



# Software Project 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	



## Revision History

Data	Versione	Descrizione	Autori
04/01/2021	1.0	Stesura documento	V. Russo
19/01/2021	1.1	Aggiornamento (Organigramma, Travis, Mockito)	V. Russo

## Sommario

Introduzione	5
1.1. Project review	5
1.1.1. Scopo, Ambito e Obiettivi	5
1.1.2. Assunzioni e vincoli	5
1.1.3. Project deliverables	6
1.1.4. Schedule and budget summary	7
1.2. Evolution of the plan	7
2. Riferimenti	7
3. Definizioni	7
4. Contesto del Progetto	9
4.1. Modello di Processo	9
4.2. Process improvement plan	10
4.3. Infrastructure plan	10
4.4. Piano di Accettazione del Prodotto	12
4.5. Project organization	12
5. Pianificazione del Progetto	13
5.1. Project initiation	13
5.1.1. Piano di Stima	13
5.1.2. Piano di Organizzazione dello Staff	14
5.1.3. Resource acquisition plan	14
5.1.4. Project staff training plan	14
5.2. Project work plans	14
5.2.1. Work activities	14
5.2.2. Schedule allocation	14
5.2.3. Resource allocation	14
5.2.4. Budget allocation	14
5.2.5. Procurement plan	14
6. Controllo e Valutazione del Progetto	14
6.1. Piano di Gestione dei Requisiti	14
6.2. Pianificazione del cambiamento dello Scope del Progetto	15
6.3. Piano di Controllo Schedule	15
6.4. Budget control plan	15
6.5. Quality Assurance plan	15
6.6. Subcontractor management plan	15
6.7. Project closeout plan	15



7.	Product delivery	15
8.	Supporting Process Plans	15
8.1.	Project supervision and work environment	16
8.2.	Decision management	16
8.3.	Risk management	16
8.4.	Configuration management	19
8.5.	Information management	19
8.6.	Quality Assurance	19
8.7.	Measurement	19
8.8.	Reviews and audits	19
8.9.	Verification and Validation	19
9.	Additional plans	21
1.1.	Glossario	21
1.1.1.	Definizioni	21

## Introduzione

### 1.1. Project review

#### 1.1.1. *Scopo, Ambito e Obiettivi*

UniSeats ha come scopo quello di offrire un servizio di prenotazione di un posto in aula studio usufruendo di un sistema software esterno di Intelligenza Artificiale per gestire la disposizione dei posti in aula. L'uso di tale software esterno è ritenuto necessario per ottimizzare la capienza delle aule tenendo conto dei vincoli imposti dalle normative circa il distanziamento sociale, evitando così lo spreco di spazi inutilizzati. Gli studenti potranno dunque prenotare il proprio posto in aula per studiare in tranquillità usufruendo degli spazi messi a disposizione dall'università e al contempo l'università potrà evitare il sovraffollamento delle aule, e nel caso particolare di questa situazione pandemica potrà anche tenere traccia degli studenti che frequentano l'aula studio e ottimizzare gli spazi grazie all'uso del software esterno di Intelligenza Artificiale. L'Università degli Studi di Salerno potrà in ogni modo decidere di interrompere l'utilizzo del sistema esterno a fine pandemia, mantenendo le funzionalità offerte dal sistema UniSeats circa la prenotazione del posto.

L'ambito del sistema è di facilitare la gestione delle aule studio e di supportare gli studenti per la prenotazione del posto in aula studio assicurando loro un luogo in cui studiare in tranquillità. Il sistema sviluppato sarà destinato all'Università degli Studi di Salerno. Esso in particolare deve supportare la richiesta di prenotazione di un posto a sedere in un'aula studio, sia singola che per un gruppo, la modifica o annullamento di una prenotazione effettuata. Il sistema non supporta il controllo dell'effettiva iscrizione all'università da parte dell'utente che presenta la richiesta. Il controllo della coincidenza della matricola presentata dall'utente in fase di registrazione con quella assegnatagli dall'università.

Gli obiettivi primari del progetto sono Fornire uno strumento per il supporto della gestione delle aule studio; con relativo tracciamento degli studenti. Ottimizzare gli spazi e i posti disponibili rispettando tutte le normative anti-Covid19, eliminare gli attuali disagi agli stakeholder.

