

Quality Management Plan SharErasmus

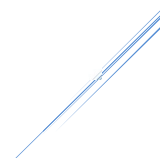
08/12/2018

Share your Erasmus experience



Riferimento	
Versione	1.0
Data	08/12/2018
Destinatario	Prof.ssa F. Ferrucci
Proposto da	Federico Vitale, Francesco Vicidomini
Approvato da	

A.A. 2018/2019



Revision History

Data	Versione	Descrizione	Autori
06/12/2018	1.0	Stesura documento	Federico Vitale, Francesco Vicidomini



Team composition

Ruolo	Nome	Posizione	Contatti
Top Manager	Filomena Ferrucci	Rappresentante del cliente	f.ferrucci@unisa.it
Project Manager	Federico Vitale	Project Manager	f.vicidomini14@studenti.unisa.it
Project Manager	Francesco Vicidomini	Project Manager	f.vitale40@studenti.unisa.it
Team Member	Alfonso Ruggiero		a.ruggiero114@studenti.unisa.it
Team Member	Davide Bottiglieri		d.bottiglieri4@studenti.unisa.it
Team Member	Francesco Breve		f.breve@studenti.unisa.it
Team Member	Giuseppe Cavaliere		g.cavaliere10@studenti.unisa.it
Team Member	Paolo Cantarella		p.cantarella1@studenti.unisa.it
Team Member	Rosaria Iorio		r.iorio11@studenti.unisa.it
Team Member	Silvio Corso		s.corso1@studenti.unisa.it
Team Member	Vincenzo Sabato		v.sabato1@studenti.unisa.it



Indice

1.	Introduzione.....	6
1.1	Scopo del documento	6
1.2	Evoluzione del documento	6
1.3	Definizioni, Acronimi e Abbreviazioni.....	6
1.3.1	Definizioni.....	6
1.3.2	Acronimi e Abbreviazioni	9
1.4	Riferimenti.....	11
2.	Struttura Gestionale.....	13
2.1	Organigramma	13
2.2	Task.....	13
2.2.1	Definizione piano di qualità	13
2.2.2	Definizione dei documenti	13
2.2.3	Elaborazione del codice	14
2.2.4	Revisione artefatto	14
2.3	Ruoli e responsabilità	14
2.4	Comunicazione	14
2.4.1	Comunicazioni Sincrone	15
2.4.1.1	Meeting.....	15
2.4.1.2	Telefonate e Skype call	15
2.4.1.3	Chat.....	15
2.4.2	Comunicazioni asincrone.....	16
2.4.2.1	Mail	16
2.4.2.2	Agenda.....	16
2.4.2.3	Minuta	16
2.4.2.4	Trello.....	16
2.4.2.5	Reporting	16
3.	Documentazione.....	17
4.	Standard, Pratiche, Convenzioni e Regole	21
4.1	Definizione di qualità	21



4.1.1 Functionality	23
4.1.2 Reliability	23
4.1.3 Usability	24
4.1.4 Efficiency.....	24
4.1.5 Maintainability.....	25
4.1.6 Portability	25
4.2 Metriche Interne.....	26
4.2.1 Tempi di risposta	26
4.2.2 Adeguatezza.....	26
4.2.3 Interoperabilità.....	27
4.3 Metriche Esterne.....	27
4.3.1 Adeguatezza.....	27
4.3.2 Sicurezza	28
4.3.3 Installabilità	28
4.4 Standard per la Documentazione.....	28
4.4.1 Standard per il Processo di Documentazione	28
4.4.2 Standard per i Documenti.....	29
4.4.2.1 Regole di Pubblicazione dei Documenti	29
4.4.2.2 Dettagli Colori	30
4.4.2.3 Stile Tabella.....	30
4.4.2.4 Frontespizio	31
4.4.2.5 Regole sugli Identificatori dei Documenti.....	31
4.4.2.6 Metriche per la valutazione del progetto.....	32
4.4.2.7 Metriche per la valutazione della documentazione	32
4.4.2.8 Regole per Riflettere i Cambiamenti tra le Versioni di un Documento	32
4.4.3 Standard per gli Artefatti.....	32
4.4.3.1 Convenzione per i Requisiti Funzionali.....	32
4.4.3.2 Convenzione per i Requisiti Non Funzionali.....	33
4.4.3.3 Convenzione per gli Scenari.....	33
4.4.3.4 Convenzione per gli Use Case.....	34
4.4.3.5 Convenzione per gli Use Case Diagram	34



4.4.3.6 Convenzione per i Class Diagram.....	35
4.4.3.7 Convenzione per i Sequence Diagram	36
4.4.3.8 Convenzione per gli Statechart Diagram	38
4.4.4 Standard di Codifica.....	38
4.4.5 Standard per il Testing.....	39
5. Revisione del Software.....	40
6. Test	42
7. Rapporto sui Problemi e Azioni Correttive	44
8. Strumenti, Tecniche e Metodologie	45
9. Controllo della Fornitura	46
10. Collezione, Manutenzione e Conservazione dei Dati.....	47
11. Training	48
12. Gestione dei Rischi	49
13. Glossario.....	56
14. Procedure di Aggiornamento del Piano.....	58



1. Introduzione

1.1 Scopo del documento

Il Quality Management Plan è il documento dove vengono 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 SharErasmus;
- 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 a garantire una alta e sempre maggiore qualità degli artefatti.

1.3 Definizioni, Acronimi e Abbreviazioni

1.3.1 Definizioni

- **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;
- **Stakeholder Register:** Documento che riporta informazioni relative ai singoli o ai gruppi che sono interessati dalla modalità di realizzazione del progetto;
- **Stakeholder Management Strategy:** Documento che raccoglie tutti coloro che sono interessati al progetto e assicura che gli stakeholder siano coinvolti in modo appropriato in tutti gli aspetti del progetto;
- **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 sottotask 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/sottotask;
- **Schedule delle Attività:** Lista, sotto forma di diagramma di Gantt 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:** 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;
- **Integration Test:** Fase di test del software in cui i singoli moduli software vengono combinati e testati come gruppo. Si verifica dopo il test di unità e prima dei validation tests;
- **Integration Test Plan:** Documento che stabilisce tutti i test di integrazione incentrati sulle funzionalità che dovranno essere eseguiti sul sistema;
- **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;
- **Unit Test Plan:** Documento che stabilisce tutti i test di unità che dovranno essere eseguiti sul

sistema;

- **Unit Test Report:** Documento che registra i dati ottenuti da ciascun test di unità in modo organizzato, descrive le condizioni ambientali o operative e mostra il confronto dei risultati del test con gli obiettivi del test;
- **Test Summary Report:** Documento che contiene un riepilogo delle attività e dei risultati finali dei test;
- **Test Execution Report:** Comunicazione inviata per stabilire le attività durante il ciclo di test. Include sia informazioni sui problemi riscontrati, sia informazioni sull'esecuzione dei test case;
- **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 installano per la prima volta il nostro sistema;
- **Manuale Utente:** 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;
- **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;
- **Percorsi di Navigazione:** Diagramma il cui scopo è quello di mostrare tutti i passaggi che bisogna compiere per poter ottenere un risultato;
- **Mock-up:** Tipicamente da metà ad alta fedeltà, sono prototipi che riflettono le scelte progettuali per schemi di colori, layout, tipografia, iconografia, grafica della navigazione e atmosfera generale del prodotto.

1.3.2 Acronimi e Abbreviazioni

- **SE_QMP_Vers.1.0:** Utilizzata per indicare il Quality Plan (in formato pdf e doc);
- **TM:** Abbreviazione utilizzata per indicare i Team Member;
- **PM:** Abbreviazione utilizzata per indicare i Project Manager;
- **QMP:** Abbreviazione utilizzata per indicare il Quality Management Plan;
- **SPMP:** Abbreviazione utilizzate per indicare il System Project Management Plan;
- **SS:** Abbreviazione utilizzata per indicare lo Scope Statement;
- **BC:** Abbreviazione utilizzata per indicare il Business Case;
- **PC:** Abbreviazione utilizzata per indicare il Project Charter;
- **TC:** Abbreviazione utilizzata per indicare il Team Contract;
- **SHREG:** Abbreviazione utilizzata per indicare lo Stakeholder Registry;
- **SHMS:** Abbreviazione utilizzata per indicare lo Stakeholder Management Strategy;



