Peer to peer Messaging App

Write Up

By Connor Cuffe

Contents

[Development model selection 2](#_Toc165807637)

[Waterfall development model description 2](#_Toc165807638)

[Pros: 2](#_Toc165807639)

[Cons: 2](#_Toc165807640)

[Summary 2](#_Toc165807641)

[Agile development model description 2](#_Toc165807642)

[Scrum: 2](#_Toc165807643)

[Kanban: 2](#_Toc165807644)

[Pros: 3](#_Toc165807645)

[Cons: 3](#_Toc165807646)

[Summary 3](#_Toc165807647)

[Spiral development model description 3](#_Toc165807648)

[Pros: 4](#_Toc165807649)

[Cons: 4](#_Toc165807650)

[Summary 4](#_Toc165807651)

[Final decision 4](#_Toc165807652)

[Authentication processes decision process. 4](#_Toc165807653)

[Process 1 4](#_Toc165807654)

[Cons 5](#_Toc165807655)

[Pros 5](#_Toc165807656)

[Process 2 5](#_Toc165807657)

[Cons 5](#_Toc165807658)

[Pros 5](#_Toc165807659)

[Process 3 5](#_Toc165807660)

[Cons 5](#_Toc165807661)

[Pros 5](#_Toc165807662)

[Final decision 6](#_Toc165807663)

# Development model selection

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.

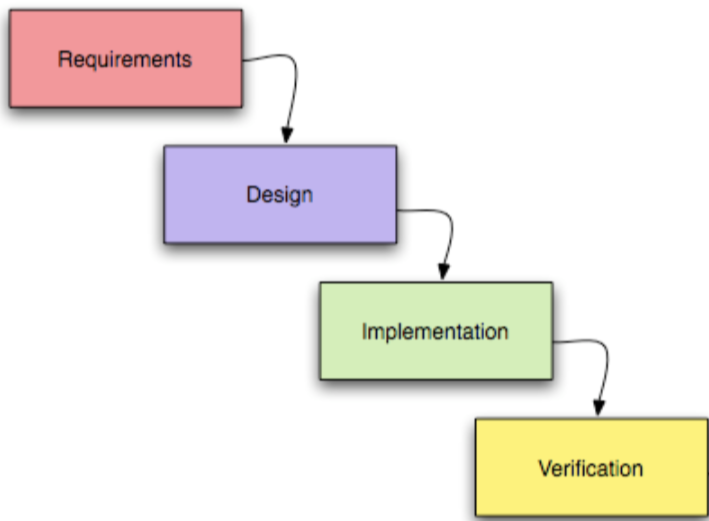


Figure 1. the four stages of the waterfall model

### 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 a increases 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 more suitable small-scale projects with limited scope.

## 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 by starting of with a minimum viable product so that stakeholders can give feedback resulting in the next iteration being more targeted towards the end user typically resulting in more effective product. The agile development model is commonly used via either the scrum or Kanban management frameworks.

### Scrum framework:

The scrum framework emphasises iterative progress, flexibility, and continuous improvement, breaking down complex projects into smaller manageable stages called sprints, that normally last around 2 weeks. The scrum framework has 3 primary roles;

* The product owner: who defines the projects requirement.
* The scum master: manages the scrum process, insuring that the team adheres to the scrum principles.
* Development team: a self-organizing group that is responsible for delivering the product in increments during each sprint.

The scrum management framework consists of several key events;

* Sprint planning: a meeting at the beginning of evert sprint where the team outlines what they expect to complete by the end of current sprint.
* Daily stand up: a shot daily meeting no more than half an hour where the teams progress is shared and the day’s work is planned.
* Sprint retrospective: a meeting at the end of every sprint where the team asses their success in delivering the sprints goals and evaluates how they should improve in the future.

### Kanban framework:

The Kanban management framework is similar to scrum in that it emphasises 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. These columns are then filled with sticky notes representing different tasks. The development team limit the number of tasks allowed in each column preventing overworking the team similar to how scrum sprints work. Kanban then aids the continual improvement of the team by monitoring the board to see any bottlenecks or stages where the tasks are being held up.

### Pros:

* A more user-friendly 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 much better product.
* has resilience to changing requirements or unforeseen barriers.
  + The ability to continually readdress the product requirements based on end user feedback and the current development of the product results in a more resilient development cycle.

