Development Model Justification

Prior to commencing the development of my software solution product for my client I have analysed and compared three popular development models that I could implement for the duration of projects development lifecycle. My comparison of the 'Waterfall', 'Spiral' and 'Agile' models / methodologies is based on a criteria of complexity for implementation , flexibility for addition of features, functionality or necessary changes and Usability within the set submission dates for the projects assessment.

The waterfall development model is a sequential process where each stage must be completed before the next stage of development can begin, this model is widely used for software development and is the simplest to understand. However, the waterfall model requires all tasks and requirements to be planned in advance making it very resistant to change and the addition of new features throughout the development process. Despite these drawbacks the waterfall methodology benefits from is sequential model as it best fits the projects submission dates and its timeline will best keep the project on track for assessment.

The spiral development model uses a iterative methodology where the product goes through each stage multiple times expanding on the product with each iteration, this model is often adopted for start-ups with an indecisive plan for their final product as this model allows for easy addition of features and shifting targets. The drawback of this model for my project is due to its iterative stages it will become extremely difficult to produce comprehensive submissions of each stage in time for each submission date. Although the spiral development model is attractive for my project due to its flexibility its iterative development stages will make it very difficult to implement.

The agile development model revolves around developing in sprints for each stage where all resources are allocated to producing specific features or set of features for a project in order to fast track a minimum viable product for the customer and receive feedback efficiently for implementation through the use of various methodologies, this development model is widely used to produce software as it commonly provides the best feedback and customer satisfaction. This development model can be very beneficial to use on my project as it allows me to focus on the design and implementation of various features for my project whilst maintaining an effective timeline for the completion of each development stage for submission, however these sprints may not be effective for the analysis and evaluation stages of the project as they favour traditional sequential development.

After the evaluation of each of these development models I have decided to continue with a waterfall development model with a nested agile model for the design and development stages during production. This model will provide me with the best balance of structure for working within the set submission dates for the project and flexibility to make changes as needed throughout the design and development stages of the project. The nested agile models will be conducted with three sprints in each the design and development stages to produce 'Core', the basic functionality and layout of the app, 'Features', the necessary integrations and functions for a minimum viable product that allows the client to review the product for any necessary changes and 'Extra' to produce the expansive tools and features for consumers that aren't necessary for the product to conduct its primary functionality.