terms—agile software development, documentation, model-driven development, natural language processing

I. INTRODUCTION

In order to clarify the expected content and quality for interim and final reports in the course “*Software Engineering Research Project*” (SENG402), we provide a template and a marking rubric. This document describes the requirements in terms of content and format the students have to comply to when writing their reports.

We advocate that complying to this template and referring to the supplied marking rubrics, students enrolled in SENG402 will be able to achieve high grades for these two assessments. Furthermore, this document will harmonise the marking and expectations across supervisors and increase the transparency among all involved parties¹.

In Section II, we describe the expected sections of the interim and final reports as well as the page limit. In Section III, we give pointers to the templates (Word and Latex) with additional resources. Then, we summarise a crash course on Latex in Section IV. In Appendix A, two separate marking rubrics are supplied, one for the interim report and one for the final report. Last, a minimum working example of a report written in Latex is given in Appendix B.

¹You may need to remember the reader if there is a Non-Disclosure Agreement by adding a footnote in your introduction, as we do here.

II. EXPECTED CONTENT

We describe the expected sections for the interim and final reports. Explicit mentions will be made in the section title if that particular section does apply to one report only. Every student is required to write its own report. The only section that may be (partially) shared is the “*Background and Objectives*”, where some overlap may obviously exist regarding the context and the main objectives. However, on top of the team main goals, individual objectives must be clearly identified.

In terms of expected length and content of the reports, we require the students to comply to the following rules:

a) *Objectives and plan*: The first report must contain the description of context, the objectives (see Section II-C) and a plan (see Section II-G with a set of **numbered** milestones). All milestones must identify actual and tangible outcomes (e.g., report, code, demo, ethics approval). The plan must contain risk-mitigation aspects where necessary (e.g., critical step in the plan may depend on the ability to get access some resources). This report must be at most **2 pages long**.

b) *Interim report*: The second report contains all sections listed hereafter except parts of the implementation details (see Section II-E1), and the evaluation and discussion (see Section II-F). This report must be at most **4 pages long**, references and appendices excluded.

c) *Final report*: The final report concludes your project. We expect to find all section, potentially except a plan (see Section II-G), unless all objectives presented in the first and interim reports haven’t been achieved. This report must be at most **10 pages long**, references and appendices excluded.

The reports **must be** written in the IEEEtran template, the exactly same format as the present document. In section III, links are provided to the template for both Word and Latex.

We expect the following:

- report is well written, *i.e.* with a clear presentation and organisation, correct grammar and spelling;
- reported work is significant (e.g., level of originality, quality of solution);
- content can be understood and assessed by any computer science and software engineering staff member, even outside of the project and domain;

- report refers to scientific literature where necessary;
- report **must be in PDF format**, any other format will be ignored.

Please refer to Tables III and IV in Appendix A for a comprehensive marking rubric.

A. Abstract

An abstract is a “codified” summary of a document and is typically composed of 150 to 250 words. It must be self-contained and awaken the interest of potential readers. As a general rule of thumb, an abstract should answer to these questions:

- what is the problem under study?
- why is it a (real) problem?
- what is the solution to the problem and/or outcomes?
- why/how is it an adequate solution to the problem?

You may see that the abstract of the present document follows the above structure.

B. Introduction

The introduction of your report must give a general description of the context of the project, its objectives and their rationale. You also may reveal the main findings or outcomes of the project (*e.g.*, research results, prototype). The goal of an introduction is to set the scene and let the reader understand what the report will be about. The interim and final reports are meant to be self-contained, thus an appropriate and sufficiently comprehensive introduction helps at handing the report to anyone that will be able to understand the goals of the conducted project or research.

It is important to back up all claims you do in an introduction by the literature if it is not your own findings or if you are not giving a full reasoning for that claim.

An introduction finishes by depicting the content of the report (see the last paragraph of Section I).

If the present work falls under a *Non-Disclosure Agreement*, this must be stated at the beginning of the introduction using a footnote.

