



# SWEnergy

[project.swenergy@gmail.com](mailto:project.swenergy@gmail.com)

## Piano di progetto

Descrizione: XXX

<b>Stato</b>	Non approvato / Approvato
--------------	---------------------------

<b>Data</b>	DD/MM/YYYY
-------------	------------

---

<b>Redattori</b>	Nome 1
------------------	--------

	Nome 2
--	--------

<b>Verificatori</b>	Nome 1
---------------------	--------

	Nome 2
--	--------

<b>Approvatori</b>	Nome 1
--------------------	--------

	Nome 2
--	--------

---

<b>Versione</b>	0.1.X
-----------------	-------

## Registro delle modifiche

Versione	Data	Redattore	Verificatore	Approvatore	Modifiche
X.Y.Z	AAAA-MM-GG	Nome Cognome	Nome Cognome	Nessuno	Se scrivi qualcosa di molto lungo va a capo da solo
X.Y.Z	AAAA-MM-GG	Nome Cognome	Nome Cognome	Nessuno	Niente
X.Y.Z	AAAA-MM-GG	Nome Cognome	Nome Cognome	Nessuno	Niente

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Scopo del documento . . . . .	4
1.2	Scopo del prodotto . . . . .	4
1.3	Glossario . . . . .	5
1.4	Riferimenti . . . . .	5
1.4.1	Normativi . . . . .	5
1.4.2	Informativi . . . . .	5
1.5	Scadenze . . . . .	5
<b>2</b>	<b>Analisi dei rischi</b>	<b>6</b>
2.1	Rischi legati alle tecnologie . . . . .	6
RT-1	Conoscenza delle tecnologie carente . . . . .	6
RT-2	Strumenti software inadeguati . . . . .	7
RT-3	Codice incomprensibile . . . . .	8
2.2	Rischi legati alla comunicazione . . . . .	9
RC-1	Comunicazione interna carente . . . . .	9
RC-2	Conflitti decisionali . . . . .	10
RC-3	Comunicazione esterna carente . . . . .	10
2.3	Rischi legati alla pianificazione . . . . .	12
RP-1	Organizzazione carente . . . . .	12
RP-2	Comprensione dei requisiti carente . . . . .	13
RP-3	Interfacce incoerenti . . . . .	14
RP-4	Costi e tempi imprevisti . . . . .	15
2.4	Pericolosità e occorrenze . . . . .	16
<b>3</b>	<b>Modello di sviluppo</b>	<b>17</b>
3.1	Modello incrementale . . . . .	17
3.2	Iterazioni . . . . .	17
3.2.1	<i>Sprint</i> . . . . .	17
3.2.2	<i>Mini-sprint</i> . . . . .	17

---

3.3	Eventi . . . . .	18
3.3.1	SAL . . . . .	18
3.3.2	<i>Stand-up</i> . . . . .	18
3.4	Motivazioni . . . . .	19
<b>4</b>	<b>Pianificazione</b>	<b>21</b>
4.1	RTB . . . . .	21
4.1.1	Prima fase . . . . .	21
4.1.2	Stesura delle bozze . . . . .	21
4.1.3	Progettazione e prototipazione . . . . .	21
4.1.4	Stesura della documentazione . . . . .	21
4.1.5	Creazione del PoC . . . . .	21
4.1.6	Retifica della documentazione . . . . .	21
4.2	PB . . . . .	21
4.3	CA . . . . .	21
<b>5</b>	<b>Consuntivo di periodo</b>	<b>21</b>
5.1	Prospetto del Preventivo . . . . .	21
5.2	Effettivo . . . . .	21

# 1 Introduzione

Il presente documento, intitolato "Piano di Progetto", descrive e spiegare le decisioni organizzative adottate dal gruppo SWEnergy per lo sviluppo del progetto "*Easy Meal*", proposto dall'azienda **Imola Informatica**. Il "Piano di Progetto" è suddiviso nelle seguenti sezioni:

- **Analisi dei rischi:** identifica i rischi individuati dal gruppo e le strategie per mitigarli;
- **Modello di sviluppo:** descrive l'organizzazione temporale del team di SWEnergy;
- **Pianificazione:** dettaglia la pianificazione del lavoro del gruppo, incluse le attività, le risorse e i tempi necessari per lo sviluppo del progetto;
- **Preventivo:** presenta il preventivo delle ore di lavoro e il costo totale del progetto;
- **Consuntivo:** riporta le ore di lavoro e il costo effettivo del progetto fino al momento della stesura del piano di progetto della fase corrente: RTB.