- **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;
- **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;
- **IT:** Abbreviazione utilizzata per indicare gli Integration Test;
- **ITP:** Abbreviazione utilizzata per indicare l'Integration Test Plan;
- **UT:** Abbreviazione utilizzata per indicare gli Unit Test;
- **UTP:** Abbreviazione utilizzata per indicare l'Unit Test Plan;
- **UTR:** Abbreviazione utilizzata per indicare l'Unit Test Report;
- **TSR:** Abbreviazione utilizzata per indicare il Test Summary Report;
- **TER:** Abbreviazione utilizzata per indicare il Test Execution Report;
- **TIR:** Abbreviazione utilizzata per indicare il Test Incident Report;
- **MI:** Abbreviazione utilizzata per indicare il Manuale di Installazione;
- **CS:** Abbreviazione utilizzata per indicare il Codice Sorgente;
- **SOW:** Abbreviazione utilizzata per indicare lo Statement Of Work;
- **PRU:** Abbreviazione utilizzata per indicare la Gestione Profilo Utente;
- **FRM:** Abbreviazione utilizzata per indicare la Gestione Forum;
- **CRD:** Abbreviazione utilizzata per indicare la Gestione Coordinatore;
- **CHT:** Abbreviazione utilizzata per indicare la Gestione Chat;
- **MU:** Abbreviazione utilizzata per indicare il Manuale Utente;
- **AD:** Abbreviazione utilizzata per indicare gli Activity Diagram;
- **AD_<ACRONIMOGESTIONE>_XY:** Abbreviazione utilizzata per indicare un particolare Activity Diagram. In particolare ACRONIMOGESTIONE può essere PRU, FRM, CRD, CHT. XY è un intero che indica il codice identificativo del particolare Activity Diagram;
- **RF:** Abbreviazione utilizzata per indicare i Requisiti Funzionali;
- **RF_<ACRONIMOGESTIONE>_XY:** Abbreviazione utilizzata per indicare un particolare Requisito Funzionale. In particolare ACRONIMOGESTIONE può essere PRU, FRM, CRD, CHT. XY è un intero che indica il codice identificativo del particolare Requisito Funzionale;
- **RNF:** Abbreviazione utilizzata per indicare i Requisiti Non Funzionali;
- **RNF_X:** Abbreviazione utilizzata per indicare un particolare Requisito Non Funzionale. In

particolare X è un intero che indica il codice identificativo del particolare Requisito Non Funzionale;

- **SC:** Abbreviazione utilizzata per indicare gli Scenari;
- **SC_<ACRONIMOGESTIONE>_XY:** Abbreviazione utilizzata per indicare un particolare Scenario. In particolare ACRONIMOGESTIONE può essere PRU, FRM, CRD, CHT. XY 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 PRU, FRM, CRD, CHT. XY è un intero che indica il codice identificativo del particolare Use Case;
- **UCD:** Abbreviazione utilizzata per indicare gli Use Case Diagram;
- **UCD_<ACRONIMOGESTIONE>_XY:** Abbreviazione utilizzata per indicare un particolare Use Case Diagram. In particolare ACRONIMOGESTIONE può essere PRU, FRM, CRD, CHT. XY è un intero che indica il codice identificativo del particolare Use Case Diagram;
- **CD:** Abbreviazione utilizzata per indicare i Class Diagram;
- **CD_<ACRONIMOGESTIONE>:** Abbreviazione utilizzata per indicare un particolare Class Diagram. In particolare ACRONIMOGESTIONE può essere PRU, FRM, CRD, CHT;
- **SD:** Abbreviazione utilizzata per indicare i Sequence Diagram;
- **SD_<ACRONIMOGESTIONE>_XY:** Abbreviazione utilizzata per indicare un particolare Sequence Diagram. In particolare ACRONIMOGESTIONE può essere PRU, FRM, CRD, CHT. XY è un intero che indica il codice identificativo del particolare Sequence Diagram;
- **SCD:** Abbreviazione utilizzata per indicare gli Statechart Diagram;
- **SCD_<ACRONIMOGESTIONE>:** Abbreviazione utilizzata per indicare un particolare Statechart Diagram. In particolare ACRONIMOGESTIONE può essere PRU, FRM, CRD, CHT;
- **PN:** Abbreviazione utilizzata per indicare i Path Navigazionali;
- **PN_<ACRONIMOGESTIONE>:** Abbreviazione utilizzata per indicare un particolare Navigational Path. In particolare ACRONIMOGESTIONE può essere PRU, FRM, CRD, CHT;
- **UI:** Abbreviazione utilizzata per indicare i Mock-ups;
- **UI_<ACRONIMOGESTIONE>_XY:** Abbreviazione utilizzata per indicare un particolare Mock-up. In particolare ACRONIMOGESTIONE può essere PRU, FRM, CRD, CHT. XY è un intero che indica il codice identificativo del particolare Mock-up.

1.4 Riferimenti

- Kathy Schwalbe, “Information Technology Project Management”, International Edition 7E, Cengage Learning, 2014;
- Bernd Bruegge, Allen H. Dutoit, “Object-Oriented Software Engineering Using UML, Patterns and Java”, Third Ed., Pearson, 2010;

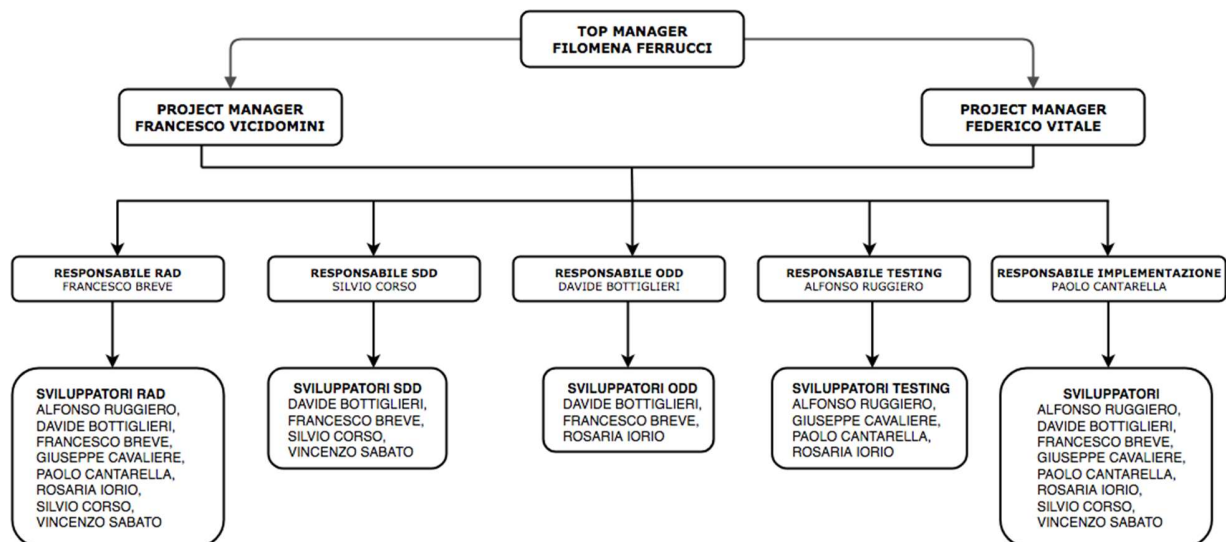


Laurea Magistrale in informatica-Università di Salerno
Corso di *Gestione dei Progetti Software*-Prof.ssa F. Ferrucci

- Sommerville, “Software Engineering”, Addison Wesley;
- PMBOK ® Guide and Software Extension to the PMBOK® Guide, Fifth Ed., Project Management Institute, 2013
- 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, check list 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: Tutti i documenti



2.2.3 Elaborazione del codice

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

Sviluppo del codice sorgente della piattaforma SharErasmus, 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

Due giorni 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: check list compilata, documento revisionato

2.3 Ruoli e responsabilità

Per ogni deliverables sono stati individuati dei responsabili, il cui ruolo è quello di revisionare il documento prodotto e compilare la check list. In questo modo si è cercato di responsabilizzare i team member. 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 anch'esso 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 member possono comunicare fra di loro e con i 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 martedì dalle 15:00 alle 17:00 e tutti i giovedì dalle 9:00 alle 11:00 la presenza è fortemente richiesta e l'assenza ingiustificata al meeting prevede una penalità di 0.50 punti sulla valutazione settimanale.

2.4.1.2 Telefonate e Skype call

Le telefonate verranno utilizzate solo in casi di estrema necessità, invece, le skype call verranno utilizzate nel caso in cui i team member o i project manager non possono recarsi sul luogo del meeting.

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 Slack che offre la possibilità di creare vari canali di comunicazione i canali che sono stati definiti sono:

- #Gestione profilo utente: dove i PM discuteranno con i team member ai quali sono stati assegnati i task relativi alla gestione profilo utente;
- #Gestione forum: dove i PM discuteranno con i team member ai quali sono stati assegnati i task relativi alla gestione forum;
- #Gestione coordinatori: dove i PM discuteranno con i team member ai quali sono stati assegnati i task relativi alla gestione coordinatori;
- #Gestione chat: dove i PM discuteranno con i team member ai quali sono stati assegnati i task relativi alla gestione chat
- #domande: dove i team member potranno porre le loro domande ai PM riguardo i task assegnato. Le domande vanno scritte in grassetto e in stampato maiuscolo così facendo sarà facile poter distinguere fra domande e risposte evitando così di porre due volte la stessa domanda
- #schedule: dove i PM pubblicheranno le varie versioni della WBS e del Gantt
- #agende: dove i PM pubblicheranno le agende dei meeting



- #minute: il team incaricato di scrivere la minuta dovrà pubblicarla su questo canale

La chat informate è Telegram 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 Mail

Le mail vengono utilizzate per comunicare la convocazione ad un meeting, ad esse è obbligatorio rispondere, in caso di mancata risposta verrà applicata una penalità di 0.25 sulla valutazione settimanale del team member che non ha risposto alla mail

2.4.2.2 Agenda

In allegato alla mail è presente l'agenda del meeting in essa vengono elencati gli argomenti che verranno affrontati nel meeting, il titolo dell'agenda è Meeting_Agenda_<data-meeting>.