C. Background and Objectives

This section gives enough detail about the surroundings of the project, *i.e.*

- environment in which the project has been conducted in terms of supervision (*e.g.*, staff, industry);
- description of involved third parties, if any;
- general organisation of the project (*e.g.*, team-based, collaboration with third parties)
- pre-existing results in the same project (*e.g.*, former year projects), if any;

The background section finishes by introducing the problem(s) the project tries to solve and its objectives. In case of a team-based project, both the higher-level goals (team-level) and individual objectives must be depicted. This section is the only one that may be partially common for all members of a team and must be clearly identified as such using a footnote at the beginning of this section.

D. Existing Solutions or Related Work

Every project has an history and tries to frame a solution that solves an actual problem (that has been introduced in the “Background” section). Therefore, either there exist solutions that are not fully satisfying, or that needs to be extended or combined to create the new solution.

This section serves as a review of the existing technologies or published research that is relevant to the present project. You are required to browse the scientific literature and, if applicable, existing technologies that could potentially be used to answer the problem addressed in this project but that are not fully satisfiable. For any discussed related work or solution:

- the relationship to the present project must be made clear (*i.e.* avoid irrelevant references to artificially *grow up* your bibliography);
- the reasons why (*i.e.* rationale) these pre-existing solutions are not sufficient must be explained;

By the end of this section, on top of understanding clearly the objectives of the project, it must be clear to the reader how your project distinguishes itself from the existing state-of-the-art or literature. You may propose a systematic comparison of existing solutions (criteria-based).

The University of Canterbury has subscriptions to a wide range of published literature, see <https://www.canterbury.ac.nz/library/search-our-collections/databases/>². You may also take a look at scientific search tools such as <https://www.semanticscholar.org/> or <https://scholar.google.co.nz/>.

E. Proposed Solution

This section is the core of your report. It must give sufficient credit to your work to demonstrate that:

- reported work is significant (for example, level of originality, quality of solution);
- extent of work is satisfactory (for example, complexity of topic, level of programming);
- own contribution is made clear (in case of team-based projects);
- all objectives are adequately addressed.

The two next sections may be switched in order, depending on the flavour of the project and preferences of the writer.

1) *Design and Implementation*: In this section, the design of the end product and its implementation must be described with a sufficient level of details. The description must give appropriate technical information about the solution:

- presentation and discussion of the (envisioned) technology stack;
- any design artefacts, *e.g.*, textual requirements or models, architecture or implementation diagrams, wireframes;
- description of the (candidate) solution;
- results achieved (so far), *e.g.*, analysis, literature review, publication, proof of concept or any other (research) results.

²You may be prompted for your credentials in order to being proxied to the publishers’ website. In that list, the Scopus indexation tool may be particularly useful.

The concrete structure of this section may vary depending on your own project and its status (interim or final version), but the above details are expected such that any reader may understand **the solution** that has been foreseen or developed as well as **the reasons why** it has been thought or implemented that way. If the space limit reveals to be an issue, appropriate references to Appendices must be made (typically if more figures or diagrams have been produced, only a representative subset can be included here and the remainder is sent to the Appendices).

The level of depth and detail will vary between the interim and the final report where an interim report will mainly report on early findings, the future design and potentially on an embryo of a prototype.

2) *Method and Project Management*: All projects, being research-oriented or software developments, must follow an appropriate design method. Therefore, this section contains an explanation of the method followed during the project. For research-oriented aspects:

- a description of the research method must be given, potentially of the form of an instance of the design science cycle [1];
- a discussion of the (planned) evaluation process must be provided (e.g., survey, experimentation [2]).

For software development aspects:

- details on the followed process must be given (e.g., iterative [3], generic Agile [4], SCRUM [5], Kanban [6]);
- details on the tools and technologies used to support the software development process, e.g.,
 - track tasks, their completion and log the time spent on the project;
 - version the source code;
 - assess its quality (e.g., tests, metrics).
- details must be given on the processes put in place to get feedback from users and/or involved parties, e.g.,
 - how feedback has been collected
 - how often feedback has been collected
 - how feedback has been used to improve the research or solution

If external tools have been used, appropriate links must be given and evidence should be provided in the dedicated Appendix section. Research-oriented projects with a software development component must explain both management aspects (i.e. research design and development process).

