

4 – Testing And Evaluation

NEA

Testing

Test Number	Test	Data	Expected Outcome
1	Entering level and pressing 'enter'	2	Error message: "Please enter a whole number between 3 and 10"

The 'expected outcome' column contains your description of what was supposed to happen as a result of this test. The 'actual outcome' column should contain what *really* happened. The test will either pass or fail.

- **If the test passes**, and the 'expected outcome' column matches what actually happened, you need only write 'success' in the 'actual outcome' column.
- **If the test fails**, and the expected and actual outcomes differ from one another, you need to describe what actually happened. Did the program display an unexpected error? Which one? Did it crash? With which error? Did a calculation produce the wrong result? Which result was produced? Failed tests need more explanation than passed tests, because there are many ways in which a test can fail.
- **If the test passes but something unexpected also happens**, this should be noted. Your 'actual outcome' entry will begin 'Success, however...'

Whether the test passes or fails, you need to provide evidence. Use screen recordings.

Testing Checklist

Testing

- | | | |
|---|--|---|
| <input type="checkbox"/> The test plan has been partially completed, missing key requirements testing | <input type="checkbox"/> The test plan has been mostly completed, with key requirements tested | <input type="checkbox"/> The test plan has been fully completed, with all requirements tested |
| <input type="checkbox"/> The solution meets a few of the success criteria | <input type="checkbox"/> The solution meets most of the success criteria | <input type="checkbox"/> The solution meets all of the success criteria |
| <input type="checkbox"/> Testing for robustness has not been completed | <input type="checkbox"/> Some testing for robustness is evidenced | <input type="checkbox"/> Testing for robustness is fully evidenced |

Evaluation

A good evaluation is a fairly complex piece of work, requiring a combination of skills.


It goes beyond simply describing something that you have done and how well it went.

In evaluating anything, it needs to be broken into pieces, and each piece is subject to a mini-evaluation.



Evaluation

Each of these 'mini-evaluations' follows the same basic pattern:

1. State
 2. Describe
 3. Explain
 4. Evaluate
 5. Suggest improvements
- 

State

What is the success criterion you are addressing?

My program was required to allow for any type of invalid data, including out of range (3-10 is the range), wrong data type (integer is required) or nothing being entered at all.

Describe

Talk about what your program does well and what it does badly.

Restrict yourself to the success criteria being addressed (other criteria will belong to their own mini-evaluations). Ideally, there will be positives and negatives to discuss, even if either is fairly insignificant.

In terms of managing out-of-range values, the program seems to work perfectly. Any numbers below 3 or above 10 are rejected with a helpful error message, and the user is allowed to re-enter a number. However, decimal values are accepted (5.7 would be accepted and truncated to a value of '5'), and text and blank entries cause the program to crash.

Explain

In this context, explain means 'why'. Why do the shortcomings exist? As there won't always be shortcomings, this part might not play a part in every mini-evaluation. While explanations that involve not having the ability or the time are valid, try not to overuse them.

A value of 5.7 would be accepted by the program because a range check, rather than a type check, is being used, and 5.7 is between 3 and 10. There is no code to deal with an entry that consists of text or nothing at all.

Evaluate

In an evaluation, you need to consider the value or importance of what you are evaluating.

Three questions need to be addressed here:

- How important was it for this criterion to be met? Consider the point of view of the end user (in this case, the researcher).
- How significant are the positives for this criterion? Parts of this success criterion have been met; how big a deal are these parts?
- How significant are the negatives? Similarly, are the shortcomings surrounding this success criterion fairly unimportant, or would they amount to the program being a complete failure?

Validation is an important part of this program. If the program crashed after a piece of invalid data, the test subject might need to ask for help, or they might not take part in the research at all. The helpful error message in the case of an out-of-range entry is helpful, and it would allow users to resolve the problem themselves. It is unlikely that anyone would enter a decimal value, but text or blank entries are a problem, as people might accidentally press enter, or enter a number and a space.

Suggest Improvements

How would you overcome the shortcomings? If there are no shortcomings, how might you make this aspect of the program even better? The more specific and technical you can be, the better. Given the number of mini-evaluations you'll be writing, you would do better to write one specific improvement than several vague ones.

The problems could be fixed using two additional validation checks. VB allows for checks that a variable contains a number, and a string can be checked to see whether it contains a decimal point. If either of these validation rules are breached, the loop could run again, as it does with out-of-range values.

Testing Checklist

Evaluation and Conclusions

- | | | |
|--|--|---|
| <input type="checkbox"/> Unresolved issues are ignored or not addressed | <input type="checkbox"/> Unresolved issues are commented on, but resolution to these are not discussed | <input type="checkbox"/> Unresolved issues and their resolutions are fully justified |
| <input type="checkbox"/> There is little or no evidence that systematic testing is used to refine the solution | <input type="checkbox"/> There is some evidence that systematic testing is used to refine the solution | <input type="checkbox"/> There is significant evidence that systematic testing is used to refine the solution |
| <input type="checkbox"/> The report is lacking coherence | <input type="checkbox"/> The report is generally coherent | <input type="checkbox"/> The report is coherent |