

THE CSC3050 SOFTWARE ENGINEERING RUBRIC

COUSE

CSC 305 exposes the student to the issues and problems of creating a larger software system. The software engineering life cycle process acts as a framework guiding the student through this larger software maze. The student thinks more clearly and rigorously about the overall problem and its solution when forced to address each of the software life cycle stages in written form.

The six Stages of the System Development Life Cycle (CDLC)

1. System requirements and objectives.
2. Design of the software system.
3. Writing the program code.
4. Testing and debugging the program
5. Implementation of the program at the customer location.
6. Maintenance of the application to include updates and new versions. .

General Writing Guidelines for the Entire Document APA Formatting. (6 points)

All students submitting a software project document must include a cover page, a table of contents, abstract, body, and reference page. Each stage of the CDLC above, must be separated into its own individual section.

- Section headers and sub-section are written in bold. (2)
- Line spacing is double. (2)
- All tables and figures are labeled. Figures and tables have separate numbering system. The labels for figures appear below the actual figures while those for tables appear above the tables. (2)

In addition, documentation of the different stages should follow the guidelines below.

1. Requirement Specification. (4 points) Paragraph Format

The requirements of the software system to be developed and implemented are documented in a requirement specification document or section. The following needs to be stated clearly:

- The client organization for which the software system is to be developed. (1)
- Purpose of the software system and the intended users of the software system. (1)
- The details of each function to be performed by the system. (1)
- Any specific software requirements by the organization for which the system is being developed and any specific user-interface required by the organization. (1)

2. Project Management Plan. (2 points) Paragraph Format

In this section, a detailed plan and schedule is presented that articulates the time and manpower required for completing each stage of the software project.

- The resources are justified by giving details in the plan. For example, if 10 days and 5 persons are needed to implement the coding phase then the plan states for each person the approximate number of lines of code to be written by the individual, the section of the software system for which it is written, and the projected time to develop it. (2)
- Software tools such as Microsoft Project may be used for creating the project plan.
- The use of tables is recommended for writing this section.
- The project plan should be updated as needed because in practice, estimates may not be accurate all the time. Each updated plan is to be shown separately in this section.

3. Design Specification. (5 points) Paragraph Format

The following items are clearly described in this section:

- The design process used; e.g. Object-Oriented Design (OOD). (1)
- All identified system components and their functions. (1)
- All user-interfaces (diagrams are recommended). (1)
- Algorithms and pseudocode (if applicable) used in the detailed design. (1)
- Details of all the different software required by the system and the reasons for their selection. (1)

4. Program Code. (5 points) Syntax Format

For each program, part/whole of the code is copied and pasted in this section with a description of the environment and tools used and a brief description of the part of the system for which the code is written.

- Each program has a meaningful name. (1)
- At the beginning of the program, the programmer will list the date the program was written, its developer(s), and its functionality. (1)
- Each update to the program is documented below with its description by date. (1)

An example is given below:

```
Date          : MM/DD/YYYY.
Written by    : XXXXX.
Description: This program develops the main menu of an ATM for a bank. It is written in
              Java in the Linux environment. The tools used are ZZZZ.
MM/DD/YYYY   : A variable aaaa was updated to bbbb.
```

- Variable and module names are declared and documented at the beginning. (2)
- Separate modules appear after the main module in the program.

5. Testing/Debugging Specification. (8 points) Tabular Format

A test plan must be presented, preferably in tabular form. This will contain:

- The system components to be tested. (2)

- The functionalities in each component to be tested. (2)
- The test cases for each programming method or algorithm. (2)
- The results of each test are documented, and it must be mentioned if it matches the expected outcome; if not, the reasons for the incorrect response must be documented as well. Debugging methods must be documented including tools used for the debugging process. (2)

6. Implementation Guide. (3 points) Paragraph Format

In this section, the implementation details are to be mentioned.

- Details of the implementation date and environment are essential. (1)
- One of the most important things to be detailed is the place where the software system will go into production. It may be same or different from the test site. (1)
- Any problems found during implementation, debugging measures and initial reaction of the users must be documented as well. (1)

7 Maintenance Guide. (5 points) Paragraph and Tabular Format

This section contains:

- Details concerning each unit of software and the required maintenance. (1)
- The time intervals during which maintenance is to be carried out. (1)
- Tools and any specialized knowledge required to perform maintenance. (1)
- If a unit fails, the corrective or debugging actions (if any) that can be taken must also be documented. (2) You can also implement a tabular form is needed.

8. User Guide. (2 points) Paragraph and Bullet Format

The user guide should specify

- Intended users of the system and all functionalities intended for the users.(1)
- Each user interface menu and its usage method. (1)