



Laurea Magistrale in informatica-Università di Salerno
Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci



Quality Management Plan Progetto UniSeats

Riferimento	
Versione	1.1
Data	19/01/2021
Destinatario	Dipartimento di Informatica dell'Università degli Studi di Salerno
Presentato da	Vincenzo Russo
Approvato da	



Laurea Magistrale in informatica-Università di Salerno
Corso di *Gestione dei Progetti Software*- Prof.ssa F. Ferrucci



Laurea Magistrale in informatica-Università di Salerno
Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci

Revision History

Data	Versione	Descrizione	Autori
13/01/2021	1.0	Stesura documento	V. Russo
19/01/2021	1.1	Aggiornamento documento (Travis, Mockito)	V.Russo



Sommario

Revision History	3
1. Introduzione	6
1.1. Scopo del documento.....	6
1.2. Evoluzione del documento	6
2. Struttura Gestionale	14
2.1. Organigramma.....	14
2.2. Task.....	14
2.2.1. Definizione piano di qualità.....	14
2.2.2. Definizione dei documenti.....	15
2.2.3. Implementazione del codice	15
2.2.4. Revisione artefatto	15
2.3. Ruoli e responsabilità	16
2.4. Comunicazione	16
3. Documentazione.....	19
4. Standard, Pratiche, Convenzioni e Regole.....	21
4.1.1. Standard per la Documentazione	21
4.1.1.1. Standard per il Processo di Documentazione.....	21
4.1.1.2. Regole di pubblicazione dei documenti	21
4.1.1.3. Stili di Base per i documenti	22
4.1.1.4. Dettagli colori e stili tabelle	23
4.1.1.4.1. Dettagli colori	23
4.1.1.2.1. Stili tabelle	23
4.2.1. Regole sugli Identificatori dei Documenti	23
4.2.2. Metriche per la valutazione del progetto.....	24
4.2.3. Metriche per la valutazione della documentazione.....	24
4.2.4. Regole per Riflettere l'aggiornamento di un Documento	24
4.3.1. Standard per gli Artefatti.....	25
4.3.1.1. Convenzione per gli Activity Diagram.....	25
4.3.1.2. Convenzione per i Requisiti Funzionali	25
4.3.1.3. Convenzione per le User Stories.....	26
4.3.1.4. Convenzione per i Requisiti Non Funzionali	26
4.3.1.1. Convenzione per gli Scenari.....	26
5. Training.....	31



Laurea Magistrale in informatica-Università di Salerno
Corso di *Gestione dei Progetti Software*- Prof.ssa F. Ferrucci

6.	Gestione dei Rischi	32
7.	Glossario	35



1. Introduzione

1.1. *Scopo del documento*

Il Quality Management Plan è il documento dove vengono gli specificati gli standard da seguire al fine di ottenere un prodotto di qualità. Il prodotto è di qualità se rispetta gli standard dettati dal Top Manager ed è conforme alle specifiche concordate con il cliente e alle normative legali vigenti.

Al fine di ottenere un prodotto di qualità in questo documento verranno descritti/e:

- I criteri di qualità del progetto UniSeats;
- Gli standard e le linee guida da applicare per la comunicazione, la stesura dei documenti e la scrittura del codice sorgente;
- Le metriche e le tecniche usate per valutare la qualità degli artefatti prodotti.

1.2. *Evoluzione del documento*

Durante lo svolgimento del progetto vi saranno continue revisioni e modifiche del documento, volte garantire una alta e sempre maggiore qualità degli artefatti.

1.3. *Definizioni, Acronimi e Abbreviazioni*

1.3.1. *Definizioni*

- **Dipartimento:** struttura organizzativa che all'interno delle università italiane promuove e
- **Business Case:** Documento utile per lo studio di fattibilità economica, utilizzato per stabilire la validità dei benefici di un progetto. Elenca gli obiettivi e le ragioni per l'avvio del progetto. Aiuta a misurare il successo rispetto agli obiettivi del progetto;
- **Project Charter:** Documento iniziale di un progetto, la sua ufficializzazione. Il project manager attraverso il project charter definisce in maniera chiara quali saranno gli impegni in termini di risorse, siano esse umane, finanziarie o tecniche, al raggiungimento dell'obiettivo del progetto;
- **Team Contract:** Documento che stabilisce in modo chiaro gli obblighi e gli impegni di ciascun membro del team;
- **Scope Statement:** Rielaborazione svolta dal Project Manager del documento di avvio progetto (o project charter);
- **Work Breakdown Structure:** Strumento utilizzato per la scomposizione analitica di un progetto in parti elementari. Lo scopo è quello di organizzare il lavoro in elementi più facilmente gestibili e rendere meno complessa la comprensione del progetto, in modo da comunicare a tutti i soggetti coinvolti (stakeholder) le fasi e le attività da svolgere per il raggiungimento di un obiettivo;



- **WBS Dictionary:** Documento che descrive ciascun task e sotto-task presente nella Work Breakdown Structure. Per ciascuno di questi, fornisce vari dettagli tra cui le persone coinvolte, i documenti collegati e le date di inizio e fine di quel task/sotto-task;
- **Schedule delle Attività:** Lista, sotto forma di diagramma di Gannt o Pert, di tutte le attività pianificate che mostrano in chiaro le date stabilite di inizio e fine attività, le tempistiche necessarie e le risorse necessarie per portare a termine tali attività;
- **Software Project Management Plan:** Processo decisionale di supporto al governo dei processi operativi. Tende ad ottenere il raggiungimento degli obiettivi del progetto, utilizzando al meglio le risorse e rispettando le scadenze di realizzazione (tempi), i limiti di costi e garantendo la soddisfazione dei requisiti (qualità);
- **Quality Management Plan:** Documento che illustra le pratiche e i processi di qualità per il progetto, assicurando che i requisiti di qualità siano pianificati e soddisfatti in modo appropriato;
- **Status Report:** Rapporto che riassume una situazione particolare relativa ad un determinato periodo di tempo;
- **Post Mortem Review:** Processo, solitamente eseguito a conclusione di un progetto, per determinare e analizzare elementi che hanno avuto esito positivo o negativo. Hanno lo scopo di rendere noto come migliorare i processi e promuovere le best practice;
- **Requirement Analysis Document:** Descrive il sistema in termini di requisiti funzionali e non funzionali e funge da base contrattuale tra il cliente e lo sviluppatore;
- **Modello funzionale:** Struttura organizzativa all'interno della quale ogni divisione aziendale svolge una funzione specializzata;
- **System Design Document:** Descrive completamente il sistema a livello di architettura, inclusi i sottosistemi e i loro servizi, la mappatura hardware, la gestione dei dati, il controllo degli accessi e le boundary conditions;
- **Test Plan:** Documento che dettaglia gli obiettivi, le risorse e i processi per un test specifico. Il piano in genere contiene una comprensione dettagliata del flusso di lavoro finale;
- **Test Case Specification:** Documento che per ogni Test Case specificato indica, input, flusso di eventi, oracolo (output atteso).
- **Test Case:** Insieme di condizioni o variabili in base alle quali un tester determinerà se un sistema in prova soddisfa i requisiti o funziona correttamente. Il processo di sviluppo dei casi di test può anche aiutare a trovare problemi nei requisiti o nella progettazione di un'applicazione;