#### 1.1.2. *Assunzioni e vincoli*

Insieme di assunzioni e vincoli riguardanti lo scope, lo schedule, il budget, le risorse, il software da riusare, il software da incorporare, tecnologie da impiegare, interfacce con sistemi esterni, ambiente di deployment, ecc.

Il sistema non supporta il controllo dell'effettiva iscrizione all'università da parte dell'utente che presenta la richiesta. Il controllo della coincidenza della matricola presentata dall'utente in fase di registrazione con quella assegnatagli dall'università.

### ***1.1.3. Project deliverables***

#### **Project management-related deliverables:**

1. Business Case;
2. Project Charter;
3. Team Contract;
4. Scope Statement;
5. Work Breakdown Structure;
6. WBS Dictionary;
7. Schedule delle Attività;
8. Software Project Management Plan;
9. Cost Baseline;
10. Status Report;
11. Quality Management Plan;
12. Final Project Presentation;
13. Final Project Report;
14. Lessons-learned Report;

#### **Product-related deliverables:**

1. RAD
2. SDD;
3. ODD;
4. Matrice di Tracciabilità;
5. Test Plan;
6. Test Case Specification (Contenuta nel Test Case Document);
7. Test Incident Report;
8. Test Summary Report;
9. Manuale D'Uso;
10. Manuale di Installazione;
11. Codice Sorgente;

Come work-product interni prevediamo:

- al termine del meeting verrà redatta una minuta in modo da documentare gli argomenti effettivamente trattati e eventuali decisioni prese o per riportare l'andamento dei task (se sono assegnati, in corso, completati o validati);
- ogni settimana verrà redatto un documento di valutazione per ognuno dei Team Member e per il progetto nella sua interezza con annotazioni, così da enfatizzare eventuali criticità da tenere sotto controllo.

#### 1.1.4. *Schedule and budget summary*

Una stima preliminare del costo dell'intero progetto è di 10.250€. Questa stima tiene conto delle ore di lavoro per ogni singolo membro (con una paga oraria di 25€ per ogni team member e 35€/h per ogni PM). Le ore di lavoro disponibili per ognuno dei membri del team (PM e membri del team) è di 50 ore totali. Il costo iniziale include anche 1000€ di spese generiche (pubblicità, server, ecc.). I benefici vengono calcolati considerando che lo studente medio paga annualmente €700 di tasse moltiplicato per il numero stimato di 14 nuovi studenti iscritti ogni anno, portati grazie alla piattaforma. Il ritorno con le spese è stimato nel primo anno.

- NPV €18.169;
- ROI 181%;

Si faccia riferimento al documento C06\_SA.pdf per quanto riguarda lo Schedule delle Attività e al C06\_BC\_1.1.pdf per quanto riguarda l'analisi completa del budget.

### 1.2. Evolution of the plan

Il PMP viene aggiornato periodicamente ogni 15 giorni

## 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:
  - C06\_SA\_Vers.1.1.mpp
  - C06\_BC\_Vers.1.1.pdf
  - C06\_SOW\_Vers.1.1.pdf
  - C06\_RAD.pdf
  - C06\_SDD.pdf
  - C06\_WBS\_Vers.1.1.pdf
  - C06\_WBS\_Dict\_Vers.1.1.pdf

## 3. 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;
- **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; tutte le definizioni e i termini usati nel seguito per un'adeguata comprensione del PMP.
- **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);
- **Status Report:** Rapporto che riassume una situazione particolare relativa ad un determinato periodo di tempo;
- **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;
- **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 Case Specification:** Specifica un test case, identificando gli input richiesti e i risultati attesi, fornisce procedure dettagliate per l'esecuzione del test e delinea i criteri di superamento/fallimento per determinare l'accettazione.
- **Test Summary Report:** Documento che contiene un riepilogo delle attività e dei risultati finali dei test;
- **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 intendono installare il nostro sistema;
- **Manuale D'Uso:** Documento di comunicazione tecnica destinato a fornire assistenza alle persone che intendono utilizzare 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.