F. Discussion and Evaluation (final report only)

Now the full details regarding both the objectives and results of the project have been presented, a discussion of these results must be provided. This evaluation may take the form of an informal product review, structured survey or general discussion highlighting the strengths and weaknesses of the project.

The discussion must be provided with sufficient evidence to support any claim made about the project, e.g.,

- report on the product review or any sort of feedback-gathering session;
- references to experimentation(s) conducted as part of the project (detailed in the “*Design and Implementation*” section);

A honest discussion usually contains a list of limitations. Therefore, it is expected to find an analysis of the shortcomings or unmet objectives of the project as well as a series of counter-measures or lessons learned that can be used by practitioners or researchers facing the same problems.

All conclusions and findings of the project must be sound and clearly addressed by the results of project.

G. Plan (optional for final report)

In this section, a clear and actionable plan describes the next milestones for the project. This plan usually takes the form of a table with milestones and clearly identified and assessable outputs in front of each milestone. This plan is used to assess the feasibility of the project for the remaining time of the project or to define longer term goals for continuous project spawning over many years.

The plan must be accompanied by a risk assessment where you discuss what potential difficulties may happen along the way (e.g., prevent data loss by frequent back-up). Critical milestones should also have a specific risk analysis to discuss what can prevent the milestone to be achieved on time.

If not already discussed in the method section (“*Project Management*”), the plan must include details on the future evaluation of the project outputs. This section is optional for the final report.

H. Conclusion and Future Work

The last expected section of your report is a summary where a quick wrap-up of the context, objectives and results is presented. As a second, more important part, a list of reasonable longer terms goals are listed if no plan has been given in the previous section. A conclusion is also the place for final thoughts regarding the project.

I. References

Your bibliography should not be filled with URLs or Wikipedia pages. Even in highly software development-focused projects, existing solutions or published literature articles make sense to a particular project. Furthermore, it is important that a project is conducted in a systematic and scientifically-grounded manner so that we expect you to have at least a minimal set of referenced scientific works.

Sufficient attention must be put into properly referring to the literature. As a reference, we expect you to conform to the Bibtex rules for all types of references (e.g., books, articles). The Wikipedia page of Bibtex³ describes concisely the expected fields for each types of reference.

³See <https://en.wikipedia.org/wiki/BibTeX>. Another web-based tool to manage Bibtex entries that may be of interest, see <https://truben.no/latex/bibtex/>.

J. Additional Material

We expect some additional material that will not be formally assessed, but that we expect you to supply in order to transfer the full picture to your supervisor. Failing to provide an Appendix if the nature of the project requires it may affect the grade. For example, if the project outputs contain a software product, we expect you to supply (or refer to) test data. Similarly, if the project is research-oriented and a survey has been conducted, we expect you to supply the raw (anonymised) data.

a) *Appendix A - Project Management Evidence:* Include all screenshots to support your software development or research method in terms of task and time management.

b) *Appendix B - User Manual:* Include an illustrated manual to deploy, use or maintain your software product.

c) *Appendix C - Test Data:* Include all manual test data used to ensure your software product is behaving correctly, if any.

d) *Other Appendices:* Anything else your supervisor(s) or partners would require from you.

III. RESOURCES

A. Templates

The IEEE templates can be found at <https://www.ieee.org/conferences/publishing/templates.html>. There are Word and Latex templates. We use the *conference proceedings* one. A minimum working example is given in Appendix B.

B. LaTeX

Latex documentation and help pages can be found at <https://www.latex-project.org/>.

C. Overleaf

Overleaf is a web-based editor that allows to edit Latex files without requiring to install anything locally. IEEE partnered with Overleaf and offer a link to a preconfigured project at <https://www.overleaf.com/latex/templates/ieee-conference-template-example/nsnscsyjfmpxy>. If your project is not highly confidential and classified, you may use Overleaf.