## 1.1 Scopo del documento

Questo documento ha lo scopo di raccogliere in modo organico, coerente e uniforme tutte le informazioni riguardanti la pianificazione del progetto, al fine di fornire un riferimento per la gestione dello stesso. Al termine della prima fase del progetto (RTB), verrà utilizzato per valutare l'andamento del lavoro e per spiegare le decisioni adottate durante la pianificazione.

## 1.2 Scopo del prodotto

"*Easy Meal*" è una web app progettata per gestire le prenotazioni presso i ristoranti, sia dal lato dei clienti che dei ristoratori. Il prodotto finale sarà composto da due parti:

- **Cliente:** consente ai clienti di prenotare un tavolo presso un ristorante, visualizzare il menù e effettuare un ordine;
- **Ristoratore:** consente ai ristoratori di gestire le prenotazioni e gli ordini dei clienti, oltre a visualizzare la lista degli ingredienti necessari per preparare i piatti ordinati.

## 1.3 Glossario

Al fine di evitare ambiguità linguistiche e garantire un'utilizzazione coerente delle terminologie nei documenti, il gruppo ha redatto un documento interno chiamato "Glossario". Questo documento definisce in modo chiaro e preciso i termini che potrebbero generare ambiguità o incomprensione nel testo. I termini presenti nel Glossario sono identificati da una 'G' (per esempio parola<sub>G</sub>) a pedice.

## 1.4 Riferimenti

### 1.4.1 Normativi

- " *Way of Working*";
- Documento del capitolato d'appalto C3 - *Easy Meal*;
- Regolamento del progetto;

### 1.4.2 Informativi

Slide dell'insegnamento di Ingegneria del Software:

- Modelli di sviluppo del software;
- Gestione di progetto;
- Analisi dei requisiti;

## 1.5 Scadenze

Il *team* di SWEnergy si impegna a rispettare le seguenti scadenze per il completamento del progetto:

- **Prima revisione (avanzamento RTB):** 21 dicembre 2023;
- **Seconda revisione (avanzamento PB):** da definire;
- **Terza revisione (avanzamento CA):** da definire;

## 2 Analisi dei rischi

La seguente sezione ha lo scopo di identificare e catalogare i rischi che potrebbero verificarsi durante lo svolgimento del progetto, per poterli prevenire o almeno provare a mitigarli. Ciascun rischio è descritto seguendo la struttura:

- codice identificativo seguito da un numero progressivo:
  - **RT**: rischi legati alle tecnologie;
  - **RC**: rischi legati alla comunicazione;
  - **RP**: rischi legati alla pianificazione.
- titolo: il nome che identifica il rischio;
- descrizione: una breve descrizione del rischio;
- identificazione: in quale modo il *team* può capire se si sta verificando qualche danno;
- mitigazione: come il *team* ha modo di prevenire o attenuare i danni causati dal rischio;

Dopo la descrizione di ciascun rischio, viene presentata una tabella che riassume i rischi individuati, associando a ciascuno un indice di gravità e un indice di frequenza.

### 2.1 Rischi legati alle tecnologie

#### RT-1 Conoscenza delle tecnologie carente

- **Descrizione:** nello sviluppo del progetto, si può incorrere nella situazione in cui almeno qualche membro non conosce almeno una tecnologia adottata dal gruppo e necessaria per lo sviluppo del progetto.
- **Identificazione:** il *team* ha individuato le tecnologie conosciute dal gruppo. Con il proponente sono state discusse e concordate le tecnologie da utilizzare per lo sviluppo del progetto. In questo modo, sono state individuate le tecnologie non conosciute dal gruppo.
- **Mitigazione:**