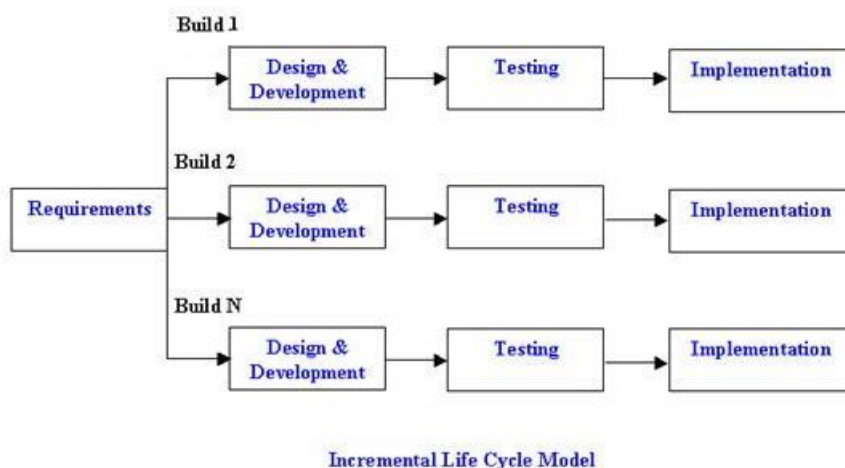
## 4. Contesto del Progetto

### 4.1. Modello di Processo

Il modello di sviluppo scelto per questo progetto è quello incrementale, ogni modulo passa attraverso le fasi di requisiti, progettazione, implementazione e test. Una versione funzionante del software viene prodotta durante il primo modulo, in modo da avere il software funzionante nelle prime fasi durante il ciclo di vita del software. Ogni versione successiva del modulo aggiunge funzioni alla versione precedente. Il processo continua fino al raggiungimento del sistema completo.



Nel diagramma sopra, quando lavoriamo in modo incrementale, stiamo aggiungendo pezzo per pezzo, ma ci aspettiamo che ogni pezzo sia completamente finito. Quindi continuiamo ad aggiungere i pezzi finché non è completo.



#### Vantaggi del modello incrementale:

- Si ottiene rapidamente software funzionante.
- Più flessibile, meno costoso per modificare lo scope e i requisiti.
- Più facile testare ed eseguire il debug durante un'iterazione più piccola.
- In questo modello il cliente può esaminare ogni release.
- Riduce il costo di consegna iniziale.
- I rischi vengono identificati e gestiti durante l'iterazione.

#### Svantaggi del modello incrementale:

- Ha bisogno di una buona pianificazione e design.
- Ha bisogno di una definizione chiara e completa dell'intero sistema prima che possa essere scomposto e costruito in modo incrementale.

#### 4.2. Process improvement plan

Problema	Descrizione Problema	Risoluzione
Difficoltà modellazione del Modulo IA all'interno del Sistema	Essendo l'utilizzo del modulo d'IA un approccio sperimentale nello sviluppo del sistema, i Team Members hanno incontrato difficoltà nella modellazione	Il Top Management e il Project Manager hanno affiancato i Team Members
Difficoltà nella gestione dei diversi corsi seguiti	A causa di prove intercorso o altri esami i Team Members hanno dovuto bilanciare il loro tempo	<ul style="list-style-type: none"> <li>• Posticipare le consegne</li> <li>• Riduzione carico di lavoro</li> </ul>

#### 4.3. Infrastructure plan

Si riporta nella seguente tabella gli strumenti utilizzati:

Tool	Descrizione	Motivazione
Microsoft Project	Strumento utilizzato dai PM per la creazione e definizione degli schedule delle attività e per i diagrammi di Gantt	<ul style="list-style-type: none"> <li>• Usabilità superiore rispetto ad altri strumenti analoghi</li> <li>• Fornito gratuitamente dall'università</li> <li>• Il più utilizzato all'interno delle aziende</li> </ul>
Microsoft Teams	Strumento utilizzato per la comunicazione e lo svolgimento dei meeting	<ul style="list-style-type: none"> <li>• Utilizzato dall'università</li> <li>• Utilizzato da tutti i membri</li> </ul>
Microsoft Office	Strumento utilizzato per redirigere i documenti e preparare le presentazioni	<ul style="list-style-type: none"> <li>• Usabilità superiore rispetto ad altri strumenti analoghi</li> <li>• Fornito gratuitamente dall'università</li> <li>• Il più utilizzato all'interno delle aziende</li> </ul>
Google Drive	Servizio web, in ambiente cloud, di memorizzazione e sincronizzazione di file	<ul style="list-style-type: none"> <li>• Standard</li> <li>• Utilizzato dai Team Members</li> </ul>
Git	Strumento utilizzato per il versioning	<ul style="list-style-type: none"> <li>• Uno dei più diffusi</li> </ul>
GitHub	Strumento utilizzato per l'hosting del codice e la	<ul style="list-style-type: none"> <li>• Uno dei più diffusi</li> </ul>