IV. CRASH COURSE, AKA LATEX 101

A. Latex document

A Latex document always starts with the definition of its class, potentially followed by additional packages. Note that comments inside a latex source are denoted by “%”.

```
1 \documentclass[conference]{IEEEtran} % preamble starts
2 \usepackage{listingsutf8}
3 \begin{document} % content starts
4   \title{My awesome project}
5   \author{Author definition}
6   % other expected attributes to create the title
7   \maketitle % that creates the title
8   % beginning of real content
9   \end{document} % end of file
```

The basic idea is comparable to the html language where tags are used to format the content of a document. In the following, we give a crash course on the basic commands.

B. Document Sections

Typical Latex documents (as this document class) use three levels of numbered sections: `\section{title}`, `\subsection{title}` and `\subsubsection{title}`. Additionally, `\paragraph{title}` can be created, but are unnumbered.

C. Typefont

You can use typefont commands to style your text. `\textbf{bold}` creates a **bold text**, `\textit{italic}` for *italic text*, `\texttt{typeset}` for typesetting text and `sans` for sans text.

D. Lists

There are three types of lists, enumerated lists, itemised lists and descriptive lists. You can create enumerated lists of items as follows:

```
1 \begin{enumerate}
2   \item numbered element
3 \end{enumerate}
```

You can create unnumbered lists of items as follows:

```
1 \begin{itemize}
2   \item itemised element
3 \end{itemize}
```

You can create descriptive lists of items as follows:

```
1 \begin{description}
2   \item[name] its description
3 \end{description}
```

E. Figures

There are two ways to include figures: inside one column or spreading over the full width. It is always better to store your images in a dedicated folder and use the relative path to include them in the latex file. You may play with different attributes to specify the size and the placement of your image:

size you can typically use the attributes `width`, `height` and `scale`. You can make them relative to the `\linewidth` or full `\textwidth`;

placement you can force ! the image to be placed here h, at the bottom b or top of a page t. The order is important, even if the template style file may ignore some (this template does not allow b);

To include an image, use the following code:

```
1 \begin{figure}[!ht]
2   \includegraphics[width=.95\linewidth]{pictures/
3     small}
4   \caption{Smaller picture.}
5   \label{fig:image-label}
6 \end{figure}
```

An example of such image is given in Figure 2.

For an image to spread over two columns, you add a `*` after the figure environment, as follows:

```
1 \begin{figure*}[!ht]
2   (...)
3 \end{figure*}
```

An example of a two-column image is given in Figure 1.



Fig. 1. Two column wide figure.



Fig. 2. Smaller picture.

F. Tables

Tables are useful to structure some details in a concise way. As for figures, you can define tables to be spreading on one or two columns. The basic code is as follows:

```

1 \begin{table}[!ht]
2   \caption{Description of your table (labelling).}
3   \label{tbl:table-label}
4   \centering
5   \begin{tabular}{column separators, e.g., r, l, c, p}
6     content of your table
7     "&" as a separator between columns
8     "\\" is used to create new lines
9     "\hline" is for horizontal borders between lines
10   \end{tabular}
11 \end{table}

```

The column can be specified using the following attributes:

- c** to centre the content of that column
- l** to align to the left the content of that column
- r** to align to the right the content of that column
- p** to create a proportional column passing a relative size, e.g., `p{0.2\textwidth}`
- | the “*pipe*” character is used to define borders between columns

As an example, Table I shows a simple one-column table corresponding the code given right after:

TABLE I
EXAMPLE OF A ONE-COLUMN TABLE

Proportional	Left	Centre	Right
A Big description	Yes	99	100.8
Another description	No	78	837834

```

1 \begin{table}[!ht]
2   \caption{Example of a one-column table}
3   \label{tbl:one-col}
4   \centering
5   \begin{tabular}{p{0.2\textwidth} c c c}
6     \textbf{Proportional} & \textbf{Left} & \textbf{Centre} & \textbf{Right} \\
7     & \textbf{Right} & \\
8     \hline \quad [-2ex]
9     A Big description & Yes & 99 & 100.8 \\
10    Another description & No & 78 & 837834 \\
11    \hline
12  \end{tabular}
13 \end{table}