- **Object Design Document:** Descrive i trade-offs della progettazione degli oggetti realizzati dagli sviluppatori, le linee guida che hanno seguito per le interfacce del sottosistema, la scomposizione dei sottosistemi in pacchetti e classi e le interfacce;
- **Unit Test:** Processo di sviluppo del software in cui le parti testabili più piccole di un'applicazione, denominate unità, vengono esaminate singolarmente e indipendentemente per verificarne il corretto funzionamento;
- **Test Summary Report:** Riassume i risultati della fase di testing;
- **Test Incident Report:** Descrizione di un incidente osservato durante il test, cioè una variazione o deviazione osservata nel comportamento del sistema da quanto previsto. L'incidente è fondamentalmente un comportamento o una risposta imprevista che richiede un'indagine;
- **Manuale di Installazione:** Documento di comunicazione tecnica destinato a fornire assistenza alle persone che utilizzano il nostro sistema;
- **Codice Sorgente:** Collezione di tutto il codice prodotto, comprensivo di commenti, che va a definire il nostro sistema. Esso si compone di tutti i package, le classi, i moduli e le librerie presenti nel progetto;
- **Statement Of Work:** Documento che definisce attività, deliverable e tempistiche specifiche del progetto per un fornitore che fornisce servizi al cliente;
- **Check List:** Documento che specifica le metriche e gli standard da seguire per effettuare un artefatto o documento di qualità;
- **Metriche Interne:** si applicano alle proprietà intrinseche del prodotto durante le fasi di analisi, design e codifica. Misurano gli attributi interni del software e forniscono indicazioni sulle caratteristiche esterne del prodotto finale, tramite l'analisi statica dei prodotti intermedi (specifiche tecniche e codice sorgente);
- **Metriche Esterne:** misurano i comportamenti del prodotto software rilevabili dai test, dall'operabilità, dall'osservazione durante la sua esecuzione. Sono scelte in base alle caratteristiche che il prodotto finale dovrà dimostrare durante la sua esecuzione;
- **Agenda:** Documento che specifica tutte le attività che verranno svolte durante un meeting;
- **Minuta:** Documento che identifica tutte le attività che sono state svolte durante un meeting;
- **Activity Diagram:** Diagramma in UML per descrivere gli aspetti dinamici del sistema. Il diagramma delle attività è fondamentalmente un diagramma di flusso che rappresenta il flusso da un'attività a un'altra attività. L'attività può essere descritta come un'operazione del sistema;
- **Attori:** Tutti coloro che si interfacciano con il sistema e anche il sistema stesso;



- **Requisiti Funzionali:** Funzione di un sistema o di una sua componente, in cui una funzione è descritta come una specifica del comportamento partendo dagli input e osservando gli output;
- **Requisiti Non Funzionali:** Un requisito che specifica i criteri che possono essere utilizzati per giudicare il funzionamento di un sistema, piuttosto che i comportamenti specifici. Sono spesso chiamati "attributi di qualità" di un sistema;
- **Modello del Sistema:** Uso di modelli per concettualizzare e costruire il nostro sistema;
- **Scenario:** Descrive un uso specifico di un sistema proposto, catturando il sistema come se fosse visto dall'esterno;
- **Use Case:** Elenco di azioni o passaggi di eventi che generalmente definiscono le interazioni tra un ruolo (noto in UML come Attore) e un sistema per raggiungere un obiettivo. L'attore può essere un umano o un altro sistema esterno;
- **Use Case Diagram:** Diagrammi dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso;
- **Modello ad Oggetti:** Sistema che è modellato attraverso l'uso di tecniche orientate agli oggetti. Consente la creazione di un software architettonico o di un modello di sistema prima dello sviluppo o della programmazione;
- **Class Diagram:** Tipo di diagramma che descrive la struttura di un sistema mostrando le sue classi, gli attributi di tali classi, le operazioni (o metodi) e le relazioni tra gli oggetti;
- **Object Diagram:** è un diagramma di tipo statico previsto per descrivere un sistema in termini di oggetti e relative relazioni. Il diagramma è molto simile a quello del Class diagram e descrive gli oggetti e le relative relazioni che sono istanziate in un determinato tempo $t+1$.;
- **Modello Dinamico:** Descrive i componenti del sistema che hanno un comportamento dinamico;
- **Sequence Diagram:** Diagramma che mostra, per uno scenario particolare di un caso d'uso, gli eventi che gli attori esterni generano, il loro ordine e gli eventuali eventi inter-sistema;
- **Statechart Diagram:** Illustrazione degli stati che un oggetto può raggiungere e le transizioni tra questi stati. In questo contesto, uno stato definisce uno stadio nell'evoluzione o nel



comportamento di un oggetto, che è un'entità specifica in un programma o l'unità di codice

che rappresenta quell'entità;

- **Interfaccia Utente:** I mezzi con cui interagiscono l'utente e il sistema, in particolare l'uso di dispositivi di input e software;
- **Navigational Path:** Diagramma il cui scopo è quello di mostrare tutti i passaggi che bisogna compiere per poter ottenere un risultato;
- **Mock-up:** sono prototipi che riflettono le scelte progettuali per schemi di colori, layout, tipografia, iconografia, grafica della navigazione e atmosfera generale del prodotto.



1.1.2. Acronimi

- **C06_QMP_1.0:** Utilizzata per indicare il Quality Plan;
- **TM:** Abbreviazione utilizzata per indicare i Team Members;
- **PM:** Abbreviazione utilizzata per indicare il Project Manager;
- **SS:** Abbreviazione utilizzata per indicare lo Scope Statement;
- **BC:** Abbreviazione utilizzata per indicare il Business Case;
- **PC:** Abbreviazione utilizzata per indicare il Project Charter;
- **WBS:** Abbreviazione utilizzata per indicare la Work Breakdown Structure;
- **WBSDic:** Abbreviazione utilizzata per indicare la Work Breakdown Structure Dictionary;
- **SA:** Abbreviazione utilizzata per indicare lo Schedule delle Attività;
- **SPMP:** Abbreviazione utilizzata per indicare il Software Project Management Plan;
- **QMP:** Abbreviazione utilizzata per indicare il Quality Management Plan;
- **SR:** Abbreviazione utilizzata per indicare lo Status Report;
- **PMR:** Abbreviazione utilizzata per indicare la Post Mortem Review;
- **RAD:** Abbreviazione utilizzata per indicare il Requirement Analysis Document;
- **SDD:** Abbreviazione utilizzata per indicare il System Design Document;
- **TP:** Abbreviazione utilizzata per indicare il Test Plan;
- **TC:** Abbreviazione utilizzata per indicare i Test Case;
- **ODD:** Abbreviazione utilizzata per indicare l'Object Design Document;
- **UT:** Abbreviazione utilizzata per indicare gli Unit Test;
- **TSR:** Abbreviazione utilizzata per indicare il Test Summary Report;
- **TIR:** Abbreviazione utilizzata per indicare il Test Incident Report;
- **SOW:** Abbreviazione utilizzata per indicare lo Statement Of Work;
- **MU:** Abbreviazione utilizzata per indicare il Manuale Utente;
- **RF:** Abbreviazione utilizzata per indicare i Requisiti Funzionali;
- **RF_[numero]:** Abbreviazione utilizzata per indicare un particolare Requisito Funzionale. Il numero è un intero che indica il codice identificativo del particolare Requisito Funzionale;
- **US:** Abbreviazione utilizzata per indicare una User Story;
- **US_[numero]:** Abbreviazione utilizzata per indicare una particolare User Story. Il numero è un intero che indica il codice identificativo della particolare User Story;