	documentazione del progetto	
Trello	Strumento per l'assegnazione e il monitoraggio dei task assegnati	<ul style="list-style-type: none"> <li>• Richiesto</li> </ul>
Travis CI	Strumento per la Continuous Integration	<ul style="list-style-type: none"> <li>• Facilmente utilizzabile</li> <li>• Richiesto</li> </ul>
Slack	Strumento di comunicazione	<ul style="list-style-type: none"> <li>• Richiesto</li> </ul>
WhatsApp	Strumento di comunicazione	<ul style="list-style-type: none"> <li>• Uno dei più diffusi</li> </ul>
Draw.io	Strumento online per la creazione di diagrammi	<ul style="list-style-type: none"> <li>• Non necessita installazione</li> <li>• Utilizzabile ovunque</li> <li>• Molte tipologie di diagrammi sono disponibili</li> </ul>
Adobe Photoshop	Strumento di elaborazione fotografica	<ul style="list-style-type: none"> <li>• Già conosciuto dai Team Members</li> <li>• Utilizzato per produrre i mock-ups</li> </ul>
IntelliJ	Ambiente di sviluppo integrato IDE	<ul style="list-style-type: none"> <li>• Conosciuto dai Team Members</li> <li>• Uno dei più diffusi</li> </ul>
Java	Linguaggio di programmazione orientato agli oggetti utilizzato nella programmazione web lato server	<ul style="list-style-type: none"> <li>• Potente</li> <li>• Sicuro</li> <li>• Elastico</li> <li>• Scalabile</li> </ul>
JavaScript	Linguaggio di scripting orientato agli oggetti utilizzato nella programmazione web lato client	<ul style="list-style-type: none"> <li>• Potente</li> <li>• Sicuro</li> <li>• Elastico</li> <li>• Scalabile</li> </ul>
JQuery	Libreria JavaScript per applicazioni web	<ul style="list-style-type: none"> <li>• Facile da usare</li> <li>• Ottimizzazione dei tempi e dei costi</li> </ul>
MySQL	Relational Database Management System	<ul style="list-style-type: none"> <li>• Già conosciuto dai TM</li> <li>• Facilmente utilizzabile</li> </ul>
JUnit	Strumento per Unit Testing	<ul style="list-style-type: none"> <li>• Già conosciuto dai TM</li> <li>• Facilmente utilizzabile</li> </ul>
Mockito	Strumento per Unit Testing dei DAO	<ul style="list-style-type: none"> <li>• Facilmente Utilizzabile</li> </ul>
Katalon	Strumento per l'automazione del testing	<ul style="list-style-type: none"> <li>• Conosciuto dai Team Members</li> </ul>
Spring	Framework per lo sviluppo di applicazioni su piattaforma Java	<ul style="list-style-type: none"> <li>• Conosciuto dai Team Members</li> </ul>
Protobuf	Protocollo di serializzazione	<ul style="list-style-type: none"> <li>• Conosciuto dai Team Members</li> </ul>
Checkstyle	Strumento di sviluppo per aiutare i programmatori a	<ul style="list-style-type: none"> <li>• Richiesto</li> </ul>

	scrivere codice Java conforme a delle code conventions ben definite	
--	---	--

#### 4.4. Piano di Accettazione del Prodotto

Il prodotto verrà mostrato al cliente nella terza settimana di gennaio mediante una presentazione ufficiale e in quel giorno verrà effettuata una live view del sistema in funzione con relativo test da parte del cliente finale.

I criteri di accettazione sono presenti nel C06\_SOW\_UniSeats\_0.2.pdf.