```

A more complex two-column table is shown in Table II. It gives an example of how you can spread multiple columns on a particular row with the command `\multicolumn{number_of_column}{text}`. Its simplified code is as follows:

```

1 \begin{table*}[!ht]
2   \caption{A two-column table}
3   \label{tbl:two-col}
4   \centering
5   \begin{tabular}{p{0.01\textwidth} p{0.8\textwidth}
6     p{0.12\textwidth}}
7     & \textbf{column} & \textbf{column} \\
8     \hline \quad [-2ex] % this [-2ex] is to make the line
9       % break a bit smaller, just a hack to make the
10       table
11       % more readable when adding a line between rows
12     \multicolumn{3}{l}{Some overlapping text} \\
13     1. & This an important statement & Yes \\
14     2. & This an even more important statement & No \\
15     3. & We can also create horizontal border & Yeah \\
16     \hline
17     \multicolumn{2}{c}{2-row overlapping text} & Sweet \\
18     \\
19     4. & This is how tables are made. & $\star\star$ \\
20     5. & You'll never go back to Word ever after. \\
21     & \includegraphics[height=.8\baselineskip]{pictures/pic} \\
22   \end{tabular}

```


G. Code listing

You can include code listings with the `lstlisting` environment that accepts the following attributes:

language the name of the language, you may create your own with `\lstdefinelanguage` preamble commands;

label to make cross-references to the listing from the text (as for Figure or Table);

caption to describe the listing (as for Figure or Table).

A code listing is added with the following command (and, as figure or table, it may be suffixed by a `*` to spread on two columns):

```
\begin{lstlisting}[language=Java,
  label=lis:java,caption=Example of a listing]
// My awesome java code
public class MyClass {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
\end{lstlisting}
```

An example of such listing:

```
1 // My awesome java code
2 public class MyClass {
3   public static void main(String[] args) {
4     System.out.println("Hello World");
5   }
6 }
```

Listing 1. Example of a listing

You can further customise your code snippets by setting some options of the listing environment in your file preamble with `\lstset` as follows (configuration used in Listing 1):

```
1 \lstset{
2   basicstyle=\ttfamily\footnotesize, % font size
3   commentstyle=\itshape\color{teal}, % colour for
   comments
4   stringstyle=\color{blue}, % colour for strings
5   aboveskip=4pt, % space above the listing
6   belowskip=4pt, % space below the listing
7   basewidth={0.55em,0.45em},
8   keywordstyle=\bfseries\color{violet}, % keywords
9   identifierstyle=\color{black}, % language identifiers
10  numbers=left, % placement of line numbers
11  numberstyle=\tiny\color{gray}, % tuning of numbers
12  stepnumber=1, % frequency of numbers
13  numbersep=5pt, % separation of numbers
14  firstnumber=1, % first line number
15  frame=lines, % frame around listing
16  rulecolor=\color{gray}, % ruler colour
17  tabsize=2, % indentation size
18  breaklines=true, % allow line to be broken
19  showspaces=false, % don't show spaces
20  showstringspaces=false, % don't show spaces in strings
21  captionpos=b, % put caption below
22  literate={~} {\$sim\$}{1} % formatting of ~ char (for
   bash)
23 }
```

H. Citations, URLs and Cross-References

Citations can be made using `\cite{some-key}` and latex will automatically manage the bibliography for you if you follow these preliminary steps:

- you have a file where you store all you Bibtex entries;
- you specify the style of the bibliography;
- you summon the bibliography calculation.

For example, the reference to the book entitled "*Experimentation in Software Engineering*" by Wohlin *et al* referenced by [2] is defined by the following entry:

```
1 @BOOK {wohlin-12,
2   author   = "Claes Wohlin and Per Runeson and Marin H
   {\\"o}st
3   and Magnus C. Ohlsson and Bj{\\"o} Regnell
4   and Anders Wessl{\\"e}n",
5   title    = "Experimentation in Software Engineering
   ",
6   publisher = "Springer-Verlag",
7   year     = "2012",
8   isbn     = "978-3-642-29043-5"
9 }
```

Any further reference to that particular entry is done using the "*key*" defined in the Bibtex entry, so `\cite{wohlin-12}`.

