

Appendix 4: evaluation (SDD)

This is an objective review of the software to establish whether it meets the required criteria.

Fitness for purpose

This reflects whether the software carries out all the tasks required of the software specification.

Your evaluation should identify any discrepancies between the software specification and the completed software.

Efficient use of coding constructs

This reflects whether the software writers have used their knowledge of constructs to help them create efficient code. For example using:

- ◆ suitable data types or structures
- ◆ conditional or fixed loops
- ◆ arrays
- ◆ nested selection
- ◆ procedures or functions with parameter passing

Your evaluation should identify where your coding has been efficient.

Usability

This reflects how intuitive the software is from a user's perspective and should include:

- ◆ the general user interface
- ◆ the user prompts
- ◆ the screen layout
- ◆ any help screens

Your evaluation should identify features of the software that have enhanced usability for the user.

Maintainability

This reflects how easy it is to alter the software. The factors affecting maintainability include:

- ◆ readability of the code — made easier by using meaningful variable names, comments, indentation and whitespace
- ◆ amount of modularity — using functions and procedures effectively

Your evaluation should identify how your code helps with the maintainability of the software.

Robustness

This reflects how well the software copes with errors during execution including:

- ◆ exceptional data, eg the computer crashing if “out of range”
- ◆ incorrect data entered

Your evaluation should reflect the testing that has been undertaken to meet the specification, as well as to demonstrate some degree of robustness.