2.4.2.3 Minuta

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 Meeting_Minuta_<data-meeting>

2.4.2.4 Trello

Trello è la bacheca virtuale dove i project manager comunicheranno ai team member a quali task sono stati assegnati. I team member per mantenere informati i project manager riguardo l'avanzamento del proprio lavoro potranno spostare il task assegnato dalla colonna "assegnato" alla colonna "in esecuzione" alla colonna "revisione", infine i project manager, una volta controllato il lavoro prodotto dai team member, sposteranno il task sulla colonna "completato" nel caso in cui il task sia stato svolto nel modo corretto altrimenti sposteranno il task di nuovo sulla colonna "in esecuzione".

2.4.2.5 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 SharErasmus verranno elencati sia i documenti di tipo manageriale redatti di PM, sia i documenti riguardanti lo sviluppo della progetto SharErasmus redatti dai TM.

Nome documento	Redatto da	Descrizione
Team Contract	TM & PM	Documento firmato dai PM e TM dove vengono stabilite tutte le regole progettuali
SE_RADDRAFT_Vers.x.y	TM	Prima stesura del documento di Requirement Analysis,
SE_BC_Vers.x.y	PM	Documento utile a capire i benefici che il progetto può apportare al dominio
SE_FA_Vers.x.y	PM	Analisi delle uscite e delle entrate portate dal sistema proposto
SE_PC_Vers.x.y	PM	Documento che descrive gli impegni in termini di risorse, siano esse umane, finanziarie o tecniche al raggiungimento dell'obiettivo del progetto
SE_SHMS_Vers.x.y	PM	Documento che raccoglie tutti coloro che sono interessati al progetto e assicura che gli stakeholder siano coinvolti in modo appropriato in tutti gli aspetti del progetto
SE_SHREG_Vers.x.y	PM	Documento che riporta informazioni relative ai singoli o ai gruppi che sono interessati dalla modalità di realizzazione del progetto
SE_WBS_Vers.x.y	PM	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
SE_WBSDIC_Vers.x.y	PM	Documento che descrive ciascun task e sottotask 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/sottotask;
SE_SS_Vers.x.y	PM	Rielaborazione svolta dal Project Manager del documento di avvio progetto (o project charter)
SE_SA_Vers.x.y	PM	Lista, sotto forma di diagramma di Gantt 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
Check List RAD Complete_V_x.y	TM	Lista di regole da rispettare per la stesura di un RAD di qualità
SE_MT_Vers.x.y	TM	Matrice riportante tutti gli artefatti e i loro identificativi
SE_MU_Vers.x.y	TM	Documento di comunicazione tecnica destinato a fornire assistenza alle persone che utilizzano il nostro sistema
SE_RAD_Vers.x.y	TM	Descrive il sistema in termini di requisiti funzionali e non funzionali e funge da base contrattuale tra il cliente e lo sviluppatore
SE_SPMP_Vers.x.y	PM	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)
Check List_SDD	TM	Lista di regole da rispettare per la stesura di un SDD di qualità
SE_SDD_Vers.x.y	TM	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
SE_TCS_Vers.x.y	TM	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
SE_CP_Vers. x.y	TM	Documento che specifica il Category Partition
SE_TP_Vers.x.y	TM & PM	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
Check List_ODD	TM	Lista di regole da rispettare per la stesura di un ODD di qualità
SE_QMP_Vers.x.y	PM	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
SE_ODD_Vers.x.y	TM	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
SE_TCI_Vers.x.y	TM	Fase di test del software in cui i singoli moduli software vengono combinati e testati come gruppo. Si verifica dopo il test di unità e prima dei validation tests
SE_ITP_Vers.x.y	TM & PM	Documento che stabilisce tutti i test di integrazione incentrati sulle funzionalità che dovranno essere eseguiti sul sistema
SE_TCU_Vers.x.y	TM	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
SE_UTP_Vers.x.y	TM & PM	Documento che stabilisce tutti i test di unità che dovranno essere eseguiti sul sistema
SE_UTR_Vers.x.y	TM	Documento che registra i dati ottenuti da ciascun test di unità in modo organizzato, descrive le condizioni ambientali o operative e mostra il confronto dei risultati del test con gli obiettivi del test
SE_TSR_Vers.x.y	TM	Documento che contiene un riepilogo delle attività e dei risultati finali dei test
SE_TER_Vers.x.y	TM	Comunicazione inviata per stabilire le attività durante il ciclo di test. Include sia informazioni sui problemi riscontrati, sia informazioni sull'esecuzione dei test case
SE_TIR_Vers.x.y	TM	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
SE_PMR_Vers.x.y	PM	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
SE_SR_Nr.x	PM & TM	Rapporto che riassume una situazione particolare relativa ad un determinato periodo di tempo

4. Standard, Pratiche, Convenzioni e Regole

4.1 Definizione di qualità

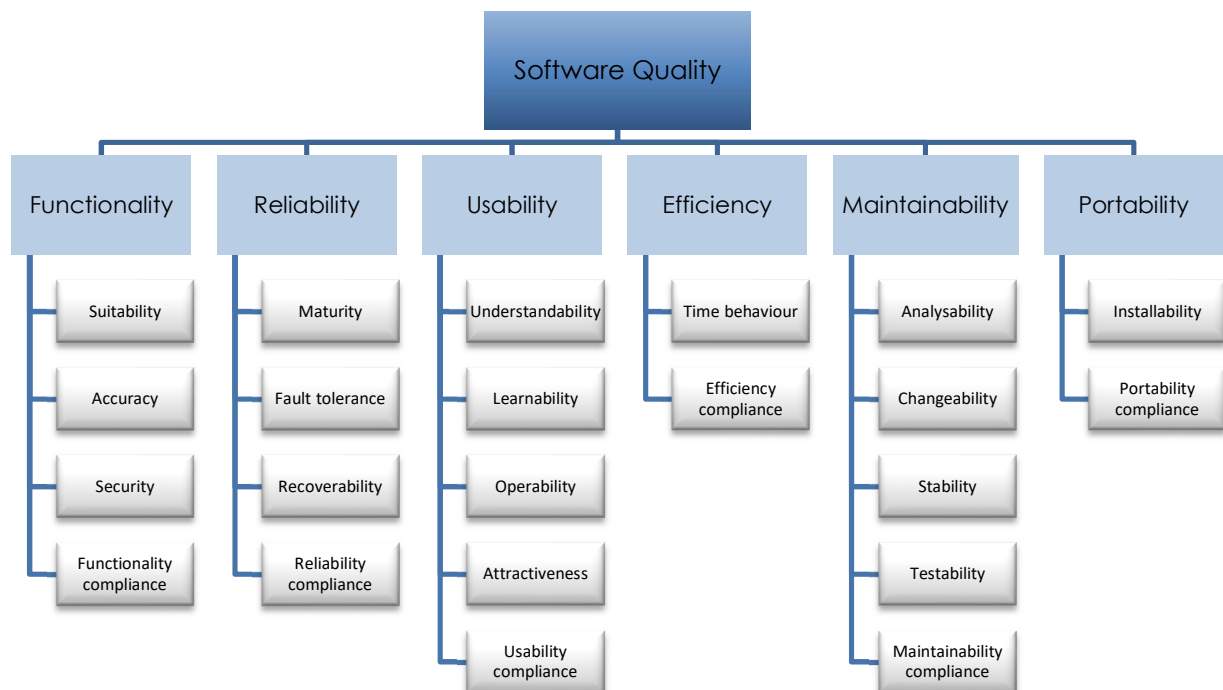
Lo standard di riferimento per la definizione del presente Quality Plan è l'ISO/IEC 9126.

Le norme ISO/IEC 9126 descrivono un modello di qualità del software, definiscono le caratteristiche che la determinano e propongono metriche per la misurazione.

Le norme relative alla qualità del software emesse da questo standard si dividono in 4 parti:

1. Modello della qualità del software;
2. Metriche esterne per la qualità;
3. Metriche interne per la qualità;
4. Metriche per la qualità in uso.

Il modello prevede che per ciascuna di esse siano associate sottocaratteristiche, dette anche attributi. Nel grafico sottostante saranno mostrate le caratteristiche e gli attributi del software proposti dal modello.





