Criteria 1 and 2 submission

Tuesday, 4 April 2023 9:33 AM

Criteria 2

The waterfall development model orders tasks in such a way that each task needs to be fully completed before the next stage can begin. Once a stage has been completed, one cannot go back and do it again. This development model has just one developer working on one section of the project plan.

The waterfall development model splits development into 5 stages, analysis, design, implementation, testing, and maintenance.
The analysis phase includes specifying the software requirements and specifications, as the waterfall development model requires the requirements to be fully established before further stages in the development can begin.
In the design phase designs are created of the program, which can include designs of the software solution, algorithm design (i.e. pseudocode), mock ups, object descriptions, and data dictionaries.
Following this is the implementation stage, where the coding of the program commences. Using the work that was previously completed in the two stages beforehand, construction of the program begins, creating the code and the files the solution will use.
After this is the testing phase, making sure the solution meets the requirements and specifications using techniques such as a testing table.
Finally the maintenance phase is where the developer makes amendments to the program in accordance to issues or feedback received during the previous stages in order to improve performance of the solution and ensure that it was made in accordance to the requirements laid out by the client.

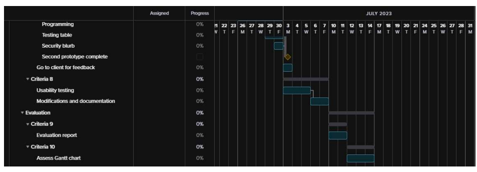
Compared to an Agile and Spiral development model, Waterfall's linear and simple nature makes it easy to use and understand, thus taking less time to plan out the development. Furthermore, this particular model is easier to manage as it uses rigid and defined phases, unlike Agile where the phases may be under constant change and thus more difficult to manage. With the Waterfall model, all phases are documented greatly and the progress of each phase can be easily seen, whereas the other two development models have it more ambiguous. Due to this, it is unlikely that the project will produce unexpected financial expenses. Moreover, testing the solution and the plan is simple and the end of the project is clearly defined.

That said, the Waterfall software development life cycle comes with disadvantages. Although its rigid nature is a strength, it is also a weakness as it also means that it cannot be easily altered. In the case of a major error popping up or the requirements need changing, it is difficult to make changes in the development cycle. Additionally, this model presents a long process with few shortcuts, thus eating into the economic cost of time (hence the developers need to have a great deal of time on their hands). Delays faced in earlier phases or even changes causes massive delays to the entire project, meaning that extra care has to be taken to avoid such circumstances. On top of that, the Waterfall development model only allows for the software solution to be available at the end of the project, meaning that during development the developers cannot receive feedback on what they are creating. Thus, there is little opportunity to identify errors and amend them.

On the other hand, the Agile development model works on some of the shortcomings of the waterfall model. It instead favours flexibility, communication, collaboration, and simplicity. The Agile development model revolves around iterations, going through cycles of development called sprints. At the end of each sprint, the software is released, feedback is taken, and improvements are made during the next sprint. As such some of the advantages of this model include reducing risks, the product is released early and multiple times, and the greater interactions with the client allows the developers to create a solution that will more likely please the client. However, some disadvantages it has is that it requires highly skilled clients and the timeline of the project may exceed the initial prediction.

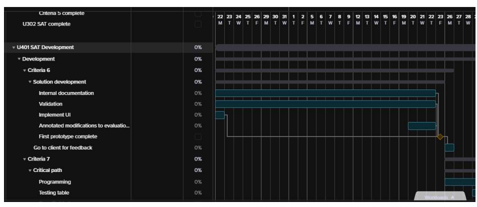
Considering both the advantages and disadvantages, I have chosen to implement a Waterfall model with a small amount of elements taken from the Agile model to improve it. The elements I have taken from the Agile model are creating prototypes to give to the client for feedback and having time to make amendments to the program in accordance to the feedback received from the client. The Waterfall model fits my project as the specifications are well known and it is small in scope (only one program that achieves a relatively small function). Also, as there is only one developer for the project, the Waterfall project suits my needs as I will already be sequencing my work through the completion of one task individually and the model fits the relatively small time frame given for the project. The other two models are not as appropriate as I do not hold the higher skills and resources that are demanded by the other two development models, hence making the Waterfall model (as a slight hybrid) the best choice. By implementing elements from the Agile model in, I have minimised some of the disadvantages of the Waterfall model, namely the software only being available at the end and having more opportunities to fix errors.