- **RNF:** Abbreviazione utilizzata per indicare i Requisiti Non Funzionali;
- **RNF_[numero]:** Abbreviazione utilizzata per indicare un particolare Requisito Non Funzionale. Il numero è un intero che indica il codice identificativo del particolare Requisito Non Funzionale;
- **SC:** Abbreviazione utilizzata per indicare gli Scenari;
- **SC_[numero]:** Abbreviazione utilizzata per indicare un particolare Scenario. Il numero è un intero che indica il codice identificativo del particolare Scenario;
- **UC:** Abbreviazione utilizzata per indicare gli Use Case;
- **UC_<ACRONIMOGESTIONE>_XY:** Abbreviazione utilizzata per indicare un particolare Use Case. In particolare ACRONIMOGESTIONE può essere PREN, GPREN, GPP
- **PREN:** Acronimo utilizzato per indicare la Prenotazione;
- **GPREN:** Acronimo utilizzato per indicare la Prenotazione (Modifica, Annullamento);
- **GPP:** Acronimo utilizzato per indicare la Prenotazione di un Gruppo Privato;
- **UCD:** Abbreviazione utilizzata per indicare gli Use Case Diagram;
- **UCD_[numero]:** Abbreviazione utilizzata per indicare un particolare Use Case Diagram. Il numero è un intero che indica il codice identificativo del particolare Use Case Diagram;
- **CD:** Abbreviazione utilizzata per indicare i Class Diagram;
- **CD_[numero]:** Abbreviazione utilizzata per indicare un particolare Class Diagram. Il numero è un intero che indica il codice identificativo del particolare Class Diagram;
- **OBJD:** Abbreviazione utilizzata per indicare gli Object Diagram;
- **OBJD_<ACRONIMOPOSTO>_[numero]:** Abbreviazione utilizzata per indicare l'Object Diagram del posto singolo. In particolare ACRONIMOPOSTO può essere PS, PG, MP
Il numero è un intero che indica il codice identificativo del particolare Object Diagram;
- **PS:** Acronimo utilizzato per indicare la Prenotazione di un Posto Singolo;
- **PG:** Acronimo utilizzato per indicare la Prenotazione posto di un Gruppo Pubblico;
- **MP:** Acronimo utilizzato per indicare la Modifica della Prenotazione;
- **SCD:** Abbreviazione utilizzata per indicare gli Statechart Diagram;



- **SCD_[numero]:** Abbreviazione utilizzata per indicare un particolare Statechart Diagram. Il numero è un intero che indica il codice identificativo del particolare Statechart Diagram;
- **SD:** Abbreviazione utilizzata per indicare i Sequence Diagram;
- **SD_[numero]:** Abbreviazione utilizzata per indicare un particolare Sequence . Il numero è un intero che indica il codice identificativo del particolare Sequence Diagram;
- **NP:** Abbreviazione utilizzata per indicare i Navigational Path;
- **NP_[numero]:** Abbreviazione utilizzata per indicare un particolare Navigational Path. Il numero è un intero che indica il codice identificativo del particolare Navigational Path;
- **UI:** Abbreviazione utilizzata per indicare i Mock-ups;
- **UI_[numero]:** Abbreviazione utilizzata per indicare un particolare Mock-up. Il numero è un intero che indica il codice identificativo del particolare Mock-up.
- **TC_[numero]:** Abbreviazione utilizzata per indicare un particolare Test Case. Il numero è un intero che indica il codice identificativo del particolare Test Case;
- **TC:** Abbreviazione utilizzata per indicare i Test Case;

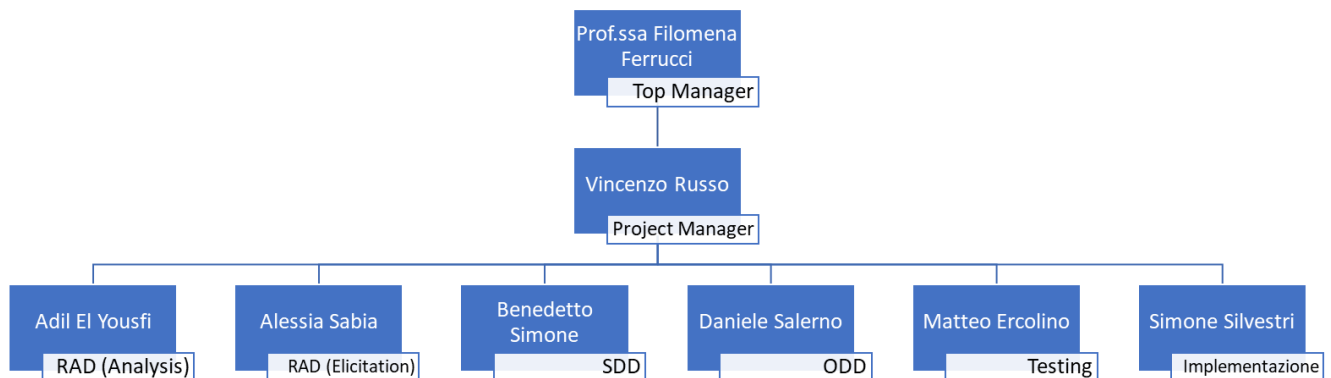
1.2. Riferimenti

- Cengage Learning – “Information Technology Project Management”, Autori: Kathy Schwalbe;
- Prentice Hall – Pearson – Object-Oriented Software Engineering – Using UML, Patterns and Java. Autori: Bernd Bruegge & Allen H. Dutoit;
- Documentazione di Progetto;



2. Struttura Gestionale

2.1. Organigramma



2.2. Task

2.2.1. Definizione piano di qualità

INPUT: Statement Of Work, informazioni condivise con il cliente (Top Manager).

Vengono definiti gli standard di qualità del progetto in base agli standard dell'organizzazione dettati dal Top manager, ai costi, benefici e criteri di accettazione del cliente.

OUTPUT: Quality Plan.



2.2.2. Definizione dei documenti

INPUT: Quality Plan, libro di testo, Checklist fornite dal Top manager.

Definizione degli standard di documentazione, i documenti che verranno sviluppati dai Project Manager e dai Team Member dovranno essere conformi agli standard definiti Quality Plan e nelle Check List e dovranno avere i contenuti descritti nel libro di testo.