Caratteristica	Attributi	Peso	Razionale	Trade-Off
Functionality	Suitability	5		
	Accuracy	5		
	Security	5		
Reliability	Functionality Compliance	3		
	Maturity	4		
	Fault tolerance	4		
Usability	Recoverability	5		
	Reliability compliance	3		
	Understandability	4		
Efficiency	Learnability	4		
	Operability	3		
	Attractiveness	4		
Maintainability	Usability compliance	3		
	Time behaviour	4		
	Efficiency compliance	3		
Portability	Analysability	4		
	Changeability	5		
	Stability	5		
	Testability	4		
	Maintainability compliance	3		
	Installability	3		
	Portability compliance	3		



4.1.1 Functionality

La funzionalità rappresenta la capacità del software di fornire funzioni, espresse ed implicite, necessarie per operare in determinate condizioni, cioè in un determinato contesto. Gli attributi del software richiesti da questa caratteristica sono:

- **Suitability:** rappresenta la capacità di un prodotto software di fornire un appropriato insieme di funzioni che permettano agli utenti di svolgere determinati task e di raggiungere gli obiettivi prefissati;
- **Accuracy:** rappresenta la capacità di un prodotto software di fornire i risultati o gli effetti attesi con il livello di precisione richiesta;
- **Security:** rappresenta la capacità di un prodotto software di proteggere le informazioni ed i dati in modo che, persone o sistemi non autorizzati, non possono accedervi e quindi non possano leggerli o modificarli. Quanto detto si applica anche alla trasmissione dei dati.
- **Functionality compliance:** rappresenta la capacità di un prodotto software di aderire a standard, convenzioni e regolamenti di carattere legale o prescrizioni simili che abbiano attinenza con la funzionalità

4.1.2 Reliability

L'affidabilità rappresenta la capacità di un prodotto software di mantenere il livello di prestazione quando viene utilizzato in condizioni specificate. Possibili limitazioni all'affidabilità del software possono essere causate da errori nei requisiti, nella progettazione e nel codice. Le evidenze di tali errori possono essere rilevate a seconda delle condizioni in cui il prodotto è utilizzato oppure alle opzioni scelte, piuttosto che al momento in cui è utilizzato.

Gli attributi richiesti da tale caratteristica sono:

- **Maturity:** rappresenta la capacità di un prodotto software di evitare che si verifichino errori o siano prodotti risultati non corretti in fase di esecuzione;
- **Fault tolerance:** rappresenta la capacità di un prodotto software di mantenere il livello di prestazioni in caso di errori nel software o di violazione delle interfacce specificate;
- **Recoverability:** rappresenta la capacità di un prodotto software di ripristinare il livello di prestazione e recuperare i dati direttamente coinvolti in caso di errori e malfunzionamenti. A seguito di un errore o malfunzionamento, il software può risultare non accessibile per un determinato periodo di tempo; tale intervallo di tempo è valutato dalla caratteristica di recuperabilità;



- **Reliability compliance:** rappresenta la capacità di un prodotto software di aderire a standard, convenzioni e regole relative all'affidabilità;

4.1.3 Usability

L'usabilità rappresenta la capacità di un prodotto software di essere comprensibile, di poter essere studiato, di risultare attraente da parte di un utente sotto determinate condizioni. Alcuni aspetti della "funzionalità", "affidabilità" ed "efficacia" possono influire sull'usabilità del prodotto software.

Gli attributi della caratteristica usabilità sono:

- **Understandability:** rappresenta la capacità di un prodotto software di permettere all'utente di capire le sue funzionalità e come poterla utilizzare con successo per svolgere particolari task in determinate condizioni di utilizzo. Essa dipende dalla documentazione disponibile e dall'impressione iniziale che si riceve dal prodotto;
- **Learnability:** rappresenta la capacità di un prodotto software di permettere all'utente di imparare ad utilizzare l'applicazione. L'attributo corrisponde alla funzionalità relativa all'apprendimento del software;
- **Operability:** rappresenta la capacità di un prodotto software di permettere all'utente di utilizzarlo e di controllarlo. Gli aspetti relativi alla funzionalità, modificabilità, adattabilità ed installabilità del software possono influire sull'operabilità del prodotto. L'operabilità del software fa riferimento anche alle aspettative dell'utente sulla sua controllabilità, tolleranza ai guasti e conformità;
- **Attractiveness:** rappresenta la capacità di un prodotto software di risultare "attraente" per l'utente. La qualità è relativa alla progettazione dell'aspetto grafico delle sue interfacce, all'utilizzo dei colori e delle immagini, ecc.;
- **Usability compliance:** rappresenta la capacità di un prodotto software di aderire a standard, convenzioni, guida allo stile e regole relative all'usabilità.

4.1.4 Efficiency

L'efficienza rappresenta la capacità di un prodotto software di realizzare le funzioni richieste nel minor tempo possibile ed utilizzando nel miglior modo le risorse necessarie, quando opera in determinate condizioni. Le risorse includono altri prodotti software e la configurazione hardware e software del sistema.

Gli attributi relativi all'efficienza del software sono:



- **Time behaviour:** rappresenta la capacità di un sistema software di fornire appropriati tempi di risposta, tempi di elaborazione e quantità del lavoro eseguendo le funzionalità previste sotto determinate condizioni di utilizzo.
- **Efficiency compliance:** rappresenta la capacità di un prodotto software di aderire a standard e convenzioni relative all'efficienza.

4.1.5 Maintainability

La manutenibilità rappresenta la capacità di un prodotto software di essere modificato. Le modifiche possono includere correzioni o adattamenti del software a modifiche negli ambienti, nei requisiti e nelle specifiche funzionali.

Gli attributi previsti per la manutenibilità sono:

- **Analysability:** rappresenta la capacità di un prodotto software di poter effettuare la diagnosi sul software ed individuare le cause e di errori o malfunzionamenti;
- **Changeability:** rappresenta la capacità di un prodotto software di consentire lo sviluppo di modifiche al software originale. L'implementazione include modifiche al codice, alla progettazione ed alla documentazione. Nel caso in cui le modifiche debbano essere fatte dagli utenti, la modificabilità può influire sull'operabilità del prodotto;
- **Stability:** rappresenta la capacità di un prodotto software di evitare effetti non desiderati in seguito di modifiche al software;
- **Testability:** rappresenta la capacità di un prodotto software di consentire la verifica e la validazione del software modificato, cioè di eseguire i test;
- **Maintainability compliance:** rappresenta la capacità di un prodotto software di aderire a standard e convenzioni relative alla manutenibilità.

4.1.6 Portability

La portabilità rappresenta la capacità di un prodotto software di poter essere trasportato da un ambiente ad un altro. L'ambiente include aspetti organizzativi e tecnologici.

Gli attributi previsti per la portabilità sono:

- **Installability:** rappresenta la capacità di un prodotto software di essere installato in un determinato ambiente;



- **Portability compliance:** rappresenta la capacità di un prodotto software di aderire a standard e convenzioni relative alla portabilità;

4.2 Metriche Interne

Le metriche interne vengono applicate agli artefatti che verranno sviluppati durante le fasi di analisi, design e codifica. Essenzialmente le metriche interne riguardano gli sviluppatori del progetto a differenza delle metriche esterne che riguardano gli utilizzatori e i manutentori del progetto.

Le metriche che verranno applicate fanno riferimento allo standard ISO/IEC 9126-3.

4.2.1 Tempi di risposta

Riguarda i tempi di risposta del nostro sistema, potremo verificare questi risultati durante la fase di testing di sistema e testing di interfaccia, il tempo viene espresso in millisecondi. Per valutare la qualità dei tempi di risposta essi dovranno ricadere nell'intervallo descritto di seguito.

Valore dell'indice	Valutazione
< 5 ms	Efficace
=> 5 ms	Non efficace

4.2.2 Adeguatezza

Per valutare la qualità dell'adeguatezza del prodotto verrà effettuato il rapporto tra funzioni che hanno presentato problemi in fase di review(FP) e numero di funzioni controllate(FC)

$$\text{Adeguatezza} = \frac{FP}{FC}$$

L'adeguatezza viene espressa in valore assoluto, senza una unità di misura.

Valore adeguatezza	Valutazione
= 1	Completamente inadeguato
> 0.5	Abbastanza inadeguato
<= 0.5	Abbastanza adeguato
= 0	Perfettamente adeguato

4.2.3 Interoperabilità

Per valutare la qualità dell'interoperabilità delle interfacce del prodotto verrà effettuato il rapporto tra numero di interfacce confermate in fase di review(IC) e numero di interfacce da modificare(IDM)

$$\text{Interoperabilità} = \frac{IC}{IDM}$$

L'interoperabilità viene espressa in valore assoluto, senza una unità di misura.

Valore interoperabilità	Valutazione
= 1	Interfacce completamente non interoperabili
> 0.5	Basso livello di interoperabilità
<= 0.5	Buon livello di interoperabilità
= 0	Ottimo livello di interoperabilità

4.3 Metriche Esterne

Come detto in precedenza queste metriche sono rivolte agli utilizzatori finali del sistema, le metriche descritte fanno riferimento allo standard ISO 9126-2.

4.3.1 Adeguatezza

La qualità dell'adeguatezza del sistema verrà misurata dalla percentuale di funzioni risultanti insoddisfacenti durante il test o l'uso(FAULT) rispetto al numero totale di funzioni definite(T_FUNCT), anche in questo caso l'adeguatezza verrà espressa in valore assoluto.

$$\text{Adeguatezza} = \frac{FAULT}{T_FUNCT} * 100$$

Valore adeguatezza	Valutazione
= 100%	Completamente inadeguato
>=50%	Abbastanza inadeguato
< 50%	Abbastanza adeguato
= 0%	Perfettamente adeguato



4.3.2 Sicurezza

La qualità della sicurezza sarà data dalla frequenza con cui si verifica una perdita di dati. La perdita dei dati è accettabile se vengono persi dati di utenti inattivi da almeno sette anni, se un utente perde i suoi dati durante il suo periodo di attività o di inattività inferiore ai sette anni allora questa metrica di qualità non sarà rispettata.