- *workshop* interni: il *team* sceglie una o due persone per ogni tecnologia non conosciuta dal gruppo. Le persone scelte si occupano di approfondire la tecnologia e di organizzare un *workshop* interno. Le persone scelte svilupperanno inoltre degli esempi di codice per illustrare l'utilizzo della tecnologia e degli appunti da condividere;
- seminari con il proponente: il *team* partecipa a dei seminari organizzati con il proponente, per approfondire le tecnologie non conosciute dal gruppo. Il proponente spiegherà le tecnologie e fornirà degli esempi di codice per illustrarne l'utilizzo;
- dialogo con il proponente: il *team* può contattare il proponente per chiedere chiarimenti sulle tecnologie non conosciute dal gruppo.
- documentazione: il *team* può consultare la documentazione ufficiale delle tecnologie non conosciute dal gruppo.
- *pair programming*: il codice viene sviluppato con almeno un altro membro del gruppo. Le modalità di lavoro sono meglio descritte nel documento " *Way of working*".
- *code review*: il codice viene revisionato da almeno un altro membro del gruppo. Le modalità di lavoro sono meglio descritte nel documento " *Way of working*".
- divisione del *front-end* e del *back-end*: il *team* si divide in due sottogruppi, uno che si occupa del *front-end* e l'altro del *back-end*. In questo modo, i membri del gruppo possono concentrarsi su un numero ridotto di tecnologie. I due gruppi si scambiano i ruoli al termine della prima fase del progetto: RTB.

## RT-2 Strumenti software inadeguati

- **Descrizione:** l'utilizzo di strumenti software datati o poco efficienti può portare a ritardi nello sviluppo del progetto;
- **Identificazione:**
  - I membri del gruppo possono lamentare l'utilizzo di strumenti software poco efficienti durante le riunioni interne;



- Nella rendicontazione delle ore, si nota che la medesima attività subisce continui ritardi.

- **Mitigazione:**

- l'amministratore deve tenere sotto controllo le versioni degli strumenti software utilizzati;
- i membri del gruppo si informano in merito a nuove tecnologie da adottare.

### RT-3 Codice incomprensibile

- **Descrizione:** il codice prodotto da qualche membro del gruppo è di difficile comprensione per gli altri membri del gruppo.

- **Identificazione:**

- *code review*: durante la verifica del codice, i verificatori possono riscontrare difficoltà nella comprensione del codice;
- *testing*: durante la fase di testing, i tester possono riscontrare difficoltà nella comprensione del codice;
- dopo un lasso di tempo ampio, i membri del gruppo possono riscontrare difficoltà nella comprensione del codice.

- **Mitigazione:**

- *Way of working*: il gruppo stila delle linee guida da seguire per la stesura del codice, in modo da uniformare la stesura del codice e facilitarne la comprensione.
- documentazione: il codice deve essere documentato in modo chiaro e preciso, in modo da facilitarne la comprensione. Per maggiori chiarimenti si rimanda al "*Way of working*".
- *testing*: il codice deve essere testato in modo approfondito, per facilitarne la comprensione e illustrarne i casi d'uso.
- librerie apposite: il *source code* comprende delle librerie apposite per aiutare la stesura della documentazione, per esempio *OpenAPI* per la documentazione delle API.

## 2.2 Rischi legati alla comunicazione

### RC-1 Comunicazione interna carente

- **Descrizione:** La comunicazione interna non è efficace ed efficiente, causando riunioni interne più lunghe del previsto e rallentando le attività.
- **Identificazione:**
  - dubbi ripetuti: durante le riunioni interne, i membri del gruppo possono porre domande già fatte in precedenza;
  - riunioni interne lunghe: le riunioni interne possono protrarsi oltre il tempo previsto;
  - qualche membro del gruppo non sa che cosa deve fare;
  - qualche membro del gruppo non sa come fare qualcosa;
  - fraintendimenti frequenti: i membri del gruppo possono fraintendersi frequentemente.
  - durante le retrospettive, i membri del gruppo possono lamentarsi di una comunicazione interna carente.
- **Mitigazione:**
  - ” *Way of working*”: il gruppo stila delle linee guida da seguire per la comunicazione interna, in modo da uniformare la comunicazione interna e facilitarla. Sono dunque creati dei canali di comunicazione come *Telegram*, *Discord* oppure su *GitHub*;
  - documentazione: il gruppo stila una documentazione adeguata per facilitare la comunicazione interna. A seconda dell’argomento la documentazione può avere diverse forme. Per esempio, il riassunto delle riunioni può essere ritrovato nei verbali delle riunioni; le discussioni in merito ai casi d’uso possono essere ritrovate nella sezione delle *Discussions* di *Github* dedicata. Ciascun macro argomento deve avere una sezione all’interno del ” *Way of working*” in cui viene descritta la documentazione adatta;

- *meeting* frequenti: il gruppo si impegna a tenere riunioni interne frequenti, in modo da ridurre la durata delle riunioni interne e facilitare la comunicazione interna;
- ordine del giorno: ciascuna riunione deve avere l'ordine del giorno ben definito, per discutere di tutti gli argomenti utili allo sviluppo del progetto e per definire la durata di ciascuno dei punti dell'ordine del giorno;
- retrospettiva: durante la retrospettiva, il gruppo deve pensare a soluzioni *ad hoc* per migliorare la comunicazione interna.