Criteria 1

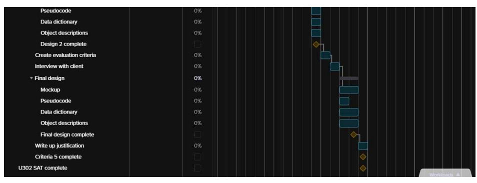


	Assigned	Progress
Programming		0%
Testing table		0%
Security blurb		0%
Second prototype complete		☐
Go to client for feedback		0%
▾ Criteria 8		0%
Usability testing		0%
Modifications and documentation		0%
▾ Evaluation		0%
▾ Criteria 9		0%
Evaluation report		0%
▾ Criteria 10		0%
Assess Gantt chart		0%

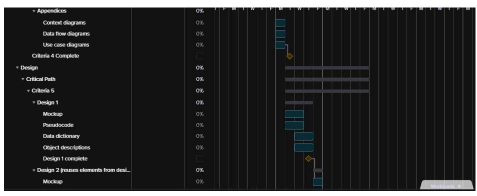
JULY 2023



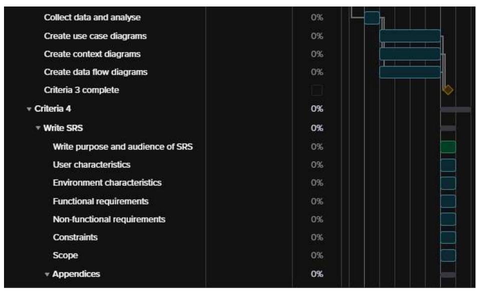
Criteria 5 complete		☐
U302 SAT complete		☐
▾ U401 SAT Development		0%
▾ Development		0%
▾ Criteria 6		0%
▾ Solution development		0%
Internal documentation		0%
Validation		0%
Implement UI		0%
Annotated modifications to evaluatio...		0%
First prototype complete		☐
Go to client for feedback		0%
▾ Criteria 7		0%
▾ Critical path		0%
Programming		0%
Testing table		0%



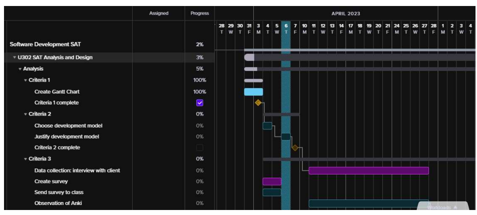
Pseudocode		0%
Data dictionary		0%
Object descriptions		0%
Design 2 complete		☐
Create evaluation criteria		0%
Interview with client		0%
▾ Final design		0%
Mockup		0%
Pseudocode		0%
Data dictionary		0%
Object descriptions		0%
Final design complete		☐
Write up justification		0%
Criteria 5 complete		☐
U302 SAT complete		☐



▾ Appendices		0%
Context diagrams		0%
Data flow diagrams		0%
Use case diagrams		0%
Criteria 4 Complete		☐
▾ Design		0%
▾ Critical Path		0%
▾ Criteria 5		0%
▾ Design 1		0%
Mockup		0%
Pseudocode		0%
Data dictionary		0%
Object descriptions		0%
Design 1 complete		☐
▾ Design 2 (reuses elements from desi...		0%
Mockup		0%



Collect data and analyse		0%
Create use case diagrams		0%
Create context diagrams		0%
Create data flow diagrams		0%
Criteria 3 complete		☐
▾ Criteria 4		0%
▾ Write SRS		0%
Write purpose and audience of SRS		0%
User characteristics		0%
Environment characteristics		0%
Functional requirements		0%
Non-functional requirements		0%
Constraints		0%
Scope		0%
▾ Appendices		0%



	Assigned	Progress
Software Development SAT		2%
▾ U302 SAT Analysis and Design		3%
▾ Analysis		5%
▾ Criteria 1		100%
Create Gantt Chart		100%
Criteria 1 complete		✓
▾ Criteria 2		0%
Choose development model		0%
Justify development model		0%
Criteria 2 complete		☐
▾ Criteria 3		0%
Data collection: interview with client		0%
Create survey		0%
Send survey to class		0%
Observation of Anki		0%

APRIL 2023