OUTPUT: Documentazione Finale.

2.2.3. Implementazione del codice

INPUT: Quality Plan, System Design Document, Object Design Document.

Sviluppo del codice sorgente della piattaforma UniSeats, il codice prodotto dovrà essere conforme agli standard definiti nel Quality Plan e rispettare la struttura definita nel System Design Document e nell' Object Design Document.

OUTPUT: Codice sorgente.

2.2.4. Revisione artefatto

INPUT: documento sviluppato e check list di revisione.

Il giorno prima della consegna il responsabile del documento dovrà procedere alla revisione e (eventualmente) dovrà rendere conforme il documento agli standard di qualità. Il giorno prima della consegna del documento i Project Manager revisioneranno il documento corretto dal responsabile, nel caso il documento presenta ancora difetti i manager li segnaleranno al responsabile che provvederà alla correzione.

OUTPUT: documento revisionato



2.3. Ruoli e responsabilità

Per ogni deliverable è stato individuato un responsabile, il cui ruolo è quello di revisionare il documento prodotto. In questo modo si è cercato di responsabilizzare i Team Members. Quest'ultimo sarà stimolato a controllare la qualità del documento sin dalle prime fasi in modo da non avere un carico di lavoro eccessivo durante il periodo di revisione, considerando che egli si occupa della stesura del documento.

I ruoli e i responsabili individuati sono presenti nell'organigramma.

2.4. Comunicazione

La comunicazione è stata strutturata in maniera Peer-To-Peer permettendo una comunicazione rapida fra tutti i membri del team. I motivi per cui è stato scelto questo approccio sono i seguenti:

- Evitare che le comunicazioni fra due sotto team debba passare per i PM che dovranno comunicare quanto concordato con un team all'altro team;
- I Team Members possono comunicare fra di loro e con il Project Manager in modo rapido;
- Aumentare la produttività

Le informazioni verranno comunicate in due modi in maniera sincrona e asincrona.



2.4.1. Comunicazioni Sincrone

2.4.1.1. Meeting

I meeting sono la parte fondamentale della nostra comunicazione, in quanto verranno prese decisioni, chiariti dubbi riguardanti i task e le valutazioni e spiegati i task da svolgere durante la settimana.

I meeting vengono svolti tutti i lunedì dalle 17:00 alle 18:00 sulla piattaforma Microsoft Teams, la presenza è fortemente richiesta.

2.4.1.2. Telefonate

Le telefonate verranno utilizzate solo in casi di estrema necessità.

2.4.1.3. Chat

Verranno utilizzate due chat una per parlare esclusivamente del progetto(formale) e un'altra per le comunicazioni informali.

La chat formale è il canale di comunicazione C06 del Team di Ingegneria del Software, creato dal Top Manager. Per evitare ridondanze non sono stati creati sotto-canali.

Inoltre il tool Slack è stato inizializzato ma in pratica inutilizzato per evitare ridondanze

La chat informale è WhatsApp dove avvengo comunicazioni che non sono critiche per lo svolgimento del progetto, infatti questa chat è libera e non presenta regole di comunicazione.



2.4.2. Comunicazioni Asincrone

2.4.2.1. Kick-off Meeting Agenda

È presente l'agenda del Kick-off Meeting in essa vengono elencati gli argomenti che Verranno affrontati nel meeting, non sono state stilate delle agende per ciascun meeting ma soltanto le minute.

2.4.2.2. Minute

Durante il meeting un team member viene scelto per redigere la minuta, essa conterrà tutte le azioni e le decisioni prese durante il meeting. Nel caso un team member sia assente potrà recuperare quanto discusso nel meeting rileggendo la minuta. Il titolo della minuta deve essere Minuta-<DataMeeting>.

2.4.2.3. Trello

Trello è la bacheca virtuale dove il Project Manager comunica ai team member a quali task sono stati assegnati. I Team Member per mantenere informati il Project Manager riguardo l'avanzamento del proprio lavoro sposterà il task assegnato dalla colonna "Assegnato" alla colonna "In esecuzione", ed infine sposteranno il task sulla colonna "Svolto" nel caso in cui il task non sia stato svolto nel modo corretto altrimenti sposteranno il task di nuovo sulla colonna "In esecuzione".

2.4.2.4. Reporting

Durante le ore di GPS il Top Manager dedicherà un minuto ad ogni project manager, in questo minuto il project manager dovrà comunicare l'andamento del team.



3. Documentazione

In questa sezione saranno elencati tutti i documenti relativi al progetto UniSeats verranno elencati sia i documenti di tipo manageriale redatti dal PM, sia i documenti riguardanti lo sviluppo della progetto redatti dai TM.

Nome documento	Redatto da	Descrizione
C06_TC_Vers.x.y	PM & TM	Team Contract
C06_BC_Vers.x.y	PM	Documento che descrive gli obiettivi del progetto e una stima iniziale dei costi. Contenente la Financial Analysis
C06_PC_Vers.x.y	PM	Documento che ufficializza formalmente l'avvio del progetto e ne dà comunicazione all'organizzazione aziendale
C06_WBS_Vers.x.y	PM	Fornisce un valido ausilio al project manager soprattutto nella definizione e nell'organizzazione delle attività di progetti complessi. Contiene l'OrgChart della WBS
C06_WBS_Dict_Vers.x.y	PM	Contiene la descrizione di tutti i Work Packages della WBS
C06_SS_Vers.x.y	PM	Scope Statement, rielaborazione svolta dal Project Manager del documento Project Charter
C06_SA_Vers.x.y	PM	Lista delle attività che compongono la WBS, realizzato con Microsoft Project
C06_SPMP_Vers.x.y	PM	Documento contenente la pianificazione in dettaglio del progetto
C06_CB_Vers.x.y	PM	Strumento utile per gestire il budget allocato
C06_QMP_Vers.x.y	PM	Documento contenente le informazioni sulla qualità del progetto
C06_EVM_Vers.x.y	PM	Tecnica di Project Management che controlla scope schedule e costo di in progetto
C06_FPP_Vers.x.y	PM	Presentazione Finale del progetto
C06_FPR_Vers.x.y	PM	Documento che riassume i risultati finali del progetto, facendo un confronto tra le aspettative iniziali ed il risultato finale



C06_LLР_Vers.x.y	PM	Documento contenente le Lessons Learned del progetto
C06_RAD_Vers.x.y	TM	Documento contenente la Requirements Elicitation
C06_SDD_Vers.x.y	TM	Documento che descrive l'ambiente operativo, i sottosistemi, il database, ovvero il sistema al livello architetturale
C06_ODD_Vers.x.y	TM	Decomponere i sottosistemi in classi e packages, le interfacce e i design pattern utilizzati.
C06_MT_Vers.x.y	TM	Documento che traccia tutti i requisiti del progetto
C06_TP_Vers.x.y	TM & PM	Documento che specifica le risorse, gli obiettivi e i processi per un test specifico di un prodotto software.
C06_TCD_Vers.x.y	TM	Specifica l'obiettivo di un test case, identifica gli input e i risultati attesi (oracolo), e un flusso di eventi.
C06_TIR_Vers.x.y	TM	Report generato alla fine del processo di testing dove i problemi vengono segnalati tra i Team Members per permettere la loro risoluzione
C06_TSR_Vers.x.y	TM	Documento indirizzato agli Stakeholder contenente una panoramica dei risultati del testing
C06_MU_Vers.x.y	TM	Manuale contenente le informazioni per l'utilizzo del software da parte degli utenti finali