4.3.3 Installabilità

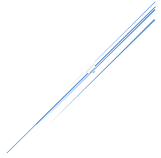
La qualità della installabilità verrà valutata in base al tempo che verrà impiegato per installare il software nell'ambiente di produzione. Assumendo che l'installazione verrà effettuata da un tecnico esperto che abbia letto il manuale di installazione, potremo dire che l'installazione è di qualità se il tecnico impiega meno di 90 minuti per installare il sistema nell'ambiente di produzione, se il tecnico impiega più di 90 minuti allora l'installabilità non avrà rispettato gli standard di qualità.

4.4 Standard per la Documentazione

4.4.1 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. Revisione della versione;
3. Modifica e aggiunta di eventuali cambiamenti individuati nella fase 2;
4. Invio della versione modificata ai PM;
5. Ricezione della versione con commenti inseriti dai PM riguardanti le correzioni da apportare al documento;
6. Ripetere i passi da 2 a 5 fin quanto non si riceve un documento privo di commenti;
7. Realizzazione versione definitiva del documento;
8. Revisione della versione definitiva da parte del reviewer;
9. Correzione al layout (laddove necessario) da parte del reviewer;
10. Sottomissione del documento su OneDrive;
11. Check dei PM al documento;
12. Consegna del documento ed eventuale materiale di supporto al documento) sulla piattaforma e-learning;



4.4.2 Standard per i Documenti

In questa sezione del documento verrà illustrato lo standard della documentazione come gli identificatori, la struttura, gli stili, i font e altro ancora.

4.4.2.1 Regole di Pubblicazione dei Documenti

La stesura dei documenti sarà guidata da template forniti dal PM. In ogni template saranno definiti la struttura e gli stili che i membri del team devono seguire per redigere il documento.

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

Ogni pagina deve essere dotata di:

- Un' intestazione in cui sia presente:
 - Il logo del dipartimento di informatica dell'Università di Salerno;
 - 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:
 - Il titolo del documento sulla sinistra;
 - Il numero di pagina sulla destra.

Il nome del file deve contenere SiglaProgetto_SiglaDocumento_Vers.x.yz



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

4.4.2.2 Dettagli Colori

Di seguito vengono descritti i dettagli dei colori utilizzati:

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

4.4.2.3 Stile Tabella

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.



4.4.2.4 Frontespizio

Ogni documento deve presentare un frontespizio standard che presenta le seguenti caratteristiche:

- Il logo del progetto;
- Il titolo del documento (acronimo e nome per esteso);
- La data dell'ultima modifica;
- La versione del documento;
- Destinatario;
- Chi lo ha presentato;
- Chi l'ha approvato.

4.4.2.5 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;
- SC: Scenario;
- UC: Use Case;
- UCD: Use Case Diagram;
- CD: Class Diagram;
- SD: Sequence Diagram;
- NP: Navigation Path;
- UI: Mock-up;
- DG: Design Goal;
- TC: Test Case;
- R: Rischio.



4.4.2.6 Metriche per la valutazione del progetto

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

4.4.2.7 Metriche per la valutazione della documentazione

Ogni documento rilasciato sarà sottoposto a doppia fase di revisione e convalida:

in primo luogo è il reviewer del documento che controllerà la qualità del documento. Una volta terminato il suo controllo sottoporrà il documento, in versione draft, ai PM.

I PM controlleranno lo stato attuale del documento e sottoporranno al reviewer il documento annotando eventuali parti da modificare e allegando la checklist che deve essere opportunamente compilata come ulteriore controllo di qualità.

Si considera superato il controllo di qualità se almeno il 75% dei parametri contenuti nelle checklist sono soddisfatti.

Sarà effettuata un'analisi quantitativa e qualitativa in base a standard dettati dal Top Manager.

4.4.2.8 Regole per Riflettere i Cambiamenti tra le Versioni 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 sottoversione che viene a crearsi quando la modifica al documento non è particolarmente ampia.

4.4.3 Standard per gli Artefatti

4.4.3.1 Convenzione per i Requisiti Funzionali

- I requisiti funzionali avranno come acronimo RF;
- La descrizione dei requisiti funzionali seguirà il seguente template:
RF_<acronimoRaggruppamento>: <nomeRaggruppamento> <Descrizione del raggruppamento>
- La descrizione della funzionalità segue la forma:
- <Il Sistema dovrà> <descrizione funzionalità>



- Per ogni requisito funzionale verrà inserita la relativa priorità.

4.4.3.2 Convenzione per i Requisiti Non Funzionali

I requisiti non funzionali sono raggruppati secondo le categorie del modello FURPS+ e avranno come acronimo RNF.

4.4.3.3 Convenzione per gli Scenari

- Ogni scenario farà riferimento ad uno e un solo requisito;
- Il nome dello scenario sarà costituito da:
SC_<acronimoGestione>_X: <nome dello scenario>, con x numero del requisito a cui fa riferimento;
- La tabella per ogni scenario sarà composta da: nome scenario, partecipanti, flusso degli eventi;
- I partecipanti saranno preceduti dal loro nome;
- Il flusso degli eventi inizierà con l'interazione di un partecipante;
- Il flusso degli eventi sarà strutturato in modo da mostrare le operazioni dei partecipanti allineate a sinistra; sarà possibile, tuttavia, avere un flusso alternato partecipante/sistema uno sotto l'altro;
- Il flusso degli eventi sarà strutturato in modo da mostrare le operazioni del sistema allineate a destra; sarà possibile, tuttavia, avere un flusso alternato partecipante/sistema uno sotto l'altro;
- Nel flusso degli eventi le operazioni dell'utente inizieranno con il nome proprio di un partecipante;
- Nel flusso degli eventi le operazioni del sistema inizieranno con "Il sistema..." o con il nome del sistema stesso;
- Il nome dello scenario sarà consistente con il contenuto;
- I riferimenti agli elementi dell'interfaccia saranno assenti;
- Tutte le funzioni del sistema descritte avranno un requisito funzionale corrispondente.

4.4.3.4 Convenzione per gli Use Case

- Il nome dello use case sarà costituito da:
UC_<acronimoGestione>_X: <nome dello Use Case>, con x numero del requisito a cui fa riferimento;
- Il nome del caso d'uso includerà un verbo e sarà univoco;
- Il nome del caso d'uso indicherà cosa intende fare l'attore;
- Il nome dell'attore sarà un sostantivo, che indica un ruolo rispetto all'uso del sistema;
- I nomi degli attori e dei casi d'uso e i termini del flusso di eventi si baseranno su elementi del dominio dell'applicazione;
- Il flusso degli eventi inizierà con l'interazione dell'attore (triggering event);
- Non sarà usata la forma passiva (le relazioni casuali tra le varie parti del flusso degli eventi saranno chiari);
- Il caso d'uso descriverà una transizione utente completa;
- Saranno descritti tutti i flussi di eventi (non solo quello principale);
- Tutti gli errori saranno descritti e trattati;
- La descrizione di ogni caso d'uso non supererà 2 pagine;
- Il caso d'uso non descriverà un'interfaccia del sistema;
- Nel caso d'uso base sarà invocato il caso d'uso incluso in un punto specifico;
- L'evento che determina l'attivazione del caso d'uso che estende sarà indicato nella condizione di ingresso del caso d'uso che estende;
- Tutti gli attori saranno coinvolti in almeno uno use case;
- Non ci saranno attori che partecipano agli stessi use case e possono essere un unico attore.

4.4.3.5 Convenzione per gli Use Case Diagram

- Il nome degli use case diagram rispetterà questo formato:
UCD_<acronimoGestione>: <nome use case diagram>;
- Nel diagramma gli attori saranno rappresentati da omini stilizzati;
- Nel diagramma i casi d'uso saranno rappresentati da ovali;
- Gli attori saranno collegati con una linea continua ai casi d'uso a cui partecipano;

- Il confine del sistema sarà indicato con un rettangolo o un package che racchiuderà tutti i casi d'uso;
- Gli attori principali saranno posizionati sul lato sinistro del rettangolo;
- Sotto l'omino che rappresenta l'attore sarà presente il suo nome;
- Gli attori secondari saranno posizionati sul lato destro del rettangolo;
- Tutti i sistemi esterni (che interagiscono con il sistema da realizzare) saranno attori;
- Non ci saranno attori che interagiscono con altri attori;
- L'include sarà applicata solo quando è opportuno e quando è esattamente noto dove viene invocato nel flusso di eventi del caso d'uso in cui è incluso;
- La generalizzazione degli attori sarà indicata con una freccia con la punta bianca;
- Il verso della freccia della generalizzazione per gli attori andrà dall'attore specializzato verso l'attore generico;
- La generalizzazione dei casi d'uso sarà indicata con una freccia con la punta bianca;
- Il verso della freccia della generalizzazione per i casi d'uso andrà dal caso d'uso specializzato verso i casi d'uso padre;
- I casi d'uso specializzati saranno posti sotto i casi d'uso padre;
- L'inclusione dei casi d'uso sarà indicata con una freccia accompagnata dallo stereotipo "<<include>>";
- Il verso della freccia dell'inclusione andrà dal caso d'uso incorporante al caso d'uso incluso;
- I casi d'uso inclusi saranno posti a destra dei casi d'uso che includono;
- L'estensione dei casi d'uso sarà indicata con una freccia accompagnata dallo stereotipo "<<extend>>";
- Il verso della freccia dell'estensione andrà dal caso d'uso che estende al caso d'uso esteso;
- I casi d'uso che estendono saranno posti sotto i casi d'uso estesi;
- Non ci saranno catene di inclusioni.