#### 4.5. Project organization

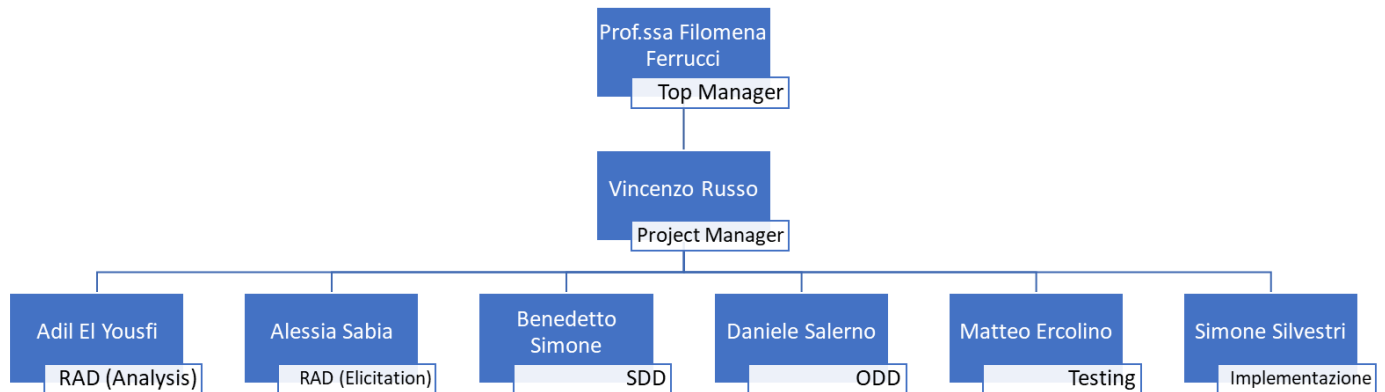
Il Top Manager è la professoressa Filomena Ferrucci.

Il Project Manager è Vincenzo Russo.

I Team Member sono:

- Adil El Yousfi
- Alessia Sabia
- Benedetto Simone
- Daniele Salerno
- Matteo Ercolino
- Simone Silvestri

L'organizzazione gerarchica risulta essere la seguente:



## 5. Pianificazione del Progetto

### 5.1. Project initiation

#### 5.1.1. Piano di Stima

Il budget di ore/uomo del progetto sono: 50 ore per 7 membri del team. Le previsioni di spesa per le diverse fasi del progetto sono così distribuite:

Fase	Ore TM	Ore PM	Totale
RAD	68,7	5	73,7
SDD	45	4	49
ODD	6	1	7
Test Documents	18	3	21
Implementazione	130	0,5	130,5

#### **5.1.2. Piano di Organizzazione dello Staff**

Lo staff è stato presentato dalla Top Manager, in seguito i Project Manager hanno scelto i candidati per formare i Teams. I principali requisiti richiesti dallo staff sono stati abilità nella progettazione e implementazione di applicazioni web e alle esperienze pregresse in progetti di gruppo.

#### **5.1.3. Resource acquisition plan**

In funzione delle esigenze che potrebbero sopraggiungere durante lo sviluppo del progetto ci si avvale della possibilità di acquisire risorse esterne.

Una risorsa esterna è sicuramente il modulo di Intelligenza Artificiale.

#### **5.1.4. Project staff training plan**

Non sono state pianificate ore per il training del Team, è stata fatta un breve introduzione sull' Unit Testing utilizzando JUnit.

### **5.2. Project work plans**

#### **5.2.1. Work activities**

Si rimanda a C06\_WBS\_Dictionary.pdf e C06\_WBS.pdf

#### **5.2.2. Schedule allocation**

Si rimanda a C06\_SA.pdf.

#### **5.2.3. Resource allocation**

Non si applica

#### **5.2.4. Budget allocation**

Si rimanda al C06\_BC\_1.1.pdf

#### **5.2.5. Procurement plan**

Non si applica

## **6. Controllo e Valutazione del Progetto**

### **6.1. Piano di Gestione dei Requisiti**

I requisiti sono stati raccolti dai Team Member, che si fanno carico di gestire la tracciabilità al fine di apportare le modifiche in modo semplice e prevedere l'impatto che un cambiamento potrebbe avere sull'intero progetto. Lo strumento utilizzato è la matrice di tracciabilità (C06\_MatriceDiTracciabilità.pdf). Non sono presenti tool, metodi o tecniche formali per analizzare l'impatto del cambiamento.

## 6.2. Pianificazione del cambiamento dello Scope del Progetto

Siccome il progetto è un progetto universitario e lo scope è stato ben definito sin dall'inizio, non si ritiene necessario l'utilizzo di uno strumento atto a gestire eventuali cambiamenti, supponendo che nel corso del progetto non saranno apportate modifiche allo scope.

L'elenco dei requisiti può subire variazioni, quali:

- Aggiunta di un requisito
- Modifica di un requisito
- Eliminazione di un requisito

Le change request non subiscono un processo formale.