4. Standard, Pratiche, Convenzioni e Regole

4.1.1. Standard per la Documentazione

4.1.1.1. Standard per il Processo di Documentazione

Il processo per la realizzazione della documentazione relativa al progetto deve seguire i seguenti passi:

1. Stesura prima versione del documento;
2. Invio della versione al PM;
3. Il PM commenta le eventuali modifiche se necessarie;
4. Realizzazione versione definitiva del documento;
5. Revisione della versione definitiva da parte del reviewer;
6. Correzione al layout (laddove necessario) da parte del reviewer;
7. Controllo del documento da parte del PM;
8. Consegna del documento sulla piattaforma E-Learning;

4.1.1.2 Regole di pubblicazione dei documenti

Tutta la documentazione dovrà essere scritta in lingua italiana (esclusi diagrammi che richiedono la lingua inglese a causa del software già esistente) utilizzando il software di word processing Microsoft Word.

Ogni documento deve contenere:

- Un frontespizio in cui sia presente:
 - o Il logo del progetto;
 - o Il titolo del documento (acronimo e nome per esteso);
 - o La data dell'ultima modifica;
 - o La versione del documento;
 - o Destinatario;
 - o Chi lo ha presentato;
 - o Chi l'ha approvato.
- Una Revision history in cui si presente:
 - o Data della stesura del documento;
 - o La versione del documento;
 - o La descrizione di eventuali cambiamenti introdotti;
 - o Il nome dell'autore del documento;
- Un indice dei contenuti.

Ogni pagina deve essere dotata di:

- Un' intestazione in cui sia presente:



- o Il logo del dipartimento di informatica dell'Università di Salerno;
- o La seguente intestazione:

Laurea Magistrale in informatica- Università di Salerno

Corso di Gestione dei Progetti Software – Prof.ssa F. Ferrucci

- Un piè di pagina in cui sia presente:
 - o Il titolo del documento sulla sinistra;
 - o Il numero di pagina sulla destra.

Il nome del file deve contenere SiglaProgetto_SiglaDocumento_Vers.x.yz

4.1.1.3 Stili di Base per i documenti

Il formato dei caratteri dei documenti dovrà seguire le seguenti convenzioni:

	Font	Grandezza	Grassetto	Corsivo	Sottolineato	Colore	Allineamento
Titolo Documento	Century Gothic	48	No	No	No	Blu	Destra
Sottotitolo Documento	Garamond	20	No	No	No	Blu	Destra
Titolo Capitoli	Century Gothic	18	No	No	Si	Blu	Sinistra
Titolo Paragrafi	Garamond	13	Si	No	No	Nero	Sinistra
Sottotitoli Paragrafi	Garamond	12	Si	Si	No	Nero	Sinistra
Testo	Garamond	12	No	Se necessario	No	Nero	Giustificato
Intestazione	Garamond	12	No	No	No	Nero	Allinea al centro
Piè di pagina	Century Gothic	8	No	No	No	Blu	Destra
Intestazione Tabelle	Century Gothic	12	Si	No	No	Bianco	Allinea al centro
Contenuto Tabelle	Century Gothic	11	No	No	No	Nero	Allinea al centro
Sommario	Century Gothic	11	No	No	No	Nero	Sinistra



4.1.1.4 Dettagli colori e stili tabelle

4.1.1.4.1 Dettagli colori

Di seguito vengono descritti i dettagli dei colori utilizzati:

- Blu, colore: 1, 50% più scuro;
- Nero, colore: automatico;
- Bianco: colore: sfondo 1.

4.1.1.2.1 Stili tabelle

Di seguito vengono riportate le caratteristiche degli stili utilizzati per le tabelle:

- Tabella: griglia 5 scura – colore 1;
- Sfondo intestazione: Blu, colore: 1, 25% più scuro;
- Testo intestazione: Bianco: colore: sfondo 1.

4.2.1 Regole sugli Identificatori dei Documenti

L'assegnamento dei nomi ai documenti prodotti durante le fasi di sviluppo del software è molto importante per la tracciabilità dei documenti stessi; pertanto la sintassi di base da seguire per identificare i documenti è la seguente:

<Sigla progetto>_<acronimoDocumento>_Vers.x.yz

È importante, inoltre, mantenere la tracciabilità anche per gli artefatti; per gli artefatti il modello è:
Nome artefatto: <acronimoArtefatto>_xy.

Gli acronimi definiti per gli artefatti sono:

- RF: Requisito Funzionale;
- RNF: Requisito non Funzionale;
- US: User Story;
- SC: Scenario;
- UC: Use Case;
- UCD: Use Case Diagram;
- CD: Class Diagram;
- SCD: Statechart Diagram;
- OBJD: Object Diagram;
- SD: Sequence Diagram;
- NP: Navigational Path;
- UI: Mock-up;
- DG: Design Goal;
- TC: Test Case;
- R: Rischio.



4.2.2 Metriche per la valutazione del progetto

Non sono state applicate metriche formali per la valutazione del progetto.

4.2.3 Metriche per la valutazione della documentazione

Il PM controllerà lo stato attuale del documento e sottoporranno al reviewer il documento annotando eventuali parti da correggere.

4.2.4 Regole per Riflettere l'aggiornamento di un Documento

Ogni qualvolta viene apportata una modifica sostanziale ad un documento deve essere aggiornata la Revision History ad esso associato, specificando la data della modifica, la versione del documento, la descrizione della modifica e gli autori.

In questo modo sarà possibile mantenere una versione distinta del documento per ogni entry della cronologia di revisione; ogni versione del documento avrà la forma x.y, dove x rappresenta la versione del documento e y la sotto-versione che viene a crearsi quando la modifica al documento non è particolarmente ampia.