4.4.3.6 Convenzione per i Class Diagram

- Ogni oggetto sarà rappresentata da un nome univoco;
- I nomi degli oggetti saranno pertinenti ad elementi del dominio del problema;



- Per ogni oggetto sarà fornita la relativa descrizione completa, chiara, corretta e non ambigua;
- Per ogni oggetto sarà indicato il tipo (entity, boundary, control);
- Gli oggetti Control saranno nominati con verbi significativi;
- I nomi degli attributi saranno unici all'interno di un oggetto;
- Gli attributi saranno proprietà statiche dell'oggetto;
- Le operazioni degli oggetti opereranno sui dati incapsulati in essi;
- Saranno evitati tutti gli attributi e le operazioni scontate;
- La specifica della visibilità rispetta lo standard:
 - + per public
 - # per protected
 - per private;
- Non saranno presenti oggetti che hanno comportamenti simili ma nomi diversi;
- Ogni oggetto control avrà le associazioni necessarie per accedere agli oggetti che partecipano nel corrispondente use case;
- Ogni associazione avrà un nome significativo;
- Per ogni relazione sarà indicata la molteplicità;
- Non saranno indicate le relazioni derivate;
- Per eventuali aggregazioni sarà utilizzata la notazione standard (freccia a rombo vuota);
- Le relazioni di aggregazione rappresenteranno un concetto del tipo "è parte di";
- Le relazioni di composizione rappresenteranno un concetto del tipo "compone/è composto";
- Per eventuali generalizzazioni sarà utilizzata la notazione standard (freccia triangolare vuota);
- Per eventuali classi astratte sarà presente la parola chiave "abstract" tra parentesi graffe;
- Le classi astratte corrisponderanno a concetti ad alto livello;
- Le relazioni di generalizzazione rappresenteranno un concetto del tipo "è specializzato in/è generalizzato da";

4.4.3.7 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;
- 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;
- Le istanze delle classi verranno rappresentate utilizzando dei rettangoli con il nome della classe sottolineato o uno stereotipo;
- I messaggi sincroni termineranno con una freccia triangolare piena;
- I messaggi asincroni termineranno con una freccia semplice (o semifreccia);
- 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 e preceduti da un messaggio di <<destroy>>;
- L'inizio della lifeline sarà allineato al punto di ricezione del corrispondente messaggio di creazione;
- La fine della lifeline sarà allineata alla fine della linea di attività corrispondente al messaggio di distruzione;
- 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;

4.4.3.8 Convenzione per gli Statechart Diagram

- Il nome dello Statechart Diagram deve rispettare questo modello:
SCD_<acronimoGestione>: <nome dell'entità coinvolta>;
- Nel diagramma sarà presente lo stato iniziale;
- Lo stato iniziale sarà rappresentato con un cerchio colorato di nero;
- Nel diagramma saranno presenti uno o più stati generici;
- Gli stati generici saranno rappresentati con un rettangolo i cui angoli sono stondati;
- Le transazioni saranno accompagnate dall'evento che le scatuisce;
- La sintassi relative alle transazioni segue questo modello:
Evento [guardia]/azione 1; azione 2;...;azione n
- Nel diagramma sarà presente lo stato finale;
- Lo stato finale sarà rappresentato dal simbolo dello stato iniziale inscritto in un cerchio più grande a sfondo bianco;

4.4.4 Standard di Codifica

Lo standard di codifica per la struttura del codice deve essere quello definito nel file .eslintrc.json, il tool ESLint si occuperà di controllare quanto il codice prodotto sia conforme agli standard definiti, in caso di incongruenze segnerà l'errore e in alcuni casi (ad esempio errori sulla spaziatura) sarà anche in grado di correggerli automaticamente. La convenzione adottate è la seguente:

- Il codice prodotto deve essere conforme alle specifiche ECMAScript 6;
- Le variabili e i metodi vanno dichiarati usando il camelcase;
- Dopo la virgola deve seguire il carattere spazio;
- I 'require' vanno fatto all'inizio del file JavaScript;
- Le funzioni 'get' devono obbligatoriamente terminare con 'return';
- Alle variabili non deve essere assegnato il valore 'null';
- Alle variabili non deve essere assegnato il valore 'undefined';



- Per indentare bisogna usare la tabulazione, non gli spazi bianchi né tantomeno una combinazione di spazi bianchi e tabulazioni;
- Ogni istruzione deve terminare con il carattere ‘;’;
- Ogni variabile o funzione va utilizzata;
- La dichiarazione di variabili va fatta utilizzando la keyword ‘let’;

4.4.5 Standard per il Testing

Per il testing verrà integrato il tool Travis.CI al repository SharErasmus di GitHub, inoltre andrà a rispettare la pianificazione delle attività dedicate al testing, seguendo le regole illustrate a lezione di IS e del libro di riferimento del corso.



5. Revisione del Software

In questa sezione viene illustrato il piano di revisione di ogni artefatto del team.

Artefatto	Data di Revisione	Metodo di Revisione	Azioni Ulteriori
RAD Draft	10/11/2018	Lettura	Check e commenti da parte dei PM e revisione successiva, con modifiche
RAD	19/11/2018	Lettura	Check e commenti da parte dei PM e revisione successiva, con modifiche
SDD	26/11/2018	Lettura	Check e commenti da parte dei PM e revisione successiva, con modifiche
TCS	30/11/2018	Lettura	Check e commenti da parte dei PM e revisione successiva, con modifiche
ODD	02/12/2018	Lettura	Check e commenti da parte dei PM e revisione successiva, con modifiche
TCI	03/12/2018	Lettura	Check e commenti da parte dei PM e revisione successiva, con modifiche
UTR	06/01/2019	Lettura	Check e commenti da



			parte dei PM e revisione successiva, con modifiche
TSR	06/01/2019	Lettura	Check e commenti da parte dei PM e revisione successiva, con modifiche
TER	06/01/2019	Lettura	Check e commenti da parte dei PM e revisione successiva, con modifiche
TIR	06/01/2019	Lettura	Check e commenti da parte dei PM e revisione successiva, con modifiche
Codice	06/01/2019	Testing Automatico e Manuale; Prova del sito	Check e commenti da parte dei PM e revisione successiva, con modifiche

A ogni consegna si ri-effettua un controllo di tutti i documenti precedenti.



6. Test

Di seguito vengono mostrati i dettagli della WBS relativi alle varie fasi di test. Per ognuna sarà visibile l'identificativo del test, i giorni necessari per portare a compimento la fase, e le date in cui queste verranno svolte. Non vengono indicate le risorse allocate in quanto tutti i TM e PM saranno interessati da queste fasi.

52	▲ TCS	4 g	mar 27/11/18	ven 30/11/18
53	Meeting	0,25 g	mar 27/11/18	mar 27/11/18
54	Introduzione	0,25 g	mar 27/11/18	mar 27/11/18
55	Test Case Specificat	2,5 g	mar 27/11/18	gio 29/11/18
56	▲ Revisione	1 g	ven 30/11/18	ven 30/11/18
57	Revisione del Rev	0,5 g	ven 30/11/18	ven 30/11/18
58	Revisione dei PM	0,5 g	ven 30/11/18	ven 30/11/18
59	Consegna SDD & TCS	0 g	sab 01/12/18	sab 01/12/18

68	▲ TCI	2,8 g	sab 01/12/18	lun 03/12/18
69	Introduzione	0,13 g	sab 01/12/18	sab 01/12/18
70	Funzionalità da test	0,25 g	sab 01/12/18	sab 01/12/18
71	Criteri di Pass/Faile	0,13 g	sab 01/12/18	sab 01/12/18
72	Approccio	0,3 g	sab 01/12/18	sab 01/12/18
73	Test Case	1,5 g	sab 01/12/18	dom 02/12/18
74	▲ Revisione	1 g	dom 02/12/18	lun 03/12/18
75	Revisione del Rev	0,5 g	dom 02/12/18	lun 03/12/18
76	Revisione dei PM	0,5 g	lun 03/12/18	lun 03/12/18
77	Consegna ODD & TP	0 g	lun 03/12/18	lun 03/12/18