You may keep your bibliography entries inside the same Latex file using the following trick in the preamble of your file:

```
1 \usepackage{filecontents}
2 \begin{filecontents}{my-bib-file.bib}
3   All my Bibtex entries
4 \end{filecontents}
```

Note that Bibtex has troubles with diacritics since it does not accept UTF8 encoded characters. If you include the content of the Bibtex inside your Latex source that support UTF8-encoded characters by setting the input encoding by `\usepackage[utf8]{inputenc}` in the preamble, you may use diacritics directly, (e.g., H \ddot{o} st instead of H{\\"o}st). To trigger the compilation of the bibliography, we use the following commands:

```
1 \bibliographystyle{IEEEtran}
2 \bibliography{inline}
```

Note the name of the artificial file `inline.bib` that will be used in the `\bibliography` command. You may pass any name (or path) to your external Bibtex file, if any.

Most of the publisher websites allow to export a bibliographic entry in Bibtex such that you mostly never need to write them by yourself.

URL may be included using `\url{some.link.url}` and will be automatically clickable.


Cross-referencing relies on two steps:

- defining a `\label{my:label}` in any numbered element (e.g., section, figure, table);
- referencing to that element using `\ref{}`.

In order to avoid a line break between the word and the reference, it is good practice to prefix `\cite` or `\ref` commands by `~` denoting a non-breaking space, as in:

```
1 (...) as shown in Figure~\ref{fig:teaser}
```

TABLE II
A TWO-COLUMN TABLE

Very Long Description	Small stuff
<i>Let's overlap over the three columns</i>	
1. This an important statement	Yes
2. This an even more important statement	No
3. We can also create horizontal border (see behind)	Yeah
<i>We can also overlap on two columns and centre the text</i>	
4. This is how tables are made in Latex with easy control on the format.	Sweet
5. You'll never go back to Word ever after (look at the figure inside table cell).	***** 

Last, footnotes can be declared with the `\footnote{text}` command.

evidence (e.g., “*Project Management*”, “*Test Data*”) are also supplied with the report.

REFERENCES

- [1] R. Wieringa, “Design science as nested problem solving,” in *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, ser. DESRIST '09. New York, NY, USA: ACM, 2009, pp. 8:1–8:12.
- [2] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Springer-Verlag, 2012.
- [3] B. W. Boehm, “A spiral model of software development and enhancement,” *Computer*, vol. 21, no. 5, pp. 61–72, May 1988.
- [4] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, “Manifesto for agile software development,” 2001, retrieved Aug. 27, 2019. [Online]. Available: <http://agilemanifesto.org/>
- [5] M. Beedle, M. Devos, Y. Sharon, K. Schwaber, and J. Sutherland, “Scrum: An extension pattern language for hyperproductive software development,” *Pattern Languages of Program Design*, vol. 4, pp. 637–651, 1999.
- [6] D. J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, apr 2010.

APPENDIX

A. Marking Rubrics

These marking rubrics will ensure the marking is consistent across all markers and transparent to the students.

1) *Plan and Objectives*: Refer to related columns in Table III, i.e. *Objectives*, *Plan* and *Quality of Writing*.

2) *Interim report*: Table III details the marking rubric for interim reports. All important criteria are listed in the first leftmost columns with their expected level of quality in regard to ranges of grades. A set of *decision rules* is given in the last line in order to explicitly refer to preponderant criteria in the final grade.

We advice students to take a look at this rubric before submitting their interim report in order to ensure all expected aspects of the report are covered with sufficient detail.

3) *Final report*: Table IV details the marking rubric for final reports in a similar fashion to Table III. As can be seen, more criteria apply to the final report and the expected level of quality is higher for some criteria in order to achieve a similar grade than the interim reports. The *decision rules* is also more demanding.

We advice students to take a look at this rubric before submitting their final report in order to ensure all expected aspects of the report are covered with sufficient detail. We also advice students to ensure the Appendices with the supporting

TABLE III
MARKING RUBRIC FOR PLAN AND INTERIM REPORTS.