4.3.1 Standard per gli Artefatti

4.3.1.1 Convenzione per gli Activity Diagram

- Ogni Activity Diagram possiede uno stato iniziale denotato con un cerchio pieno nero;
- Lo stato iniziale possiede un solo arco, in uscita;
- L'activity state è indicato con un rettangolo dai bordi arrotondati e sfondo giallino;
- Le condizioni di guardia sono indicate con un rombo (giallino), con due archi uscenti (si/no);
- Le swimlane sono raggruppate in colonne, la prima per l'utente, la seconda per il sistema, la terza per il modulo di IA;
- Il control node è indicato con una barra rossa e può essere:
 - Fork: 1 arco entrante N archi uscenti
 - Join: N archi entranti 1 arco uscente
- Gli oggetti sono indicati con un rettangolo (sfondo bianco), contenente il nome dell'oggetto;
- Ogni Activity Diagram possiede uno stato finale indicato da un cerchio pieno nero con un contorno esterno rosso;
- Lo stato finale possiede un unico arco, in entrata;

4.3.1.2 Convenzione per i Requisiti Funzionali

- I requisiti funzionali avranno come acronimo RF;
- La descrizione dei requisiti funzionali seguirà il seguente template:
- RF_<numero>;
- Dove il numero è un intero che indica l'univoco requisito ;
- La descrizione della funzionalità segue la forma:
- <Il Sistema deve> <descrizione funzionalità>[Priorità: X];
- Dove X può assumere 3 valori :
 - 1: Alta;
 - 2: Media;
 - 3: Bassa;
- I requisiti funzionali vanno convertiti in User Stories;



4.3.1.3 Convenzione per le User Stories

- US_<numero>;
- Un user story può descrivere più requisiti funzionali;
- <Come> <NomeAttore> <devo> <descrizione funzionalità> <per> <motivazione>

4.3.1.4 Convenzione per i Requisiti Non Funzionali

I requisiti non funzionali sono raggruppati secondo le categorie del modello FURPS+ e avranno come acronimo RNF_<numero>. Dove numero è un intero crescente che identifica lo specifico requisito non funzionale. La priorità associata a ciascuno assume gli stessi valori denotati in precedenza

4.3.1.1 Convenzione per gli Scenari

- La tabella per ogni scenario deve essere composta da:
 - nome scenario;
 - attori;
 - descrizione;
 - flusso degli eventi;
- Il nome deve essere SC_<numero>;
- Dove il numero è un intero che identifica lo Scenario univoco;
- I partecipanti devono essere preceduti dal rispettivo nome proprio.
- Il flusso degli eventi deve iniziare con l'interazione di un partecipante.
- Il flusso degli eventi deve essere strutturato in modo da mostrare le operazioni dei partecipanti allineate a sinistra.
- Il flusso degli eventi deve essere strutturato in modo da mostrare le operazioni del sistema allineate a destra.
- Nel flusso degli eventi le operazioni dell'utente devono iniziare con il nome proprio di un partecipante.
- Nel flusso degli eventi le operazioni del sistema devono iniziare con "Il sistema..."



4.3.1.2 Convenzione per gli Use Case

- La tabella per ogni caso d'uso deve essere composta secondo il template allegato.
- Il nome del caso d'uso deve includere un verbo e deve essere univoco.
- Il nome del caso d'uso deve indicare cosa intende fare l'attore.
- L'identificativo del caso d'uso deve essere preceduto da UC_ eventualmente seguito da acronimo che denota il package e seguito da x dove x assume i valori 1,2,3...(o da altro identificativo per use case)
- Il nome dell'attore deve essere un sostantivo, che indica un ruolo rispetto all'uso del sistema.
- I nomi degli attori e dei casi d'uso devono basarsi su elementi del dominio dell'applicazione, anche i termini del flusso di eventi deve far riferimento al dominio del problema.
- Il flusso degli eventi deve iniziare con l'interazione dell'attore (triggering event)
- Un caso d'uso deve descrivere una transizione utente completa.
- La descrizione di un caso d'uso non deve superare le 2 pagine, altrimenti deve essere decomposto in casi d'uso più piccoli.
- Definire i termini importanti nel glossario.

4.3.1.1 Convenzione per gli Use Case Diagram

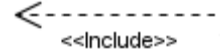
- Il diagramma dei casi d'uso deve essere composto da attori (icona umana) e casi d'uso (ovali). Gli attori devono essere collegati con una linea continua ai casi d'uso a cui partecipano. Il confine del sistema deve essere indicato tracciando un rettangolo (o package) che racchiude tutti i casi d'uso.
- Fornire i diagrammi dei casi d'uso a diversi livelli di astrazione.
- L'attore principale dovrebbe essere posizionato al lato sinistro del rettangolo.
- Sotto l'attore deve essere indicato il nome dell'attore.
- Gli attori secondari devono essere posizionati al lato destro del rettangolo.
- Prima del nome dell'attore che indica un sistema deve essere aggiunto lo stereotipo <<system>>.
- L'attore tempo deve essere denominato Time.
- Applicare <<include>> quando si conosce esattamente quando invocare il caso d'uso.
- Applicare <<extend>> quando un caso d'uso può essere invocato in varie parti del flusso di eventi del caso d'uso.



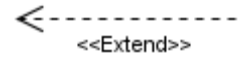
- La generalizzazione degli attori deve essere indicata con una freccia del tipo:



- L'inclusione dei casi d'uso deve essere indicata con una freccia del tipo:
indicando vicino alla freccia lo stereotipo <<include>>.



- L'estensione dei casi d'uso deve essere indicata con una freccia del tipo:
indicando vicino alla freccia lo stereotipo <<extend>>.



- Ogni Use Case Diagram è denominato:
- UC_<ACRONIMOGESTIONE>_XY: In particolare ACRONIMOGESTIONE può essere:
 - PREN: Acronimo utilizzato per indicare la Prenotazione;
 - GPREN: Acronimo utilizzato per indicare la Prenotazione (Modifica, Annullamento);
 - GPP: Acronimo utilizzato per indicare la Prenotazione di un Gruppo Privato;

4.3.1.2 Convenzione per i Class Diagram

- Ogni oggetto sarà rappresentato da un nome univoco;
- I nomi degli oggetti saranno pertinenti ad elementi del dominio del problema;
- I nomi degli attributi saranno unici all'interno di un oggetto;
- Saranno evitati tutti gli attributi e le operazioni scontate;
- Non saranno presenti oggetti che hanno comportamenti simili ma nomi diversi;
- Ogni associazione avrà come nome un verbo significativo;
- Per ogni relazione sarà indicata la molteplicità; (0, 1, *)
- Le relazioni di composizione (Freccia rombo piena) rappresenteranno un concetto del tipo "compone/è composto";
- Per eventuali generalizzazioni sarà utilizzata la notazione standard (freccia triangolare vuota);



4.3.1.1 Convenzione per gli Object Diagram

- Ogni oggetto è indicato con un rettangolo;
- Ogni oggetto è l'istanza di una classe
- Gli oggetti sono connessi con archi
- Ogni oggetto può avere la seguente label a seconda del suo tipo:
 - <<boundary>>
 - <<control>>
 - <<entity>>
- Ogni Object Diagram possiede una tabella contenente per ogni oggetto presente nel diagram:
 - Nome
 - Tipologia
 - Descrizione
- Ogni Object Diagram è denominato come:
- OBJD_<ACRONIMOPOSTO>_[numero]:. In particolare ACRONIMOPOSTO può essere
 - PS: Acronimo utilizzato per indicare la Prenotazione di un Posto Singolo;
 - PG: Acronimo utilizzato per indicare la Prenotazione posto di un Gruppo Pubblico;
 - MP: Acronimo utilizzato per indicare la Modifica della Prenotazione;
- Il numero è un intero che indica il codice identificativo del particolare Object Diagram;