78	4 Implementazione	28,2 g?	lun 03/12/18	dom 06/01/19
79	Tutorato NodeJS	1 g	lun 03/12/18	mar 04/12/18
80	Tutorato Mocha e Chai	0,5 g	mar 04/12/18	mer 05/12/18
81	Tutorato Bootstrap	0,5 g	lun 03/12/18	mar 04/12/18
82	Tutorato Firebase	1 g	mar 04/12/18	mer 05/12/18
83	4 Gestione Profilo Personale	5 g	mer 05/12/18	lun 10/12/18
84	Coding	5 g	mer 05/12/18	lun 10/12/18
85	Testing	1 g	mer 05/12/18	gio 06/12/18
86	4 Gestione Coordinatori	10 g?	lun 10/12/18	gio 20/12/18
87	Coding	10 g?	lun 10/12/18	gio 20/12/18
88	Testing	1 g	lun 10/12/18	mar 11/12/18
89	4 Gestione Forum	10 g?	lun 10/12/18	gio 20/12/18
90	Coding	10 g?	lun 10/12/18	gio 20/12/18
91	Testing	1 g	lun 10/12/18	mar 11/12/18
92	4 Gestione Chat	10 g?	gio 20/12/18	ven 04/01/19
93	Coding	10 g?	gio 20/12/18	ven 04/01/19
94	Testing	1 g	gio 20/12/18	ven 21/12/18
95	Testing di Integrazione	0 g	dom 09/12/18	dom 09/12/18
96	Testing di Integrazione	0 g	dom 16/12/18	dom 16/12/18
97	Testing di Integrazione	0 g	dom 23/12/18	dom 23/12/18
98	Testing di Integrazione	0 g	dom 30/12/18	dom 30/12/18
99	Testing di Integrazione	0 g	dom 06/01/19	dom 06/01/19
100	Consegna del Codice	0 g	dom 06/01/19	dom 06/01/19

Si rimanda alla lettura del documento SE_TP_Vers.1.0 per maggiori dettagli.



7. Rapporto sui Problemi e Azioni Correttive

Ogni problematica o questione verrà risolta durante i meeting. Inoltre per il monitoraggio e la segnalazione è possibile attraverso o incontri individuali tra il PM e il membro del team oppure tramite l'utilizzo della chat di Slack o Telegram.

Una volta segnalata la problematica, si procede immediatamente alla sua risoluzione, in modo da non recare danni o ritardi al progetto e al clima interno del team.

L'intervento dei PM nella risoluzione di una qualunque problematica sarà sempre tempestivo. Sarà inoltre possibile, in caso di gravi criticità, la discussione con il Top Manager per la ricerca di una soluzione.



8. Strumenti, Tecniche e Metodologie

Il controllo della qualità e della produttività è un processo che avviene ogni settimana, per quel che riguarda i Team Member, con la compilazione da parte dei PM di un documento di valutazione per ogni singolo membro del team. Oltre a questo, i PM ogni settimana compileranno una valutazione generale dell'andamento del progetto nella sua interezza.

Per quanto riguarda il controllo della qualità rispetto ai deliverables, si procederà, ogni qual volta un artefatto sarà pronto, ad effettuarne un controllo preliminare da parte dei TM (e in caso di necessità o delucidazioni, da parte dei PM). Una volta che tutti gli artefatti che compongono un documento saranno pronti e il documento verrà strutturato e messo insieme, si procederà con il controllo da parte del reviewer (scelto a rotazione tra i membri del team) che, con il supporto delle checklist, procederà alla correzione del documento. Il reviewer potrà interrogare i TM che si sono occupati di un particolare artefatto in caso di delucidazioni e/o correzioni. Una volta compiuta questa operazione il documento revisionato verrà mandato ai PM tramite OneDrive che procederanno alla correzione. A questo punto i PM commenteranno il documento utilizzando le funzionalità offerte da Word Online. I TM, osservando i commenti lasciati dai PM, effettueranno le dovute correzioni necessarie. Questa operazione verrà iterata finché il documento non soddisferà i requisiti di qualità necessari. A quel punto i PM daranno la loro accettazione e invieranno il documento al Top Manager tramite la piattaforma di e-learning.



9. Controllo della Fornitura

La qualità per quanto riguarda la realizzazione e interazione con la funzionalità chat è assicurata dall'utilizzo del tool Firebase, piattaforma di Google che garantisce scalabilità, funzionamento 24h su 24, 365 giorni all'anno, tempi di risposta pressoché inesistenti e sicurezza assoluta.

Amazon AWS, sono i servizi Cloud offerti da Amazon (noi utilizzeremo i servizi EC2 e RDS) sono di qualità perché abbassano notevolmente i costi di gestione delle macchine remote dato che la manutenzione e l'alimentazione viene gestita da Amazon, inoltre le macchine che mette a disposizione AWS sono reperibili 24h su 24, 365 giorni l'anno. A rendere AWS uno strumento di qualità è anche il fatto che esso viene ampiamente utilizzato da molte aziende multinazionali e non.

Node.js è una piattaforma Open source event-driven per l'esecuzione di codice JavaScript lato server, esso risulta uno strumento di qualità in quanto permette, in poche istruzioni, di realizzare server molto ben strutturato ed efficienti. La piattaforma risulta molto modulare ed è molto facile aggiungere librerie esterne che possono semplificare molto il lavoro degli sviluppatori. Node.js (come molte delle sue librerie esterne) è dotato anche di una documentazione molto dettagliata che può facilitare ulteriormente il lavoro di sviluppatori meno esperti.

NPM è il package manager di Node.js esso si occupa dell'aggiunta di librerie esterne, la sua facilità di utilizzo lo rende uno strumento molto efficiente e rapido da imparare questo lo rende uno strumento di qualità.

Bootstrap è un framework HTML che permette di strutturare in maniera molto rapida una pagina HTML, esso mette a disposizione anche molti template se gli sviluppatori si attengono quanto più è possibile agli standard del template essi realizzeranno facilmente una applicazione usabile e responsive.

GitHubPages è un servizio offerto da GitHub.com la sua qualità è data dalla sua facilità di utilizzo dato che per caricare le pagine web sul web host non ci sarà bisogno di caricarle tramite il protocollo FTP ma basterà effettuare una semplice commit, inoltre, con questo strumento non avremo costi di gestione e manutenzione riguardanti il web host.



10. Collezione, Manutenzione e Conservazione dei Dati

Come strumento per la raccolta e conservazione dei dati viene usata una cartella condivisa su Dropbox. La cartella condivisa è organizzata secondo un'infrastruttura di cartelle che raggruppano informazioni relative ai documenti, alle minute, alle valutazioni, materiale utile, materiale relativo all'implementazione. All'interno di ogni cartella vengono conservate anche le diverse versioni dello stesso documento.

Inoltre è disponibile sul servizio di hosting per progetti GitHub un repository che contiene tutti i deliverables del progetto, documentazione e codice inclusi.



11. Training

Sono state pianificate e quindi allocate alcune ore per il training del team. Queste ore saranno utili per colmare le lacune del team rispetto a Tool, Framework e Linguaggi di Programmazione che utilizzeremo durante il progetto. Di seguito sono elencati tutte i tutorati pianificati:

- Tutorato sull'utilizzo del tool MarvelApp;
- Tutorato sull'utilizzo del tool Draw.io;
- Tutorato su NodeJS;
- Tutorato su Mocha;
- Tutorato su Chai;
- Tutorato su JQuery;
- Tutorato su Bootstrap.

Inoltre, i TM hanno seguito attivamente il corso di Ingegneria del Software.



12. Gestione dei Rischi

I rischi identificati dai PM sono classificati in funzione della probabilità e dell'impatto che avrebbero sul progetto.

La scala di classificazione della Probabilità è la seguente:

- Molto bassa [0%, 19%]
- Bassa [20%, 39%]
- Media [40%, 59%]
- Alta [60%, 79%]
- Molto Alta [80%, 100%]

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

- Trascurabile
- 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 membri del team durante lo sviluppo del progetto e potrebbe essere necessario un tempo supplementare per l'apprendimento degli argomenti trattati che causerebbe ritardo nella realizzazione del progetto.	Processo
R_03	Poca attenzione nella stesura del documento, il che comporta errori, anche banali e produzione di	I membri del team potrebbero completare i task in modo grossolano, senza	Persone



	documenti di scarsa qualità.	prestare attenzione alla qualità del prodotto.	
R_04	Le ore dichiarate sono discordi da quelle previste.	Le ore sono in nostro budget e se le ore dichiarate sono molto discordi rispetto a quanto previsto potremo avere notevoli problemi di budget.	Finanza
R_05	Carico di lavoro ingente e tempi di consegna ristretti.	La quantità di lavoro da realizzare è ingente e il tempo per completare non lascia molto margine di errore dato che le scadenze sono molto ravvicinate.	Processo
R_06	Mancata comprensione degli argomenti trattati dai tutorati dei PM.	La mancanza di attenzione o di interesse nel progetto potrebbe comportare una mancata comprensione degli argomenti trattati dai PM.	Persone
R_07	Mancata comprensione del flusso di esecuzione del sistema.	Data la particolarità del progetto, il rischio che i membri del team non abbiamo ben chiaro in mente il flusso di esecuzione del sistema è molto alto.	Processo
R_08	Mancanza di conoscenza degli strumenti di implementazione.	Le tecnologie scelte quali, ad esempio, Bootstrap, potrebbero non essere note ai team member rallentando i tempi di realizzazione a causa della necessità di un training preventivo.	Tecnologie
R_09	Mancanza di precisione che potrebbe portare a incoerenze sostanziali nella realizzazione del prodotto.	I membri del team potrebbero mancare di precisione e consegnare prodotti di scarsa qualità.	Persone
R_10	Perdita totale del lavoro prodotto.	I documenti e il progetto potrebbe essere persi in un qualsiasi momento del progetto.	Processo



