COMP1682 – Final Year project
Project Report - Feedback sheet

Student Name and ID: Tristan Read 001151378
Supervisor Name: John Ewer

Overall assessment

The project concerns the development of a configurable source code enhancer for C#.

The project write up is excellent, though the report would benefit from a stronger narrative and figures to show how decisions were evidenced and how the software design was formulated.

There are also weaknesses in the project planning, e.g., the Gantt chart is excellent and very detailed, but does not indicate how task estimation was evidenced and how the planning of tasks (i.e., a WBS) related to the approach and decisions taken in the project.

The use of SMART objectives would have facilitated QA evaluation of the overall project success.

Overall, an excellent piece of work, that would have scored higher if it had evidenced and included additional software engineering skills from across the whole degree program (e.g., LSEP issues, Project Planning, etc).

Comments on the viva/demonstration

Have you held a demonstration or viva for this project? YES

The student gave an excellent account and confident presentation of the project, leading to a demonstration of the product used both as an integrated line-by-line code improver in the visual IDE or as a batch file processing utility.

There were a couple of minor issues in the nature of the product and its operation (e.g., checking changed names against the existing namespace to avoid conflicts), but the whole body of work was excellent and could serve as a platform for a number of code maintenance, improvement and, potentially, translation strategies.

The student answered all questions posed and is clearly a very skilled and capable software developer.

Comments on the Standard of Presentation

Has the report met presentation criteria? YES

The project title is very brief and could give a more detailed description of the technologies involved in the approach.

The Abstract is a very good, but rather brief, overview summary of the project.

The introduction makes an excellent case for the project, but would have benefitted from a more in-depth discussion of the nature and impact (ideally with example code fragments and citations of stats from industry) of the issues that have been raised.

The literature review, whilst detailed and well evidenced, could do a little more to highlight how the lit. review findings set the strategic course of the project and the decisions that were made.

Discussion of the project methodology is excellent.

The project development section is excellent. A sound and reasoned approach was taken for the software development. The discussion of design decisions and reasoning is detailed but would benefit from the use of figures to facilitate understanding of the points being made (or design overview) and with suitable code fragment examples.

Try to avoid the use of 'we' in formal/professional reports to keep the tense impersonal.

Testing was comprehensively and appropriately covered. The work is excellent.

The Epilogue could be structured with more clarity. Without being too prescriptive about sub-sections, the project report should summarise the project, describe what was achieved, critically reflect on the product, the project methodology and the learning outcomes, and discuss the scope for further work. These were somewhat merged in the final report.

The References are appropriate and correctly formatted.

The standard of English and grammar are excellent throughout the clearly written report.

Project planning is somewhat covered in the appendices, though there could be more detail about QA and WBS in support of the methodology.

There was no consideration of LSEP issues and how these were managed through the project.

Authenticity of the Submission

All final reports are uploaded into TurnItIn for plagiarism checking.

TurnItIn Percentage Match = 4%

Comments on the TurnItIn report, and any other plagiarism issues.

Minor matches. No issues of concern with the report.

The marks entered here should roughly average to the overall grade awarded:

---

1. **Understanding of the Problem Domain**: Demonstration: 80% | Report: 80%
2. **Development of Product and Ideas**: Demonstration: 75% | Report: 80%
3. **Product Build and Evaluation**: Demonstration: 80% | Report: 75%
4. **Conclusions and Critical Review**: Demonstration: 70% | Report: 75%

COMP1682 – Final Year project
Project Report - Feedback sheet

Student Name and ID: Tristan Percy Read 001151378
Second Marker Name: Cornelia Boldyreff

Overall assessment

This project tackles a key concern in Software Engineering: how to enhance software quality where development is undertaken by a team or individual over time.

Comments on the viva/demonstration

Have you held a demonstration or viva for this project? YES

The student gave an excellent overview of the rationale for the product he developed. He discussed in detail the design and testing major components and the integration. He demonstrated individual components running on single files and in batch mode.

Throughout he was able to confidently explain his work and answer our questions.

Comments on the Standard of Presentation

Has the report met presentation criteria? YES

The report is well written and suitably structured.

Authenticity of the Submission

All final reports are uploaded into TurnItIn for plagiarism checking.
TurnItIn Percentage Match = _4_%
Comments on the TurnItIn report, and any other plagiarism issues.

I have no concerns regarding the authenticity of this report.

The marks entered here should roughly average to the overall grade awarded:

5. **Understanding of the Problem Domain**: Demonstration: 80% | Report: 75%
6. **Development of Product and Ideas**: Demonstration: 75% | Report: 75%
7. **Product Build and Evaluation**: Demonstration: 75% | Report: 75%
8. **Conclusions and Critical Review**: Demonstration: 75% | Report: 70%

There is well researched account of the problem domain and nature of the problem and its

relevance to Software Engineering. An approach to development, the spiral model with continuous integration, is well justified. Consideration was given to underlying components, potential and those needing to be developed. e.g. regex.

Prior to development, consideration was given to evaluation.

Object oriented principles and design practices have been taken into account during the design stage and GitHub has been employed for project management of the code and YAML was used for configuration management. Problems encountered during implementation are described and solutions adopted are indicated.

The product was comprehensively tested using unit tests as well as integration testing and as a standalone application.

The performance of the tool was evaluated. Limitations are noted and potential enhancements outlined.