4.3.1.1 Convenzione per i Sequence Diagram

- Le colonne rappresenteranno gli oggetti che partecipano al caso d'uso;
- La prima colonna corrisponderà all'attore che ha avviato il caso d'uso; Il secondo elemento nel sequence sarà un oggetto boundary;
- Il terzo elemento nel sequence sarà l'oggetto control responsabile del coordinamento dello use case;
- Ci sarà almeno un oggetto control che viene creato all'inizio del sequence diagram e si estenderà per tutta la durata dello stesso;
- In cima al diagramma si troveranno gli oggetti che esistono dapprima che il flusso abbia inizio;
- L'ordine dei messaggi ricalcherà l'ordine sequenziale con il quale vengono scambiati;
- Gli oggetti creati durante l'interazione saranno preceduti da un messaggio di <<create>>;



- Gli oggetti distrutti durante l'interazione saranno evidenziati da una croce X;
 - L'inizio della lifeline sarà allineato al punto di ricezione del corrispondente messaggio di creazione;
 - I messaggi verranno ricevuti dagli oggetti solo in presenza della lifeline;
 - La ricezione di un messaggio determinerà l'attivazione di un'operazione (con relativo box di attivazione);
 - L'inizio del box di attività sarà allineato al punto di ricezione del corrispondente messaggio;
 - La fine del box di attività sarà allineata al punto di partenza del corrispondente messaggio di ritorno;
 - L'invio di un messaggio a se stesso sarà indicato con una freccia circolare;
 - Gli oggetti Entity non richiederanno operazioni ad oggetti Boundary e Control;
 - Gli oggetti Entity saranno acceduti da oggetti Boundary e Control;
- Il sequence avrà un nome che richiama il relativo caso d'uso e sarà preceduto da SD_<numero> dove numero è un intero crescente identificativo dello specifico Sequence Diagram;

4.3.1.2 Convenzione per gli Statechart Diagram

- Il nome dello Statechart Diagram deve rispettare questo modello:
SCD_<numero>: dove numero è un intero crescente identificativo dello specifico Statechart Diagram;
- Nel diagramma sarà presente lo stato iniziale, con un unico arco, di tipo uscente;
- Lo stato iniziale sarà rappresentato con un cerchio pieno di colore nero;
- Nel diagramma saranno presenti uno o più stati generici;
- Gli stati generici saranno rappresentati con un rettangolo con gli angoli arrotondati;
- Le transizioni sono indicate con delle frecce che collegano gli stati;
- Le transizioni saranno accompagnate dall'evento che le scatuisce;
- La sintassi relative alle transizioni segue questo modello:
- Evento [guardia](argomento);
- La condizione di guardia non è sempre presente;



4.3.1.3 Convenzione per i Test Case

- I test case avranno come acronimo TC;
- La descrizione test case seguirà il seguente template:
- TC_<categoria>;
- Dove il categoria è un numero che identifica la categoria del test case e.g. TC 1.1 Registrazione;
- Test Case ID: della forma TC_<categoria>_[numero]
 - Dove numero è un intero crescente che identifica il singolo test case appartenente alla specifica categoria;
- Pre Condizione: contiene tutte le condizioni necessarie per eseguire il TC;
- Flusso di Eventi: descrive cosa accade quando il test case viene eseguito;
- Oracolo: output atteso;

4.4.1 Standard di codifica

Il Tool utilizzato è Checkstyle che utilizza le regole di stile di Google (google_checks.xml), si occuperà di controllare il codice prodotto sia conforme agli standard definiti, in caso di incongruenze segnalerà l'errore e in alcuni casi (ad esempio errori sulla spaziatura).

4.4.1 Standard per il Testing

Il Tool utilizzato è JUnit, saranno seguite le regole illustrate al corso di Ingegneria del Software e il libro di testo del corso. Inoltre è utilizzato Mockito per il testing dei DAO e Travis CI per la Continuous Integration.

5. Training

I Team Members hanno seguito il corso di Ingegneria del Software, ed i relativi tutorati.

Il Project Manager ha organizzato un breve esempio didattico per il Unit Testing di una JSP, utilizzando il Tool JUnit.



6. Gestione dei Rischi

I rischi identificati dal PM sono classificati in funzione della probabilità e dell'impatto che potrebbero avere sul progetto.

La scala di classificazione della Probabilità è la seguente:

- Bassa [10%, 30%]
- Media [30%, 60%]
- Alta [60%, 80%]

Lo schema di classificazione dell'impatto consiste di quattro misure:

- Tollerabile
- Serio
- Catastrofico

ID	Nome	Descrizione	Categoria
R_01	Mancata comprensione dello scope	Dato che non esiste un sistema preesistente a cui far riferimento, il rischio potrebbe essere la mancata comprensione dei requisiti del sistema	Processo
R_02	Mancata comprensione della teoria in funzione della quale devono essere realizzati i documenti	Gli argomenti necessari per la realizzazione dei documenti vengono apprese dai Team Members durante lo sviluppo del progetto ciò potrebbe causare ritardi nella realizzazione del progetto	Processo
R_03	Poca attenzione nella stesura del documento, il che comporta errori, anche banali e produzione di documenti di scarsa qualità	I Team Members potrebbero completare i task in modo grossolano, senza prestare attenzione alla qualità del prodotto	Persone



R_04	Le ore dichiarate non sono in linea con quelle previste	Se le ore dichiarate sono molto discordi rispetto a quanto previsto si potrebbero avere notevoli problemi di budget	Finanza
R_05	Perdita totale del lavoro prodotto	I documenti e il progetto potrebbe essere persi in un qualsiasi momento del progetto	Processo
R_06	Poca conoscenza sull'argomento di Intelligenza Artificiali	Dato l'argomento avanzato è probabile che i Team Members non riescano a destreggiarsi e ciò porterebbe ad un rallentamento del progetto.	Tecnologie
R_07	Modifica dei requisiti	In un qualsiasi momento lo sponsor potrebbe chiedere modifiche ai requisiti	Mercato
R_08	Assenza dei Team Members	I Team Members potrebbero risultare assenti in una qualsiasi fase del progetto	Persone
R_09	Abbandono dei Team Members	I Team Members potrebbero decidere di abbandonare il progetto in un qualsiasi momento	Persone
R_10	Difficoltà nella reperibilità	I Team Members potrebbero risultare non raggiungibili	Persone
R_11	Scadenze mancate	Le scadenze potrebbero non essere rispettate	Processo
R_12	Partecipazione non produttiva	I Team Members potrebbero dimostrare una partecipazione discontinua e poco significativa	Persone
R_13	Assenza di connessione Internet	I Team Members potrebbero non disporre temporaneamente di accesso ad Internet, dunque non è possibile partecipare ai meeting e condividere il lavoro	Infrastruttura

ID	Probabilità	Impatto
R_01	Media	Serio
R_02	Bassa	Catastrofico
R_03	Media	Serio
R_04	Bassa	Serio



