**Procedure CDP02 – Software engineering** *(version 2.0)*

# Choosing the relevant technologies (10 points)

You should write a short report (max 2 sides of A4) justifying the technologies you have decided to use to implement the customer requirements, including:

* An account of all the technologies available and suitable for the job (e.g. different programming languages, platforms, DBMS etc.)
* A justification of the technologies you have chosen
* The risks/limitations, if any, associated with the chosen technologies

## What to submit

A written report (e.g. Word document)

## Scale

The following scale should be used as a guideline when evaluating this objective:

|  |  |
| --- | --- |
| 10 points | A detailed account of the points listed above, showing a good understanding of the subject matter.  **References have been used** to underpin the work. |
| 8 points | A detailed account of the points listed above, showing a good understanding of the subject matter.  **References have not been used** to underpin the work. |
| 5 points | A partial account of the points listed above, but not all aspects are covered.  A good understanding is shown. |
| 2 points | A partial account of the points listed above, with very little understanding shown. |
| 0 point | No work submitted towards this objective |

# Implementing functional requirements (10 points)

You should implement all functional requirements as per agreed specification, i.e. **every item on the current sprint’s Backlog, as agreed with project manager.**

## Use of Version Control

You should commit your code regularly to a **Version Control system**.

The preferred option is the University GitLab server, located here:

* <https://fsegitlab.wlv.ac.uk/>

Please note that many other services offer student tariff (often free!), for example:

* <https://education.github.com/>
* <https://bitbucket.org/plans>

Version Control systems are a great place to **collaborate**. Developers can share code, comments, create issues/bug entries, etc. Make sure that you use these features where applicable for a higher grade.

## Commenting your code

Collaboration amongst developers is also greatly helped by the use of comments that explain certain key sections of your code. You should always comment the code written as part of this assessment.

## What to submit

* Source code listings, highlighting sections that **you** have produced
* GitLab commit logs showing **your** commits
* GitLab commit graphs showing **your** commits

## Scale

The following scale should be used to evaluate the points above:

|  |  |  |
| --- | --- | --- |
| Implementation of functional requirements  *7 points* | **All** requirements have been implemented and are fully functional. Code is fully commented. | 7 points |
| **Most** requirements have been implemented and are fully functional – some minor omissions or bugs | 5 points |
| **Very few** of the requirements have been implemented, and/or a lot of **errors** and **bugs** | 2 points |
| **No** implementation | 0 point |
| Use of Version Control  *3 points* | **All code was committed** to a version control system on a regular basis | 3 points |
| **All code was committed** to a Version Control system, but intermittently. | 2 points |
| Partial use of a Version Control System (not all code submitted, intermittent use) | 1 point |
| **No Version Control system** used | 0 point |

# Bug fixing (10 points)

You should work with the Business Analyst and rest of the team to fix as many bugs as possible.

## What to submit

* Source code extracts showing specific amendments/fixes to your code.
* Screenshots showing contributions to GitLab “Issues”, comments, fixes etc.

## Scale

The following scale should be used to evaluate this objective:

|  |  |  |
| --- | --- | --- |
| Code  *7 points* | **All** issues allocated to you have been resolved | 7 points |
| **Most** issues allocated to you have been resolved | 5 points |
| **Very few** of the issues allocated to you have been resolved | 2 points |
| **No** work | 0 point |
| Use of Version Control issues  *3 points* | **Full use** of Version Control issues (allocation of issues, closing issues when resolved, comments detailing the fix etc.) | 3 points |
| **Partial use** of Version Control issues. | 1 point |
| **No use** of Version Control issues. | 0 point |