**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. I metodi *agili* coinvolgono gli utenti in ogni passaggio del ciclo di vita, per rispondere alle loro esigenze e alzare il livello di qualià finale.

Al fine di implementare ancora di più la creazione del progetto, si è scelto di seguire l’approccio ***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, il *product owner* deve definire i *requisiti* prioritari del progetto, **product backlog**, contenente le varie funzionalità del sistema e che poi sarà suddiviso dallo *SCRUM Master* secondo le parti da completare (**sprint backlog**), che devono essere soddisfatte

nei periodi di tempo prestabiliti (*sprint*).

* 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 e non ne abbassi la qualità finale.
* La comunicazione è molto importante all’interno del gruppo: attraverso app di messagistica o issue di GitHub, si deve sempre notificare al gruppo sulle nuove proposte/modifiche apportate e i tempi di risposta devono essere brevi, al fine di evitare ritardi nella tabella di marcia.
* 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 ha la durata di una settimana.
* Uno sprint deve rispondere ai requisiti di priorità indicati (*Sprint Backlog*).
* Ogni settimana, i membri del team definiscono un giorno in cui incontrarsi, per effettuare un consulto sull’andamento del progetto e risolvere eventuali problematiche/errori si consultano (*Weekly Scrum*).
* 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); ciò permette di imparare il ruolo e adattarsi alle varie situazioni, sia come “leader” che come “sottoposto”. Lo *Scrum Master* ha il compito di confermare le modifiche richieste (valutando se sono necessarie o meno), ma deve anche fornire supporto e motivazione per il resto del gruppo)
* 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 (*Pair Programming*): un membro scrive mentre l’altro revisiona e segnala eventuali errori (i ruoli possono scambiarsi in qualsiasi momento); ciò ottimizza il lavoro di squadra.