## RC-2 Conflitti decisionali

- **Descrizione:** Il gruppo potrebbe dilungarsi nella discussione di una sola idea, senza raggiungere una decisione finale.
- **Identificazione:**
  - un punto dell'ordine del giorno subisce un ritardo grave;
- **Mitigazione:**
  - dibattito: i membri del gruppo discutono riguardo all'importanza del punto dell'ordine del giorno, per capire se è necessario approfondire la discussione o meno;
  - approfondimento: se il punto dell'ordine del giorno è ritenuto importante, almeno due membri del gruppo si impegnano a studiare i pro ed i contro delle varie soluzioni possibili. Può essere chiesto un supporto al proponente oppure al committente per chiarire i dubbi;
  - votazione: alla fine del dibattito i membri del gruppo votano per la soluzione che ritengono più opportuna. La votazione si ritiene conclusa quando la maggioranza dei membri del gruppo ha espresso la propria preferenza e il risultato non è un pareggio.
  - il responsabile ha il compito di vigilare sul corretto svolgimento del dibattito e della votazione, in modo da evitare che si dilunghi troppo.

## RC-3 Comunicazione esterna carente

- **Descrizione:** Le comunicazioni con il proponente o con il committente non sono efficaci ed efficienti, causando riunioni esterne più lunghe del previsto e rallentando le attività; oppure rallentando le attività del gruppo a causa di risposte tardive o mancanti.
- **Identificazione:**
  - dubbi ripetuti: durante le riunioni esterne, i membri del gruppo possono porre domande già fatte in precedenza;
  - riunioni esterne lunghe: le riunioni esterne possono protrarsi oltre il tempo previsto;
  - risposte tardive o mancanti: il proponente o il committente può rispondere in ritardo o non rispondere affatto alle comunicazioni del gruppo;
  - durante le retrospettive, i membri del gruppo possono lamentarsi di una comunicazione esterna carente.
- **Mitigazione:**
  - ordine del giorno: il responsabile si impegna a stilare l'ordine del giorno delle riunioni esterne, per tempo, ne discute la struttura con il gruppo e lo condivide con il proponente e con il committente in anticipo;
  - SAL: il gruppo si impegna a mantenere il proponente aggiornato sullo stato di avanzamento del progetto, in modo da ridurre la durata delle riunioni esterne e migliorare la qualità del supporto del proponente;
  - retrospettive: sono previste delle retrospettive all'interno dei SAL con il proponente, si discute della qualità delle comunicazioni e si pensa a soluzioni *ad hoc* per migliorare la comunicazione esterna;
  - comunicazioni frequenti: il proponente viene tenuto aggiornato frequentemente sullo stato di avanzamento del progetto mediante gli appositi canali di comunicazione, per esempio *Telegram*;
  - diario di bordo: il gruppo si impegna a tenere dei diari di bordo, quando richiesti dal committente, per aggiornarlo sullo stato di avanzamento del progetto;

- *meeting* supplementari: se il gruppo manifesta dei dubbi o delle incertezze, può richiedere un *meeting* supplementare con il proponente o con il committente;
- documentazione: il responsabile si impegna ad aggiornare la documentazione inerente agli argomenti trattati durante le riunioni esterne, per dare modo ai membri del gruppo di consultarla in caso di dubbi o incertezze.

## 2.3 Rischi legati alla pianificazione

I membri del gruppo non hanno mai assunto un ruolo manageriale in precedenza e non hanno mai lavorato in un gruppo di lavoro così numeroso. Questo porta a problemi di gestione del tempo e delle risorse. D'altro canto, SWEnergy si rende conto che lo scopo del progetto è proprio quello di acquisire esperienza, anche in questi termini. Per cui, il gruppo ha deciso di individuare alcuni rischi legati alla pianificazione, per poterli prevenire o mitigare.