Criteria	Fail	C- to C+	B- to B+	A- to A+
<i>Objectives</i>	The objectives are unclear or incomplete. No detail is given on the context of the project.	The objectives are unclear or incomplete. Little detail is given on the context of the project.	The objectives are clearly defined. Sufficient details are given on the context of the project.	The objectives are clearly defined, explained and justified. The context of the project is fully described.
<i>Related Work</i>	No discussion of related work is provided.	A basic discussion of related work is provided but lacks an analysis.	A discussion of related work is provided accompanied by an incomplete analysis in regards to the project.	A comprehensive discussion of related work is provided with a systematic analysis in regards to the project.
<i>Method</i>	No research or poor research method is applied and no or poor software development method is applied.	Poor research method is applied or poor software development method is applied.	A systematic research method or software development is applied and reported adequately.	A systematic research method or software development is evaluated in regards to the context of the project, applied and reported adequately.
<i>Design & Impl.</i>	No progress has been made and no solution or results are foreseen.	Insufficient progress has been made or results are presented poorly.	Sufficient progress has been made or foreseen solution or research is explained sufficiently.	Good progress has been made and foreseen solution or research is explained in detail.
<i>Plan</i>	No plan for the remaining of the project is given.	Poor plan and/or poor risk analysis for the remaining of the project is given.	A complete plan for the remaining of the project is given, but some milestones may not be fully described or assessable. A (potentially incomplete) risk analysis has been carried for critical milestones.	A complete plan for the remaining of the project is adequately reported with clearly identified milestones accompanied by clearly identified outputs. A systematic risk analysis has been carried for all critical milestones with potential countermeasures or adaptations to the project goal, if necessary.
<i>Quality of Writing</i>	The text is written with poor grammar and/or does not comply to the IEEE template and size constraints.	The text is written with adequate grammar, complies to the IEEE template and size constraints.	The text is written with adequate grammar, complies to the IEEE template, size constraints and is easy to follow for non-experts.	The text is written with adequate grammar, complies to the IEEE template, size constraints, is easy to follow for non domain experts and is pleasant to read.
<i>Decision rules</i>	A majority of above criteria apply.	At least <i>Objectives</i> , <i>Related Work</i> and <i>Quality of Writing</i> criteria apply to award a grade in this range.	At least <i>Objectives</i> , <i>Related Work</i> , <i>Plan</i> and <i>Quality of Writing</i> criteria apply to award a grade in this range.	At least <i>Objectives</i> , <i>Design & Implementation</i> , <i>Plan</i> and <i>Quality of Writing</i> criteria apply and remaining <i>Related Work</i> and <i>Method</i> criteria in the B-range apply to award a grade in this range.

TABLE IV
MARKING RUBRIC FOR FINAL REPORTS.