R_11	Cancellazione di Account o profili gratuiti	Visto l'utilizzo di servizi come Educate che ci permettono di sfruttare tecnologie solitamente a pagamento, in modo del tutto gratuito, o in generale visto l'utilizzo di piani gratuiti (ad esempio per la chat o il server) potrebbe succedere che per qualunque motivazione tali account o tali piani vengano soppressi o chiusi	Tecnologie
R_12	Tecnologie scelte non adeguate allo sviluppo del software.	Le tecnologie scelte potrebbero non essere note ai team member e non essere adatte alla realizzazione del progetto.	Tecnologie
R_13	Tempi prolungati per l'apprendimento delle tecnologie scelte per l'implementazione.	La mancanza di conoscenza degli strumenti di implementazione potrebbe ritardare i tempi di completamento del progetto.	Persone
R_14	Aggiunta requisiti.	In un qualsiasi momento lo sponsor potrebbe chiedere modifiche ai requisiti.	Mercato
R_15	Assenza dei membri del team.	I membri del team potrebbero risultare assenti in una qualsiasi fase del progetto.	Persone
R_16	Abbandono membri del team.	I membri del team potrebbero decidere di abbandonare il progetto in un qualsiasi momento.	Persone
R_17	Impegno squilibrato.	I membri del team potrebbero non essere pienamente dedicati al progetto causando uno squilibrio dell'impegno rispetto agli altri membri.	Persone
R_18	Difficoltà nella reperibilità.	I membri del team potrebbero risultare non	Persone



		raggiungibili, soprattutto nel periodo di Natale.	
R_19	Scadenze mancate.	I membri del team potrebbero non rispettare le scadenze.	Processo
R_20	Poca partecipazione o partecipazione non produttiva.	I membri del team potrebbero dimostrare una partecipazione discontinua e poco significativa.	Persone
R_21	Assenza di luoghi, all'interno dell'università, dove effettuare meeting	A causa del fatto che è sempre molto complesso trovare aule libere negli orari curricolari, potrebbe richiedere un gran dispendio di tempo la ricerca di un luogo dove incontrarsi per effettuare meeting o sedute di lavoro	Infrastruttura

ID	PROBABILITÀ	IMPATTO
R_01	Media	Serio
R_02	Bassa	Catastrofico
R_03	Media	Serio
R_04	Bassa	Serio
R_05	Alto	Tollerabile
R_06	Media	Tollerabile
R_07	Media	Serio
R_08	Alto	Tollerabile
R_09	Media	Tollerabile
R_10	Bassa	Catastrofico
R_11	Media	Serio
R_12	Bassa	Serio



R_13	Alta	Tollerabile
R_14	Molto Alto	Tollerabile
R_15	Media	Serio
R_16	Bassa	Serio
R_17	Alta	Tollerabile
R_18	Alta	Serio
R_19	Media	Catastrofico
R_20	Media	Serio
R_21	Alta	Serio

ID_RISCHIO	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, incontri che potrebbero coinvolgere anche lo sponsor.
R_02	Non anticipare la realizzazione dei documenti in modo da fornire il tempo ai membri del team per studiare gli argomenti in funzione dei quali produrre i documenti.	Tutorato 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. Nel caso fosse necessario Tutorato per spiegare come applicare gli standard decisi. 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 member per capire le ragioni dei tempi prolissi per la realizzazione dei task e un riconoscimento parziale del tempo dichiarato.



R_05	Schedule dettagliato e aggiornato ogni volta che si presenta una nuova esigenza. Ripartizione equa del carico.	Ridefinizione dello schedule e assegnazione di nuovi task per soddisfare le nuove esigenze.
R_06	Presentazione degli argomenti fornendo esempi esplicativi e nel modo più chiaro possibile mostrando solo aspetti dell'argomento veramente utili per il progetto.	Ulteriori tutorati o altri esempi per presentare gli strumenti o le tecnologie da usare.
R_07	Spiegazione dettagliata del flusso di esecuzione del sistema e realizzazione guidata delle parti dei documenti di particolare interesse per la riuscita del progetto. Discussione prima di ogni decisione.	Spiegazioni aggiuntive.
R_08	Spiegazione mirata sui tool e linguaggi utilizzati durante il progetto. Fornitura di esempi esplicativi e di esercizi da svolgere atti all'apprendimento.	Ulteriori tutorati e spiegazioni aggiuntive. Aiuto in un semplice caso di utilizzo del sistema.
R_09	Monitoraggio e suggerimenti relativi a come migliorare la qualità dei prodotti.	Controllo del prodotto e correzioni.
R_10	Backup periodici e mantenimento dei dati su due server distinti (Dropbox e Slack)	Ricominciare da capo.
R_11	Mantenimento di backup su servizi analoghi.	Ricreazione degli account con l'utilizzo di altre email o utilizzo degli account prestati dai propri colleghi.
R_12	Documentarsi adeguatamente sulle funzionalità offerte dalle tecnologie scelte. Affidarsi a strumenti software di cui è nota l'efficacia.	Scelte di nuove tecnologie.
R_13	Sessioni di training intensive volte all'apprendimento di strumenti di sviluppo.	Scelta di nuove tecnologie.
R_14	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_15	Utilizzo di strumenti che possono essere utilizzati a distanza e un numero di incontri elevato in modo da evitare che	Incontri mediante l'utilizzo di Skype. I task vengono distribuiti in modo tale



	la mancata partecipazione a un numero ristretto di incontri possa compromettere la riuscita del progetto. Stesura di minute in modo che tutti possano essere a conoscenza degli argomenti trattati in ogni incontro.	che nessuno è indispensabile in ogni fase. Se la presenza è fondamentale e l'assenza è ingiustificata vi saranno penalità nelle valutazioni.
R_16	Coinvolgimento dei partecipanti in modo da farli sentire parte di una squadra e spronare la partecipazione attiva.	Riassegnazione dei ruoli.
R_17	Incentivo per una equa ripartizione del lavoro e responsabilizzazione di ogni membro.	Differenze nelle valutazioni.
R_18	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_19	Prevedere un margine di tolleranza tra la data di scadenza fissata dal cliente e quella prestabilita nello schedule.	Ridistribuzione del carico di lavoro e ridefinizione schedule.
R_20	Coinvolgimento di tutti i membri del team nel progetto.	Maggiore incentivo alla partecipazione.
R_21	Sfruttare al massimo luoghi anche non adatti agli incontri. Utilizzo di strumenti che non richiedono incontri face-to-face.	Richiesta al Top Manager di un luogo adibito ai meeting.



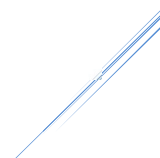
13. Glossario

- **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;
- **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;
- **Coordinatori:** Membro del corpo docente che si occupa di essere un referente per lo studente che si appresta a svolgere una attività di Erasmus o Erasmus+ Traineeship. Vi saranno un coordinatore dell'università di partenza e uno dell'università/azienda di arrivo;
- **Criterio:** Norma su cui si fondano le distinzioni, i giudizi, le diverse linee d'azione o di condotta;
- **Deliverable:** oggetto materiale o immateriale realizzato (fornito/consegnato) come risultato di un'attività del progetto;
- **Difetto:** Mancata completezza, sufficienza o efficienza di qualcosa;
- **Erasmus:** Attività offerta dalle università che prevede un soggiorno di studio all'estero da parte dello studente che ne fa richiesta e ha come condizione di riferimento un precedente accordo stipulato tra due determinate università appartenenti a paesi aderenti all'Unione Europea;
- **Erasmus+ Traineeship:** Al contrario dell'Erasmus, l'Erasmus for Traineeship è volto allo svolgimento di tirocini presso imprese, centri di formazione e ricerca, istituti di istruzione superiore localizzati in uno dei paesi partecipanti al programma;
- **Framework:** piattaforma che funge da strato intermedio tra un sistema operativo e il software che lo utilizza;
- **Input:** Insieme dei dati, informazioni, istruzioni, quantità di risorse o materie prime, immessi nella fase iniziale in un processo;
- **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.
- **Mobilità Internazionali:** possibilità offerta a studenti, ricercatori e professori di spostarsi presso altre istituzioni, all'interno del proprio paese o all'estero, per un periodo più o meno limitato di tempo, al fine di attendervi lo studio, l'insegnamento, o la ricerca;
- **Normativa:** Il complesso di norme relative a una data attività;
- **Output:** momento operativo o il risultato finale o l'elemento terminale di un procedimento;
- **Produttività:** L'attitudine a conseguire un risultato superiore ai mezzi impiegati;
- **Qualità:** Nozione alla quale sono ricondotti gli aspetti del progetto suscettibili di



classificazione o di giudizio;

- **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;
- **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) che si verificano prima di partire, durante e dopo l'Erasmus;
- **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;



14. Procedure di Aggiornamento del Piano

Questo documento sarà aggiornato ogni qualvolta sarà necessario.