

Gestione del Progetto

Gabriele Chignoli

Maggio 2025

Contents

1	Ciclo di Vita del Software	2
2	Configuration Management	2
3	People Management	3

1 Ciclo di Vita del Software

Dopo un'analisi dei principali modelli disponibili per la modellazione del ciclo di vita del software, si è giunti alla conclusione che sia gli approcci "pesanti" che quelli più agili hanno utili proprietà che vanno combinate per portare ad un processo più, secondo il team, completo e "umano".

Dei processi pesanti, viene considerata la pianificazione iniziale fondamentale a garantire che il progetto si mantenga nel tempo all'interno di binari definiti e, preferibilmente, solidi. Tuttavia si considera che tale processo rischi di diventare troppo morbosio e vincolante; in particolare l'impossibilità di tornare ad operare su fasi iniziali già completate, viene vista come una grande limitazione.

Per questi motivi è stato deciso di adottare il *Modello a Cascata* come modello base per la produzione del progetto; in aggiunta si intende concedere al team di "risalire la cascata" ogni qualvolta sia necessario (credendo nel fatto che le risalite diminuiscano con l'avanzamento del progetto) e di quanti step si vuole, adottando così alcuni aspetti caratteristici di un approccio più iterativo, e quindi in parte anche più agile. Infine, si considerano le tre fasi finali di implementazione, testing e manutenzione (che include il refactoring) come un processo continuo in cui ogni parte è fondamentale per la seguente, e necessita di quella che l'ha preceduta. Si ritiene dunque che una visione e rappresentazione più ciclica di tali fasi sia più adatta.

Tenendo in mente queste accortezze, si è tentato di rappresentare anche graficamente il modello adottato:

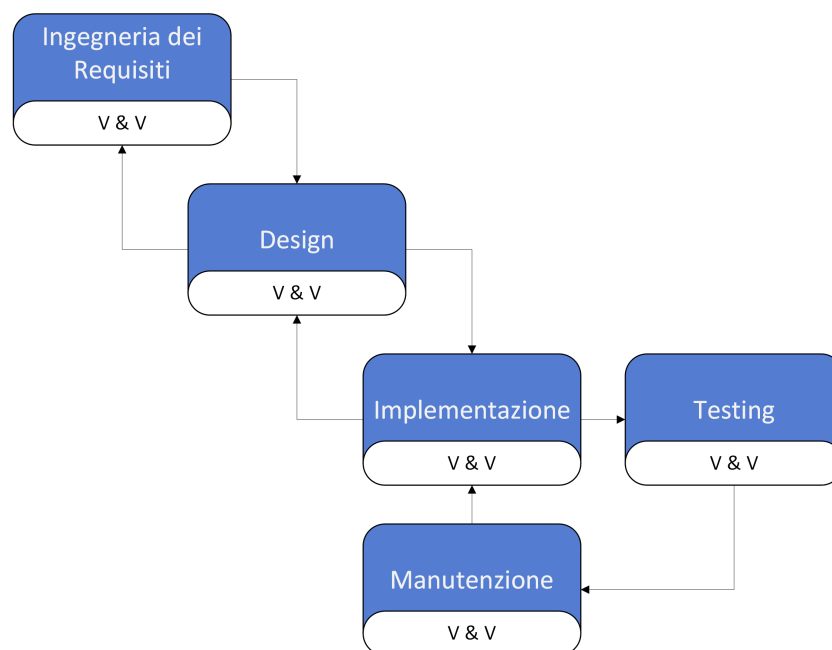


Figure 1: Modello a Cascata Modificato

2 Configuration Management

Il progetto utilizza per la gestione della configurazione la piattaforma *GitHub*, della quale si intendono utilizzare le principali funzionalità disponibili. Inoltre si utilizzerà la *Kanban*

Board messa a disposizione nella sezione *Progetti* di *GitHub*, con la quale tenere traccia dei compiti e del loro stato.

3 People Management

Pur essendo composto attualmente da un solo membro, verrà comunque simulata la presenza di un secondo sviluppatore (sempre controllato dallo sviluppatore principale). Si procede dunque a stabilire l'organizzazione del team secondo la quale si intende operare.

Date le dimensioni del progetto abbastanza contenute e il tipo di personale disponibile (studenti universitari), si ritiene che la struttura più adatta per lo svolgimento del compito sia un'**Organizzazione a Matrice**. In tal modo, tutti i membri possono operare su diverse (anche tutte) parti del progetto e quindi imparare, facendo, come costruire un software nella sua completezza, dalla documentazione e progettazione, al testing e all'implementazione, sfruttando questo ambiente "sandbox", invece che avere effettive responsabilità come nel mondo lavorativo.

Allocazione lavoro al 24/05/2025 Le principali aree di lavoro e la loro allocazione al personale attuale sono elencate nella seguente tabella:

Area di Lavoro	Personale	
	Gabriele Chignoli	...
Documentazione	100%	...
Interfaccia Grafica	100%	...
Logica dell'applicazione	100%	...
Testing	100%	...
Data Base	100%	...
Manutenzione	100%	...

Allocazione lavoro al 01/08/2025

Area di Lavoro	Personale	
	Gabriele Chignoli	Developer #2
Documentazione	80%	20%
Interfaccia Grafica	50%	50%
Logica dell'applicazione	80%	20%
Testing	...	100%
Data Base	100%	...
Manutenzione	50%	50%

Commenti su allocazione effettiva - 04/09/2025 Non è stato possibile tener traccia delle effettive percentuali di lavoro esercitate da ogni membro. Sarebbe stato utile introdurre un sistema di *#tag* da utilizzare per i commit (o per i branch) in modo da poter contare il numero di commit per membro per area di lavoro. Si può notare comunque dalle statistiche ottenute da *Github* che vi è un forte squilibrio, principalmente causato dall'entrata tardiva del secondo membro nello sviluppo.



Figure 2: Statistiche Contributors Github

Si stima che i risultati effettivi siano più in linea con la seguente tabella:

Area di Lavoro	Personale	
	Gabriele Chignoli	Developer #2
Documentazione	100%	...
Interfaccia Grafica	20%	80%
Logica dell'applicazione	80%	20%
Testing	40%	60%
Data Base	100%	...
Manutenzione	50%	50%