R_05	Bassa	Catastrofico
R_06	Alta	Serio
R_07	Media	Tollerabile
R_08	Bassa	Serio
R_09	Bassa	Serio
R_10	Media	Serio
R_11	Media	Catastrofico
R_12	Media	Serio
R_13	Bassa	Catastrofico

ID	Strategia di minimizzazione	Piano di contingenza
R_01	Incontri mirati per presentare in modo dettagliato i requisiti e le caratteristiche principali del sistema	Incontri aggiuntivi per chiarire i requisiti non pienamente compresi
R_02	Non anticipare la realizzazione dei documenti in modo da fornire il tempo ai Team Members di studiare gli argomenti necessari per produrre i documenti.	Incontri aggiuntivi per approfondire gli argomenti non compresi o poco approfonditi
R_03	Template mirati a fornire una struttura di base a cui i team member possano far riferimento e che forniscano un minimo di qualità al prodotto	Controlli meticolosi e segnalazione di errori e suggerimenti su come migliorare la qualità del prodotto Utilizzo di checklist
R_04	Stima del tempo di realizzazione necessaria per il completamento di un task in funzione dei tempi di completamento dei task precedenti	Confronto con i Team Members comprendere le ragioni dei tempi prolissi per la realizzazione dei task e un riconoscimento parziale del tempo dichiarato
R_05	Backup periodici e mantenimento dei Dati su Google Drive	Ricominciare dall'inizio
R_06	Cercare di definire chiaramente il problema che si intende risolvere e perché è necessario l'utilizzo dell'Intelligenza Artificiale	Incontri mirati per discutere della tematica e come applicarla all'interno del progetto
R_07	Riflessione profonda sui requisiti in modo da prevedere la maggior parte dei problemi e delle necessità già nella fase di individuazione e analisi dei requisiti.	Analisi di quali sono le parti interessate e aggiornamento dei documenti e del sistema. Modifica al sistema se i requisiti vengono



		aggiunti in fase avanzata di sviluppo
R_08	Utilizzo di strumenti che possono essere utilizzati a distanza come Microsoft Teams. Stesura di minute in modo che tutti possano essere a conoscenza degli argomenti trattati in ogni incontro.	Incontri mediante l'utilizzo di Teams. Utilizzo di un foglio Excel con le valutazioni
R_09	Coinvolgimento dei partecipanti in modo da farli sentire parte di una squadra e spronare la partecipazione attiva	Riassegnazione dei ruoli
R_10	Accordo condiviso con i membri al fine di chiarire le responsabilità e accettare l'impegno per il progetto	Incontro face to face per capire i motivi del mancato coinvolgimento Differenze nelle valutazioni
R_11	Prevedere un margine di tolleranza tra la data di scadenza fissata dal cliente e quella prestabilita nello schedule	Redistribuzione del carico di lavoro e ridefinizione schedule
R_12	Coinvolgimento di tutti i membri del team nel progetto	Maggiore incentivo alla partecipazione
R_13	Utilizzare un canale di comunicazione tradizionale a distanza, come la rete telefonica.	Redistribuzione del carico di lavoro

7. Glossario

- **Artefatto:** Output ottenuto alla fine di un task;
- **Asincrono:** modalità di trasmissione dati che non dipende dal compiersi di altri processi;
- **Check:** Sinonimo di Controllo;
- **Continuous Integration:** Tecnica di sviluppo agile di software. Con questo metodo di integrazione gli sviluppatori integrano porzioni di codice finiti nell'applicazione anche più volte al giorno, piuttosto che integrarle tutte soltanto alla fine del progetto (Non è l'Integration Testing);
- **Codice Sorgente:** Testo di un algoritmo di un programma scritto in un linguaggio di programmazione da parte di un programmatore in fase di programmazione, compreso all'interno di un file sorgente. Esso definisce dunque il flusso di esecuzione del programma stesso;
- **Controllo:** Azione continuata diretta a disciplinare e garantire che un'attività rispetti determinati standard;
- **Criterio:** Norma su cui si fondano le distinzioni, i giudizi, le diverse linee d'azione o di condotta;
- **DAO:** nell'ambito della programmazione Web, il DAO (Data Access Object) è un pattern architetturale per la gestione della persistenza;



- **Deliverable:** oggetto materiale o immateriale realizzato (fornito/consegnato) come risultato di un'attività del progetto;
- **Difetto:** Mancata completezza, sufficienza o efficienza di qualcosa;
- **Framework:** piattaforma che funge da strato intermedio tra un sistema operativo e il software che lo utilizza;
- **Fork:** Nodo di controllo degli Activity Diagram, che divide un flusso di attività in più flussi paralleli
- **Input:** Insieme dei dati, informazioni, istruzioni, quantità di risorse o materie prime, immessi nella fase iniziale in un processo;
- **Join:** Nodo di controllo degli Activity Diagram, che sincronizza più flussi di attività in un unico flusso di attività;
- **JUnit:** Framework di Unit Testing per il linguaggio di programmazione Java;
- **Java:** Linguaggio di programmazione Object Oriented specificamente progettato per essere il più possibile indipendente dalla piattaforma hardware di esecuzione;
- **Linee guida:** Insieme di raccomandazioni sviluppate sistematicamente, sulla base di conoscenze continuamente aggiornate e valide, redatto allo scopo di rendere appropriato, e con un elevato standard di qualità, un comportamento desiderato;
- **Linguaggio di programmazione:** linguaggio formale dotato di una sintassi ben definita che viene utilizzato per scrivere programmi che realizzano algoritmi;
- **Mockito:** Framework di testing Java;
- **Output:** momento operativo o il risultato finale o l'elemento terminale di un procedimento;
- **Reviewer:** Membro del team che si occupa della revisione di un documento o artefatto;
- **Revisione:** Esame o controllo, per lo più periodico, inteso a verificare il grado dell'efficienza, della funzionalità, della corrispondenza a determinati requisiti, in quanto può implicare apporto di modifiche o di correzioni;
- **Sincrono:** Caratterizzato da coincidenza e accordo temporale di fasi;
- **Standard:** Tipo, modello, norma, cui viene uniformata una data produzione o attività;
- **Swimlane:** è un elemento grafico utilizzato nei diagrammi di flusso che suddivide visualmente le responsabilità dei sub-processi di una macro-attività;
- **Task:** Attività assegnata ad uno dei TM;



- **Template:** modello predefinito che consente di creare o inserire contenuti di diverso tipo in un documento;
- **Test:** Esperimento variamente espletato allo scopo di saggiare, mediante determinate reazioni, l'entità o la consistenza di un'attitudine o di una capacità individuale;
- **Timeline:** Rappresentazione grafica della sequenza cronologica degli eventi più significativi (che include anche i documenti più importanti);
- **Tool:** piccolo programma di ausilio per attività specifiche, in genere fornito a corredo di pacchetti software;
- **Training:** Attività volte all'insegnamento o al potenziamento di determinate conoscenze utili per lo svolgimento del progetto;
- **Travis:** Tool per la Continuous Integration;