

# COS 497: Capstone II

---

## Code Inspection Report Template

(Adapted from Susan Mitchell)

### General Instructions

1. Provide a cover page that includes the document name, product name, customer name, team name, team member names, and the current date. You may include your team logo.
2. Number the pages of the document.
3. Number and label all figures. Refer to the figures by number in the text.
4. All sections should have an introductory sentence or two.
5. Do not use vague words and phrases such as may, might, could, possibly, assumed to be, some, a little, and a lot. Use strong, definite words and phrases such as shall, will, will not, can, and cannot. Words such as “should” can be used to show suggestions, but use it sparingly.
6. Watch your spelling, punctuation, and grammar. It is a reflection on your professionalism.

Be sure that your document is

- Complete - No information is missing
- Clear - Every sentence's meaning must be clear to all parties
- Consistent – The writing style and notation is consistent throughout the document and the document does not contradict itself
- Verifiable - All requirements and other facts stated are verifiable

Remember that you are required to do a team review of this document.

## **Purpose of This Assignment**

- To achieve group consensus on developing or adopting practical coding and commenting conventions.
- To learn how to perform code inspections among "egoless" team members.
- To appreciate how error correction costs can be significantly reduced by discovery during the coding phase rather than later during testing or after installation.

## **Required Activities**

- Define project-specific commenting and coding conventions (or adopt/modify other popular conventions).
- Compile and build all code without fatal errors and with only minimal warnings.
- Inspect together, as a full team, all significant pieces of code written to date. (Code inspection meetings may be virtual rather than physical, although this may be more difficult.) You will need to take copious notes during inspection meetings in order to fulfill the content requirements of the Code Inspection Report. Understand what information you will need to record before each meeting.
- The CIR should document that a thorough Code Inspection has been completed by the team. Inspection reveals flaws and inconsistencies, but it does not necessarily fix them.

*Product Name*  
Code Inspection Report

**Table of Contents**

	<u>Page</u>
1. Introduction.....	4
1.1 Purpose of This Document.....	4
References.....	4
1.2 Coding and Commenting Conventions.....	4
1.3 Defect Checklist.....	4
2. Code Inspection Process .....	5
2.1 Description.....	5
2.2 Impressions of the Process.....	5
2.3 Inspection Meetings.....	5
3. Modules Inspected .....	6
4. Defects .....	6
Appendix A - Coding and Commenting Conventions.....	7
Appendix B – Team Review Sign-off .....	7
Appendix C – Document Contributions .....	7

# 1. Introduction

## 1.1 Purpose of This Document

State the purpose of the document and specify the intended readership. Briefly summarize the content. [one paragraph]

## References

Provide a list of all applicable and referenced documents and other media used during code inspection or in the preparation of this document. Minimally, references to the SRS and the SDD go here. See the Writing Resources on Blackboard for the appropriate formats for references.

## 1.2 Coding and Commenting Conventions

Summarize your coding and commenting conventions. Explain the rationale underlying their development (if original to your team) and/or adoption (if adopted from elsewhere).

- If you adopted the conventions fully, give the complete information for the source.
- If you adopted any conventions, but adjusted them, give the complete information for the source. Elaborate on the adjustments and provide examples.
- If you created your own conventions, use Appendix A to elaborate on the conventions and provide examples for each. Refer the reader to the appendix.

[minimum of one to two substantial paragraphs]

## 1.3 Defect Checklist

Give a comprehensive tabular checklist of possible defects that you used during the inspection process. Create categories for the defects such as Coding Conventions, Logic Errors, Security Oversights, and Commenting (you may create your own categories). Remember to take into consideration defects that are programming language-specific. (I expect to see at least **15** possible defects.)

## 2. Code Inspection Process

### 2.1 Description

Briefly describe the code inspection process that your team followed. Give the process *really* followed, not what you perceive to be the "ideal" process. If your process was relatively far from the "ideal," explain why you chose to so diverge as well as how.

- Credit will not be taken off for following a non-standard process if that process seems likely to have led to your reported results.
- Credit will be taken off for describing a process that does not seem to match your reported results.

[Two to three substantial paragraphs]

### 2.2 Impressions of the Process

Summarize your general impressions of the code inspection process (e.g., was it effective, why/why not, what you might do differently the next time). Your impressions may be positive, negative, or a combination. I am interested in what you really thought, not what you think I want to hear.

Additionally, indicate

- which are the "best" one or two modular units in your program (in terms of relatively small likelihood of remaining flaws) and why you think so, and
- which are the "worst" one or two modular units in your program (in terms of relatively large likelihood of remaining flaws) and why you think so.

[two to three substantial paragraphs]

### 2.3 Inspection Meetings

For each code inspection meeting that was held, specify the date, location (virtual as well as physical), time started, time ended, who participated, who performed which roles (e.g., moderator, scribe), and which particular code units were covered in the meeting.

### 3. Modules Inspected

Note: It is possible that you have not completed all coding at this point. However, I expect that at least 75% of the code has been written and inspected. Provide a list of the modules that have not been completed (if any), giving their names and brief descriptions of their functionalities. Provide projected dates by which you expect to complete each.

Present the following for every source code module that was inspected.

- Give its name, a description of its functionality, and a description of where it fits into the design and architecture in the SDD. Normally, it should implement a modular unit that comprises all or part of a design component or class (from the SDD). If the module did not appear in the SDD, explain why it was added.
- If there are any differences between the modular unit's design (from the SDD) and the actual code, then
  - Compare the planned module interface to the actual interface implemented.
  - Compare the planned data structures to the actual data structures implemented.
  - Compare the planned user and file/database I/O to the actual I/O implemented.
  - Compare the planned algorithm(s) to the actual algorithm(s) implemented.
  - Consider, with respect to all of the above, tradeoffs (e.g., worst case time vs. space, ease of implementation and testing, and opportunities for design/code reuse).

### 4. Defects

Reports indicating zero or very few defects are unlikely to be believable. Think of this as a "treasure hunt" and find as many defects as possible. But please do not insert defects intentionally.

Present the following for each defect found, including coding and commenting convention noncompliance. A tabular format is strongly suggested.

- Name of the module in which the defect was found
- Description of the defect, making clear in what way it is deficient. Categorize it in one of the following defect areas:
  - *Correctness* with respect to the functional specification (does it look like it does what is required?)

- *Coding Convention* violation
- *Commenting* convention violation, missing or incorrect comments
- *User friendliness* (consider the handling of user input errors as well as prompts and all other information displayed to the user)
- *Other* (e.g., minimizing inter-module coupling, maximizing intra-module cohesion, refactoring)

You may add other categories if you wish.

## **Appendix A - Coding and Commenting Conventions**

See Section 1.3 above. It is possible that you may not need this appendix. If not, re-letter the remaining two appendices.

## **Appendix B – Team Review Sign-off**

Place on a separate page. Provide a brief paragraph stating that all members of the team have reviewed the document and agree on its content and format. Provide lines for typed names, signatures, dates, and comments for each team member. The comment areas are to be used to state any minor points regarding the document that members may not agree with. Note that there cannot be any major points of contention.

## **Appendix C – Document Contributions**

Identify how each member contributed to the creation of this document. Include what sections each member worked on and an estimate of the percentage of work they contributed. Remember that each team member must contribute to the writing (includes diagrams) for each document produced.