Il modello del ciclo di vita scelto è stato il Rational Unified Process (RUP) (Capitolo 2), con un team Cross-Funzionale, cioè ogni membro del team può assumere più ruoli (sviluppatore, tester…), (Capitolo 5), e per il configuration management useremo git hub con i relativi comandi (Capitolo 4).

**Capito 3: Software Life Cycle**

Strutteremo il Rational Unified Process (RUP) in questi 4 fasi:

1. Una fase iniziale:

le principali attività di questa fase sono:

* Identificazione degli stakeholder
* Raccolta dei requisiti principali
* Analisi di fattibilità

Output: Modelli dei casi d’uso principali.

1. Una fase di elaborazione:

le principali attività di questa fase sono:

* Progettazione dell’architettura del software
* Pianificazione dettagliata.

Output: Modello architetturale, Modelli UML

1. Una fase di costruzione:

le principali attività di questa fase sono:

* Sviluppo del software
* Test continuo
* Gestione della configurazione tramite github

Output: Versione funzionante del software.

1. Una fase di transizione

Le principali attività di questa fase sono:

* Test finale con simulazioni reali

Output: Versione stabile del software, documentazione completa.

Oltre al modello RUP, seguiremo anche un approccio Model Driven Architecture (MDA), in particolare generiamo automaticamente porzioni di codice per:

* lo schema del Database: Generiamo uno Script SQL per la creazione della Database dal diagramma UML delle Classi

Adottiamo un'organizzazione si sprint flessibile, senza un periodo di tempo fisso, con un meeting alla fine di ogni sprint:

|  |  |
| --- | --- |
| Sprint | Attività |
|  |  |
|  |  |
|  |  |

**Capito 4: Configuration Managment**

Per il configuration management useremo github con i relativi comandi.

Alcune Statistiche: (**ToDo**)

**Capito 5: People Management and Team Organization**