Peer to Peer Messaging App Documentation

Connor Cuffe

Beaumaris Secondary College

Table of Contents

[Development model selection 4](#_Toc167355905)

[Waterfall development model description 4](#_Toc167355906)

[Pros 4](#_Toc167355907)

[Cons 4](#_Toc167355908)

[Summary 5](#_Toc167355909)

[Agile development model description 5](#_Toc167355910)

[Scrum 5](#_Toc167355911)

[Kanban 6](#_Toc167355912)

[Pros 6](#_Toc167355913)

[Cons 7](#_Toc167355914)

[Summary 7](#_Toc167355915)

[Spiral development model description 7](#_Toc167355916)

[Pros 8](#_Toc167355917)

[Cons 8](#_Toc167355918)

[Summary 8](#_Toc167355919)

[Conclusion 9](#_Toc167355920)

[Bibliography 10](#_Toc167355921)

# Development model selection

A development model helps to aid in the successful development of a final version of a product that satisfies all the requirements. In this section we consider the pros and cons of three major development models.

## Waterfall development model description

The waterfall development model breaks project development down into 4 distinct stages; requirements, design, implementation, and verification. These stages are then progressed though linearly with each succeeding stage requiring the full completion of previous stages.

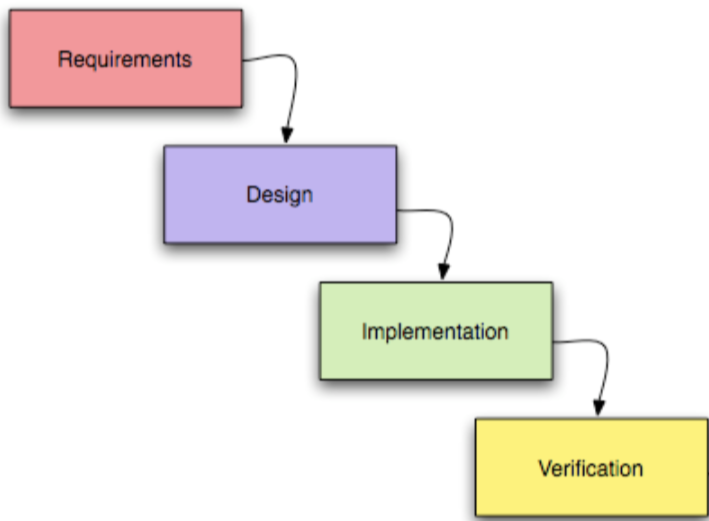


Figure 1. the four stages of the waterfall model

### **Pros**

* Fast completion of the projects original scope
  + The linear approach is suitable to small projects that are expected to be completed before significant change in the end user requirements. This is because changes are prohibited during later stages of the product.
* Cheap completion of the original scope
  + Prohibiting changes to the original scope allows the project to maintain a controllable budget and avoid cost overruns due to an increase in the scope of the project. In the industry this is referred to as ‘scope creep’.

### Cons

* Unable to grow and change based on new information during the development stages.
  + This is a considerable issue for large scale projects where the end user requirements, or unforeseen complications are more likely to necessitate fluidity of the design.

### Summary

In summary, the waterfall development model is suitable for this project due to its small-scale and limited scope reducing the chances of the users’ needs changing and invalidating the software solution.

## Agile development model description

The Agile development model is built around the belief that project design should be focused on the end user. Agile attempts to involve the end user as much as possible in the development of the project. Agile practitioners achieve this by creating a prototype product so that stakeholders can give feedback that can be incorporated in the next iteration of the product. Agile practitioners use this, and other techniques to focus their design on end users, which typically results in a more effective product. The agile development model is commonly implemented via either the Scrum, or the Kanban management frameworks which are discussed below.

### Scrum

The scrum framework emphasizes the following;

* iterative progress,
* flexibility,
* and continuous improvement.

This emphasis breaks down complex projects into smaller manageable stages called sprints, that normally last around 2 weeks and deliver the next iteration of the product that can be used as the final version if the stakeholders desire.

The scrum framework organizes a project team into three primary roles, these are the;

* Product owner: this is someone who defines the project requirements.
* Scrum master: the scrum master manages the scrum process, insuring that the team adheres to the scrum principles.
* And the Development team: a self-organizing group that is responsible for delivering the product in increments during each sprint.

