**Software Life Cycle**

Pur essendo legati ad un piano da seguire nella realizzazione del progetto, il team ha optato per un approccio *Agile*, affinché ci fosse una rapida risoluzione ai problemi. Al fine di implementare ancora di più la creazione del progetto, si è scelto di seguire lo ***SCRUM LIFE CYCLE***:

* Pianificare tutti gli aspetti del progetto in anticipo è difficile, ma grazie allo SCRUM Life Cycle è possibile enfatizzare la gestione di ogni componente nel tempo.

Prima di tutto, bisogna definire i *requisiti* del progetto, per poi definire un **product backlog**, contenente le varie funzionalità del sistema.

* Serve molta coordinazione all’interno del gruppo e ogni suo membro mette le proprie conoscenze/competenze al servizio del team.
* Il team deve adattarsi rapidamente ai cambiamenti in maniera flessibile: se si verificassero modifiche durante il processo, bisogna essere pronti ad apportarle, purché ciò non danneggi l’intero sistema.
* Il metodo *Agile* coinvolge ogni utente di ogni passo intrapreso; pertanto, i cicli di sviluppo del software devono essere brevi e incrementali, tenendo conto che non è mai presente un piano estensivo per i cicli futuri.
* ***Sprint***:
* Ogni sprint dura una settimana.
* Ogni giorno i membri del team si consultano, attraverso il *Daily Scrum*, per risolvere eventuali anomali e correggere errori
* Alla fine di ogni sprint avviene lo *Sprint Review*, in cui viene definito quanto è migliorato il progetto con le nuove modiche
* ***Organizzazione Gruppo:***
* Il ruolo di *Scrum Master* non è mai appartenuto ad un singolo membro: al fine di ottimizzare l’implementazione del progetto, il ruolo cambia di volta in volta e viene passato ai vari collaboratori, in relazione alla componente del progetto che deve essere completata (Documentazione, Diagrammi, Codice).
* La divisione dei compiti è importante nello sviluppo di un progetto, ma data la mancanza di specializzazione, ogni membro del team ha svolto le diverse attività ricomprendo il ruolo adatto alla situazione (*backend/frontend*).
* Spesso è necessario lavorare in coppia: un membro scrive mentre l’altro revisiona e segnala eventuali errori; ciò ottimizza il lavoro di squadra.