### Cons:

* Harder to predict budgets and timeline.
  + Due to the continual reassessment of the original scope the projects budget and timeline are hard to predict resulting in an project with a larger than anticipated cost and development cycle.
* Can lead to scope creep.
  + The continual reassessment of the project requirements could if not manages efficiently lead to expensive and time-consuming scope creep which may result in the end product not being deliverable any more due to going over budget or simple not being needed any more due to the increased dev time.

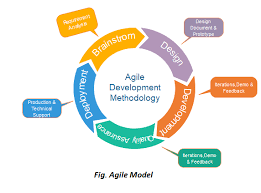


Figure 2: the agile development cycle

### Summary

In summary the agile development model is best used for large scale projects where the end users’ requirements are likely to change before the completion of the project, the agile development model also requires highly skilled management to combat the possibility of scope creep and to make sure that the project stays in budget.

## Spiral development model description

The spiral development model is designed to allow for early risk mitigation, it achieves this in a similar way to agile. The major difference being the stricter approach as agile allows for the developers to move freely between stages skipping them when it is necessary, whereas spiral calls for strict adherence to the project outline completing all steps in sequence to build the next iteration of the project and then again for the next.

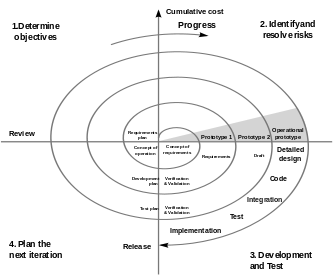


Figure 3: the spiral development cycle

### Pros:

* Significantly reduces risk
  + The continual repetition of development stages results in higher resilience to development error as there are a large number of chances to detect any possible issues with the final product before delivery.

### Cons:

* Much slower and more expensive development cycle than waterfall and agile.
  + The strict repetition of development stages along with the need for risk assessments results in a much more expensive development cycle.
* Require significantly skilled management.
  + The need for in depth risk assessments though the development process results in the need for management that is adequately adapt with risk mitigation.

### Summary

In summary the spiral development model is best suited for projects that require a flawless end product, these tend to be project where the stakes for the end product are extremely high such as NASA`s space shuttle [https://blog.logrocket.com](https://blog.logrocket.com/product-management/risk-driven-development-with-the-spiral-model/%23:~:text=An%20example%20product%20that%20was,in%20the%20product%20development%20process.) as the development extremely expensive and even the smallest flaw in the final product would have lead to it being completely unusable.

## Final decision

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 friendly product that the agile model would provide. Due to the low cost and stakes of this project the risk management provided by the spiral model is unnecessary. The increased user centricity of the agile model is also unnecessary as the small time scale of the project makes it unlikely for the end users requirements to change. In conclusion due to the small scale of this project the benefits of the spiral and agile models are outweighed when compared to the increased efficiency of the waterfall model.

# Authentication processes decision process.

### Process 1

* hash the password and username creating a unique value for each user.
* then send that hash to every other user on the network.
* the other users will check if hash is in their dictionary of valid hashes.
* if a majority agree then you have been validated.

### Cons

* insecure as each user must have a list of valid hashes, which they could just use one of
* also, may get false negatives due to hashes not being synced network wide.
* a small network is easily overwhelmed as by hacked clients faking valid or invalid responses either locking all accounts out or allowing easy access by bad actors.

### Pros

* Easy.

## Process 2

* encrypt the user name with the password using RSA.
* all clients on network know private key for the encryption.
* send encrypted username and username to all clients on network.
* the clients then attempt to unencrypt encrypted username if that equals the username then they validate the login.
* if majority of clients validate login - login continues.

Cons

* possible that as the client would have access to list of unencrypts they could find a username value and an encrypted value that would decrypt with one of the unencrypts.
* the username and encrypted username could be packed sniffed then used.

### Pros

* more secure than process one.
* password remains hidden from all users.
* a small network is easily overwhelmed as by hacked clients faking valid or invalid responses either locking all accounts out or allowing easy access by bad actors.

## Process 3

* send username to all users on network if that users unencrypt or is not in local database.
* users on network return the unencrypter for that user encrypted with a key unknown to everyone.
* 3. local client the attempts to unencrypt unencrypter or with password.
* 4. local client then unencrypts the user data with unencrypted unencryptor.

### Cons

* could attempt to brute force unencrypter unencrypting as they would be downloaded onto every user device.
* would require long calculation at account creation to mitigate threat of brute forcing.

### Pros

* more secure than Process 1 and 2.
* does not require network access.

## Final decision