During each sprint of a scrum project the scrum master organizes these key events to uphold scrum principles;

* Sprint planning: is a meeting held at the beginning of every sprint where the team outlines what they expect to complete by the end of current sprint.
* Daily stand up: a brief daily meeting where the team members progress is shared and the day’s work is planned.
* Sprint retrospective: these meetings are held at the end of every sprint to allow the team to assess their success in delivering to the sprints goals and to evaluate how they will improve in next sprint.

### Kanban

The Kanban management framework is similar to scrum in that it emphasizes iterative progress, flexibility and continual improvement. Kanban achieves this by having a visual board with columns representing different stages of work, such as ‘To Do’, ‘In Progress’, and ‘Done’. The size of these columns, and subsequently the number of tasks they can hold prevents overloading of the development team similar to sprint planning in scrum projects.

Kanban project managers also monitor the Kanban board for bottlenecks, or other opportunities for continual improvement of the process.

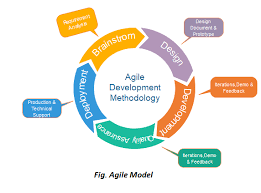


Figure 2: the agile development cycle

### Pros

The following is a list of positive benefits of all the Agile models discussed;

* User focused and empathic product.
  + The continual reassessment of the project’s requirements based on end user feedback of smaller prototypes, results in a much more user centric process ultimately creating a better product.
* Accounts for changing requirements or unforeseen barriers.
  + The act of continually readdressing the product requirements based on frequent end user feedback, and the inclusion of specific points in the project where the development can be revaluated to account for any unforeseen barriers.

### Cons

* Harder to predict budgets and timeline.
  + Due to the continual reassessment of the projects scope during its life time, budget and timeline are hard to predict and estimate.
* Can lead to costly redevelopment due to excessive scope creep.
  + The continual reassessment of the project requirements must be managed correctly to avoid expensive and time-consuming redevelopment caused by poorly prioritized changes to the projects scope. Incorrect prioritization can result in the end product not being viable as feature set of the delivered product may lack key features.

### Summary

In summary the Agile development models are not suited to this project due to the smaller scope resulting in more static user needs over the development period, therefore not requiring Agile`s more are adaptive and resilient work flow. This overall make agile not worth the additional overhead required to effectively prevent the possibility of scope creep and to make sure that the project stays within budget.

## Spiral development model description

The Spiral development model prioritizes the early mitigation of risks through the use of small iterations. While Agile models allows developers to move freely between development stages, the Spiral model calls for strict adherence to the project outline, and the completion of all steps in sequence to build each iteration of the project.

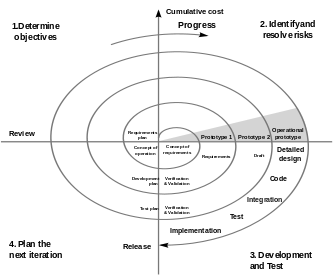


Figure 3: the spiral development cycle

### Pros

* Suitable for risk adverse projects.
  + The continual repetition of development stages results in higher resilience to development error as there are a large number of opportunities to detect any possible issues before the final product is delivered.

### Cons

* Development time often overruns because of risk management processes.
  + The strict repetition of development stages along with the need for risk assessment at every step, results in a much more expensive development cycle.
* Requires significantly skilled management.
  + The need for in-depth risk assessments though-out the development process requires practitioners to be highly skilled in risk mitigation.

### Summary

In summary the spiral development model is not suited for this project as unlike large risk project such as the North American Space Agency`s (NASA) space shuttle program (Beram, 2023) which used the Spiral model because they were very concerned about the potential for the loss of life should the product fail, this project does not involve any high risk elements. This ultimately results in the increased workload of the risk management being an unnecessary burden to the products development.

## Conclusion

The waterfall development model was chosen for this project as it would streamline the efficiency of the project, allowing for faster creation of a final product than the Agile and Spiral models due to its more direct approach. However, this does result in the loss of the risk mitigation provided by the spiral model and the more user-focused product that the agile model would provide.

This projects scope calls for a low-cost development and the product is intended for low-risk applications, so the risk management provided by the spiral model is considered unnecessary. The increased focus on user satisfaction of the agile model is also unnecessary as the small time-scale of the project makes it unlikely to require changes in scope.

In summary, the small scale of this project would not benefit from the spiral and agile models, when compared to the simplicity of the waterfall model.

# Bibliography

Beram, S. (2023, may 1). *risk driven development with the spiral model*. Retrieved from logrocket.com: https://blog.logrocket.com/product-management/risk-driven-development-with-the-spiral-model