### RP-1 Organizzazione carente

- **Descrizione:** Il gruppo, oppure qualche membro, potrebbe non essere in grado di svolgere le proprie attività, oppure potrebbe riscontrare delle difficoltà a causa di una cattiva organizzazione.
- **Identificazione:**
  - gravi rallentamenti: si tratta del sintomo più evidente di un'organizzazione carente; purtroppo tutti i rischi causano ritardi, quindi è difficile capire se il rallentamento è causato da un'organizzazione carente o da un altro rischio. D'altro canto, tendenzialmente, il rallentamento non ha una sola causa, ma è dovuto a più fattori;
  - membri confusi: i membri del gruppo non sanno quali sono i compiti a loro assegnati, oppure non sanno come svolgerli;
  - carenza di risorse: sono stati assegnati più incarichi di quelli sostenibili con le risorse disponibili.
  - scadenze non aggiornate: il gruppo o qualche suo membro non è in grado di rispettare le scadenze e non sono aggiornate. Si tratta di un modo molto semplice, per ricadere nel sintomo individuato precedentemente.

- **Mitigazione:**

- il responsabile mantiene aggiornate le *issue*. Ciascuna *issue* deve essere ben documentata nella propria descrizione; se è il caso, si possono aggiungere i riferimenti a della documentazione supplementare. Maggiori informazioni sono presenti nel documento ” *Way of working*”;
- in caso di dubbi, i membri di SWEnergy possono rivolgersi al responsabile, che si occuperà di chiarire la situazione, o di indirizzare il membro verso chi può aiutarlo;
- il responsabile mantiene aggiornato il *project* su *GitHub*, in particolare per quanto riguarda l’assegnamento delle *issue* e delle scadenze. Maggiori informazioni sono presenti nel documento ” *Way of working*”;
- sono utilizzate delle tecnologie che permettono di tenere traccia delle attività svolte e di quelle da svolgere su *GitHub*;
- durante le retrospettive, il gruppo discute di eventuali problemi organizzativi e cerca di trovare soluzioni per migliorare la pianificazione;
- dialogo con il proponente: sono chiesti consigli al proponente in merito, in quanto ha più esperienza nel settore e ha modo di collaborare con figure manageriali.

## **RP-2 Comprensione dei requisiti carente**

- **Descrizione:** Il gruppo o qualche suo membro potrebbe non essere in grado di comprendere i requisiti del progetto, oppure potrebbe riscontrare delle difficoltà a causa di una cattiva comprensione dei requisiti.
- **Identificazione:**
  - dubbi: i membri del gruppo hanno dei dubbi in merito ai requisiti;
  - dibattiti sui requisiti: i membri del gruppo discutono tra loro in merito ai requisiti;
  - discrepanza nella progettazione: i membri del gruppo progettano in modo diverso, a causa di una cattiva comprensione dei requisiti.

- **Mitigazione:**

- "Analisi dei requisiti": il metodo più formale per ovviare a questa situazione risulta essere l'"Analisi dei requisiti". I requisiti dovrebbero essere chiari e completi. Inoltre, il documento contiene i casi d'uso, che aiutano a comprendere meglio i requisiti concordati con il proponente.
- *GitHub Discussions*: in caso di dubbi, i membri documentano le domande e le risposte in una sezione apposita della *repository* su *GitHub*. In questo modo, è più semplice ripercorrere le motivazioni di alcune scelte progettuali. Inoltre, rende più semplice il dialogo con il proponente, in quanto è possibile consultare le domande e le risposte precedenti.
- dialogo con il proponente: si discute con il proponente in merito ai requisiti, per chiarire eventuali dubbi e per decidere in maggiore dettaglio le funzionalità del prodotto.
- messaggi tempestivi con il proponente: in caso di dubbi semplici e veloci da risolvere, si inviano dei messaggi al proponente, per ottenere una risposta tempestiva.

### RP-3 Interfacce incoerenti

- **Descrizione:** Durante la fase integrativa di più componenti, risultano delle incongruenze che rendono impossibile l'integrazione, allungandone i tempi ed i costi.

- **Identificazione:**

- test di integrazione falliti: i test di integrazione falliscono a causa di incongruenze tra le interfacce delle componenti;
- discussioni interne in merito alle interfacce: i membri del gruppo discutono tra loro in merito alle interfacce delle componenti, per capire come risolvere le incongruenze.
- l'applicativo non funziona in seguito ad un'integrazione.
- interfacce discordanti tra le componenti che sono da integrare.