## 6.3. Piano di Controllo Schedule

Lo schedule è stato pianificato e viene monitorato mediante l'utilizzo di Microsoft Project. Lo schedule sarà modificato in Project nel caso in cui si verifichino notevoli cambiamenti alle scadenze.

Un altro tool è Trello, che però è stato poco utilizzato anche data l'elevata funzionalità di Microsoft Teams.

## 6.4. Budget control plan

Il budget in euro è stato calcolato a partire dalle ore impiegate all'interno dell'Activity Schedule in Project. Il costo di ciascuna attività è dato da  $(\text{OrePM} * 35\text{€}) + (\text{OreTeamMembers} * 25\text{€})$ , come si può notare dalla formula le ore dei Team Members non sono moltiplicate per 6 in quanto sono state pre-aggregate. Inoltre è stato utilizzato un foglio di lavoro Excel, fornito dal Top Manager, per riportare le ore di lavoro. I fogli di lavoro sono stati inviati al Top Manager bisettimanalmente.

## 6.5. Quality Assurance plan

Sono state utilizzate le Checklists fornite dal docente sulla piattaforma di e-learning

## 6.6. Subcontractor management plan

**QUESTA PARTE NON SI APPLICA**

## 6.7. Project closeout plan

**QUESTA PARTE NON SI APPLICA**

# 7. Product delivery

La consegna del prodotto è prevista per la terza settimana di gennaio 2021

# 8. Supporting Process Plans

Pianificazione dei processi che durano per tutto il progetto (funzioni).

### 8.1. Project supervision and work environment

**QUESTA PARTE NON SI APPLICA**

### 8.2. Decision management

**QUESTA PARTE NON SI APPLICA**

### 8.3. Risk management

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	Se le ore dichiarate sono molto discordi rispetto a	Finanza



	con quelle previste	quanto previsto si potrebbero avere notevoli problemi di budget	
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

#### 8.4. Configuration management

**QUESTA PARTE NON SI APPLICA**

#### 8.5. Information management

**QUESTA PARTE NON SI APPLICA**

#### 8.6. Quality Assurance

Si rimanda al Quality Plan.

#### 8.7. Measurement

Definizione delle misure da raccogliere e dei metodi di validazione, analisi e reportistica. Fare riferimento alle metriche di Quality Management Plan

#### 8.8. Reviews and audits

Pianificazione delle review e degli audit. **QUESTA PARTE NON SI APPLICA**

#### 8.9. Verification and Validation

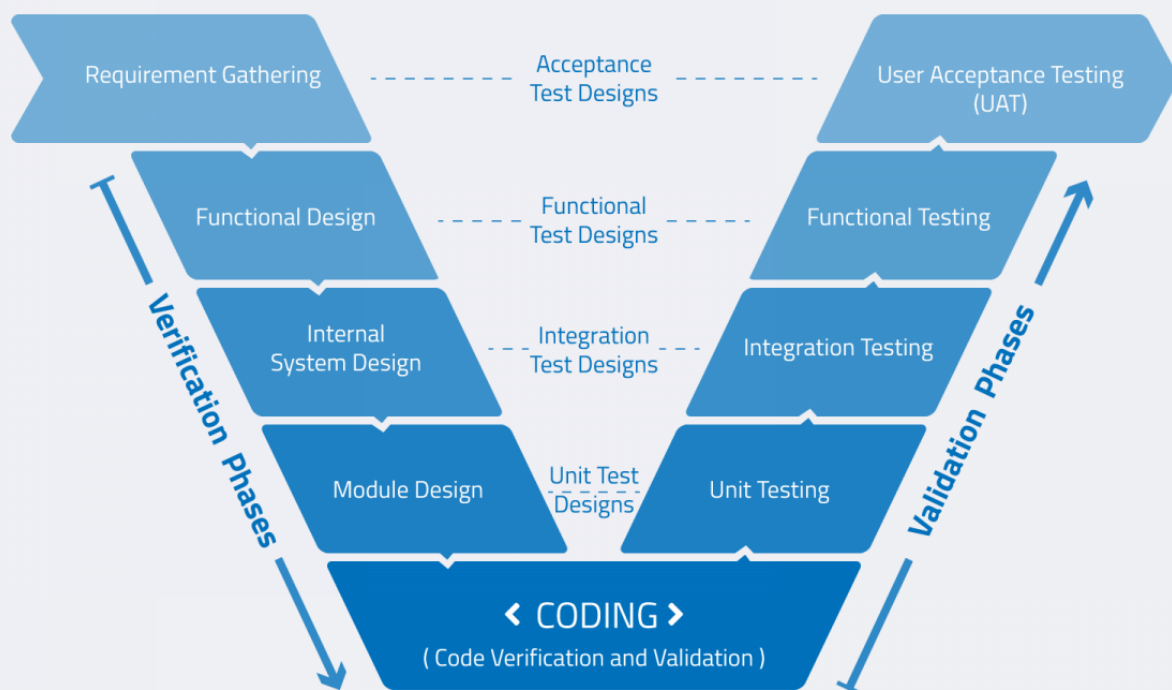
I requisiti di sistema definiscono ciò che il sistema deve fare per soddisfare le esigenze degli utenti identificate nel concetto di operazioni.

