Software Development Life Cycle (SDLC) options

I have chosen to use an iterative waterfall life cycle, I have given my reasoning below.

The four main parts of a life cycle are analysis, design, implementation and evaluation.

***Classic* Waterfall**

A classic waterfall is a life cycle which follows an order and does not revisit previous parts of the life cycle. It follows analysis -> design -> implementation -> design

Benefits

The main benefit of the waterfall method is that it is harder to deviate off track because it is a flow through one cycle. You do each task in a linear fashion, i.e. you do the analysis, then the design, then implement your design and evaluate at the end.

Disadvantages

The big disadvantage is also because of the linear fashion of the life cycle, you do not revisit any from before. If there is an oversight which is discovered at the end, using this life cycle, you would not revisit and improve.

Potential usefulness in project

This would be useful for keeping me on track since I have a set deadline which is the length of the module, however, by not allowing iteration this would not work for my project.

Accept/reject

Reject

***Iterative* Waterfall**

An iterative waterfall life cycle is like the classic waterfall but iterates over and over each part.

Benefits

The biggest benefit of an iterative waterfall is that each of the four main parts are revisited over and over and can be revisited from any part, e.g. when you are at the evaluation stage of the project, you can return to the analysis, beginning, stage.

Disadvantages

A disadvantage is that this can mean you might keep revisiting earlier stages of the life cycle and not finish in the timeline given.

Potential usefulness in project

This would be useful for my project if I manage my time using a schedule to ensure that I don’t stray away from the target.

Accept/reject

Accept

**Agile**

An agile life cycle sets out work for a set period, usually two weeks, called a sprint. The progress made from each sprint is then factored in the following sprint.

Benefits

With agile development, you may have long term goals, but you are mainly thinking in terms of every two weeks. You set yourself goals for each two-week period. This is very good for keeping you on track short term as you’re not allowed to deviate from the goals you’ve set yourself for that two weeks. If you surpass your goals, you may have work to do from the backlog.

Disadvantages

A disadvantage for this project as it is very for collaborative work because you will all collaborate on how things have gone and review each sprint. It may be difficult for me to think in my mind after every two weeks what went well and what didn’t.

Potential usefulness in project

This could be useful for my project, but I don’t think I would get all the benefits of an agile life cycle given that I’m working on my own. I also have a set deadline so working in sets of two weeks may put me behind and I wouldn’t realise until too late.

Accept/reject

Reject