**Software Configuration Management**

GitHub è lo strumento prescelto per la gestione di configurazione dell’intero progetto, sia per il codice che per la documentazione; a tale scopo, è stata creato un repository, suddiviso in varie parti (branches, “rami”) per la gestione ottimale delle singole componenti.

***GESTIONE DELLE MODIFICHE***

La baseline del progetto si trova nel branch “main”: per effettuare modifiche su una determinata parte del progetto, viene creato un nuovo branch al fine di lavorare sullo specifico componente (codice, interfaccia grafica, diagrammi UML…); ogni membro del gruppo può lavorare su tutti i branch creati e, per evitare incongruenze/collisioni tra i vari lavori, si possono creare più sotto-branch per risolvere problemi specifici.

Eventuali modifiche sui rami secondari possono essere trasferite su quello principale, attraverso l’uso di pull-request, che poi vengono approvate dallo scrum master assegnato per tale componente del progetto (prima di effettuare la modifica, i membri del team devono confrontarsi per capire se tale modifica è necessaria o meno, ma spetta allo scrum master avere l’ultima parola sull’approvare o meno la nuova componente). In conclusione al progetto, si effettuano attività di *merge* (“unione”) per trasferire tutte le parti più importanti del progetto, situate nei vari branches, verso il main.

***ASSEGNAZIONE DEI COMPITI***

I compiti assegnati ai tre collaboratori vengono definiti in fase di configurazione, così come il ruolo di *scrum master*, che viene assegnato ad un singolo membro per ogni parte del Progetto: egli ha sempre l’ultima parola sulle modifiche da apportare, ma cerca il confronto con gli altri membri, per essere sicuro di non abbassare il grado di qualità del progetto.

Prima di iniziare con il lavoro vero e proprio, deve essere definita una lista di desideri con priorità per i vari punti del progetto (*product backlog*); successivamente, all’inizio di ogni *sprint* (ognuno della durata di una settimana), lo *scrum master* deve definire una parte del backlog (*sprint backlog*) da implementare. Il team, motivato dallo scrum master, deve essere in grado di completare i punti dello *sprint backlog* entro la conclusione dello stesso sprint: durante la settimana, i collaboratori si incontrano per valutare i progressi svolti nei giorni precedenti e assicurarsi che il lavoro proceda senza troppi intoppi (*weekly scrum*). L’obiettivo è andare a concludere, passo per passo, i vari punti del *product backlog*.

Per risolvere errori/malfunzionamenti/difetti, o per valutare nuove modifiche proposte, si aprono apposite *issue*, assegnate al membro designato che deve risolvere il problema e poi chiudere l’issue.