Il piano di convalida del sistema delinea come le parti interessate determineranno, alla fine del progetto, se il sistema completato soddisfa le esigenze degli utenti.

Il piano di verifica del sistema delinea i metodi di verifica da utilizzare per testare le operazioni del sistema. Ciò include strategie di test, definizioni di ciò che verrà testato, i livelli a cui verranno testati i diversi elementi del sistema e una matrice di test con mappatura dettagliata che collega il test eseguito ai requisiti di sistema.

Nel framework di ingegneria dei sistemi, un piano di convalida del sistema iniziale viene in genere sviluppato all'inizio del processo di progettazione, spesso dopo il completamento del concetto di operazioni, e può essere aggiornato periodicamente con l'avanzamento del progetto. Il piano può includere procedure di convalida specifiche da seguire durante il processo di convalida, ma tali procedure potrebbero anche essere sviluppate più vicino all'inizio dell'effettiva convalida del sistema quando è disponibile una comprensione più chiara delle funzionalità del sistema. Alla fine del processo di convalida, il team del progetto produce un rapporto che descrive in dettaglio i risultati dei test di convalida.

## Verification And Validation Model



È importante distinguere la convalida dalla verifica. Entrambe le attività valutano come è stato costruito un sistema. La differenza è al centro della valutazione:

**Verifica:** il sistema è stato costruito correttamente? La verifica è la conferma, attraverso prove oggettive, che i requisiti di sistema sono stati soddisfatti. La domanda che viene posta qui è se il progetto del sistema incorpora correttamente e completamente i requisiti, cioè se il sistema è stato costruito correttamente.

**Convalida:** è stato costruito il sistema giusto? La convalida è la conferma, sempre attraverso prove oggettive, che il sistema sviluppato raggiunge efficacemente lo scopo previsto e soddisfa le esigenze dell'utente per cui è stato sviluppato. In altre parole, la convalida tenta di determinare se è stato costruito il sistema corretto.

La maggior parte della verifica del sistema può essere eseguita prima che un sistema venga distribuito. La convalida, tuttavia, in realtà non può essere completata finché un sistema non si trova nel suo ambiente operativo e viene utilizzato dagli utenti previsti. Ad esempio, la convalida di un nuovo sistema di controllo dei segnali stradali non può essere completata fino a quando il nuovo sistema non sarà in vigore e non sarà possibile effettuare osservazioni sull'efficacia del controllo del traffico.

## 9. Additional plans

QUESTA PARTE NON SI APPLICA

### 1.1. Glossario

#### 1.1.1. Definizioni

- **Dipartimento:** struttura organizzativa che all'interno delle università italiane promuove e coordina le attività di uno o più settori della ricerca scientifica che siano omogenei per fini e per metodo
- **Aula:** sala destinata a riunioni importanti e solenni, come quelle dei tribunali e dei parlamenti, o alle lezioni nelle scuole e nelle università
- **Lezione:** attività didattica svolta da un docente con uno o più allievi in un tempo determinato.
- **Covid:** nome dato alla malattia associata al virus. SARS-CoV-2 è un nuovo ceppo di coronavirus, pandemico nell'anno 2020/21;
- **JUnit:** Framework di Unit Testing per il linguaggio di programmazione Java;
- **Travis:** Tool per la Continuous Integration;
- **Mockito:** Framework di testing Java;
- **Katalon:** Strumento di automazione del testing;
- **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;
- **DAO:** nell'ambito della programmazione Web, il DAO (Data Access Object) è un pattern architetturale per la gestione della persistenza;