Criteria	Fail	C- to C+	B- to B+	A- to A+
<i>Objectives</i>	The objectives are unclear or incomplete. No detail is given on the context of the project.	The objectives are unclear or incomplete. Little detail is given on the context of the project.	The objectives are clearly defined. Sufficient details are given on the context of the project.	The objectives are clearly defined, explained and justified. The context of the project is fully described.
<i>Related Work</i>	No discussion of related work is provided.	A basic discussion of related work is provided but lacks an analysis.	A discussion of related work is provided accompanied by an incomplete analysis in regards to the project.	A comprehensive discussion of related work is provided with a systematic analysis in regards to the project.
<i>Method</i>	No research or poor research method is applied and no or poor software development method is applied.	Poor research method is applied or poor software development method is applied.	A systematic research method or software development is applied and reported adequately.	A systematic research method or software development is evaluated in regards to the context of the project, applied and reported adequately with supporting evidence in the Appendices.
<i>Design</i>	No design of the solution or research is reported.	The design of the solution or research is poorly reported and lacks details or justification.	The design of the solution or research is adequately reported with some rationale.	The design of the solution or research is adequately reported supported by a comprehensive justification.
<i>Implementation</i>	Insufficient or no progress has been made or results are presented poorly.	Insufficient progress has been made or results are presented poorly.	Sufficient progress has been made and solution or research is explained sufficiently.	Good progress has been made and solution or research is explained in detail with its rationale and supporting evidence in Appendices.
<i>Discussion</i>	No discussion of results of project is given.	Poor discussion of results of project is given.	A sufficient discussion of the results is given highlighting their strengths and weaknesses.	A comprehensive and constructive discussion of the results is given highlighting their strengths and weaknesses. A honest reflection on the method applied during the project is given.
<i>Quality of Writing</i>	The text is written with poor grammar and/or does not comply to the IEEE template and size constraints.	The text is written with adequate grammar, complies to the IEEE template and size constraints.	The text is written with adequate grammar, complies to the IEEE template, size constraints and is easy to follow for non-experts.	The text is written with adequate grammar, complies to the IEEE template, size constraints, is easy to follow for non domain experts and is pleasant to read.
Decision rules	A majority of above criteria apply.	At least <i>Objectives</i> , <i>Related Work</i> , <i>Method</i> , <i>Implementation</i> and <i>Quality of Writing</i> criteria apply to award a grade in this range.	At least <i>Objectives</i> , <i>Related Work</i> , <i>Method</i> , <i>Implementation</i> and <i>Quality of Writing</i> criteria apply to award a grade in this range.	At least <i>Objectives</i> , <i>Method</i> , <i>Design Discussion</i> and <i>Quality of Writing</i> criteria apply and remaining <i>Related Work</i> and <i>Implementation</i> criteria in the B-range apply to award a grade in this range.

B. Minimum Working Example

We provide a minimum working example you can use as a template for your report.

```
\documentclass[conference]{IEEEtran}
\IEEEoverridecommandlockouts

%% some useful packages
\usepackage[utf8]{inputenc}
\usepackage{cite}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{algorithmic}
\usepackage{graphicx}
\usepackage{textcomp}
\usepackage{xcolor}
\usepackage{url}
\usepackage{inconsolata}
\usepackage{enumitem}

%% bibtex hack needed for IEEE
\def\BibTeX{{\rm B\kern-.05em{\sc i}\kern-.025em b}\kern-.08em
T\kern-.1667em\lower.7ex\hbox{E}}\kern-.125emX}}

%% if you do not want to keep your bibfile separated,
%% you can input your bibitems here
\usepackage{filecontents}
\begin{filecontents}{inline.bib}
@misc{sample,
author = {Last, First},
title = {Some sample},
}
\end{filecontents}

%% start the document
\begin{document}

%% Your ptoject title
\title{My Awesome SENG402 Project}

\author{\IEEEauthorblockN{Your name}
\IEEEauthorblockA{\textit{Computer Science and Software Engineering} \\\
\textit{University of Canterbury}}\\\
Christchurch, New Zealand \\\
your.email@uclive.ac.nz}
}

\maketitle

%% we need to force the page numbering
\thispagestyle{plain}
\pagestyle{plain}

\begin{abstract}
What did you do?
\end{abstract}

\begin{IEEEkeywords}
some, comma, separated, keywords
\end{IEEEkeywords}

%% The real content starts here, we prepared a template outline
\section{Introduction}
\label{sec:intro}

\section{Background and Objectives}
\label{sec:background}

\section{Related Work}
\label{sec:rel-work}

\section{Solution}
\label{sec:solution}

\subsection{Design}
\label{sec:design}

\subsection{Implementation}
\label{sec:implem}

\subsection{Project Management (Development Process)}
\label{sec:process}

\section{Discussion, Evaluation and Limitations}
\label{sec:discussion}

\section{Plan}
\label{sec:plan}

\section{Conclusion}
\label{sec:concl}

%% The next two lines define the bibliography style to be used,
%% and the bibliography file.

\bibliographystyle{IEEEtran}
\bibliography{inline}

%% your appendices start here
\appendix

\section{Project Management Evidence}
\label{annex:proj-mgmt}

\section{User Manual}
\label{annex:manual}

\end{document}
```