- **Mitigazione:**

- dialogo interno: i membri del gruppo discutono tra loro in merito alle interfacce delle componenti, prima di cominciare a sviluppare le componenti stesse;
- test di integrazione: vengono effettuati dei test di integrazione, per verificare che le componenti siano compatibili tra loro;
- documentazione: le interfacce delle componenti sono documentate in modo chiaro e completo, per evitare incomprensioni.

#### RP-4 Costi e tempi imprevisti

- **Descrizione:** Durante lo sviluppo del progetto, si può incorrere in costi o rallentamenti imprevisti. Si tratta, a tutti gli effetti, di arginare il danno prodotto da un rischio che si è verificato.
- **Identificazione:**
  - cambiamenti significativi nelle tempistiche di completamento del progetto;
  - variazioni notevoli dei costi di realizzazione.
  - monitoraggio costante: monitoraggio dei costi e delle tempistiche al completamento delle *milestone* e ad ogni *stand-up*.
- **Mitigazione:**
  - *buffer* di tempo: il *team* ha preventivamente inserito dei *buffer* di tempo tra le varie attività, per poter gestire eventuali ritardi;
  - *buffer* di costi: il *team* ha preventivamente inserito dei *buffer* di costi tra le varie attività, per poter gestire eventuali costi imprevisti;
  - pianificazione in itinere: il *team* si adatta alle variazioni dei costi e delle tempistiche di completamento, per poter gestire eventuali costi imprevisti. In questo caso, sono aggiornate le scadenze nel *project* su *GitHub* e i costi. A seconda della situazione, le *issue* sono riassegnate e le *milestone* sono adattate allo *status quo*.



## 2.4 Pericolosità e occorrenze

Per ogni rischio, il *team* ha individuato un indice di gravità e un indice di frequenza, per poter stimare il rischio residuo. L'indice di gravità e quello di frequenza sono due numeri compresi tra 1 e 5. L'indice di rischio residuo è il prodotto tra i due indici, può quindi assumere valori compresi tra 1 e 25. Più il rischio residuo è alto, maggiori sono i danni che può causare e più è probabile che si verifichi. Si noti che non è detto che il verificarsi del rischio causi i danni massimi, le strategie di mitigazione servono proprio per prevenire e attenuare i danni.

Rischi tecnologici	Gravità	Frequenza	Rischio residuo
RT-1 Conoscenza delle tecnologie carente	5	4	20
RT-2 Strumenti software inadeguati	1	2	2
RT-3 Codice incomprensibile	2	2	4

Rischi comunicativi	Gravità	Frequenza	Rischio residuo
RC-1 Comunicazione interna carente	3	3	9
RC-2 Conflitti decisionali	1	2	2
RC-3 Comunicazine esterna carente	2	2	4

Rischi organizzativi	Gravità	Frequenza	Rischio residuo
RP-1 Organizzazione carente	3	4	12
RP-2 Comprensione dei requisiti carente	2	3	6
RP-3 Interfacce incoerenti	4	2	8
RP-4 Costi e tempi imprevisi	5	3	15

Tabella 1: Tabella della pericolosità e dell'occorrenza dei rischi.

## 3 Modello di sviluppo

### 3.1 Modello incrementale

SWEnergy ha deciso di adottare il modello di sviluppo incrementale, che prevede la suddivisione del progetto in incrementi. Al completamento di ciascun incremento viene rilasciata una nuova versione del prodotto, che è mostrata al proponente durante le riunioni di revisione. Poiché il *team* non ha mai lavorato in ambito professionale, il modello di sviluppo è ispirato al *framework* Scrum con *sprint* di due settimane, con qualche modifica per adattarlo alle esigenze del progetto e del gruppo. In particolare, SWEnergy introduce una retrospettiva *in media res* per valutare il lavoro svolto e apportare eventuali modifiche al processo di sviluppo. In aggiunta, non sono previste le *daily stand-up meeting*, in quanto il gruppo ritiene che la cadenza delle suddette sia troppo elevata considerando che il progetto è svolto da studenti universitari e non da lavoratori a tempo pieno.

### 3.2 Iterazioni

#### 3.2.1 *Sprint*

Uno *sprint* ha una durata di due settimane. Durante questo periodo, il gruppo si impegna a sviluppare l'incremento del prodotto concordato con il proponente. La durata di uno *sprint* è stata scelta sulle esigenze del proponente; permette al *team* di effettuare un mini-*sprint* di una settimana; consente di ricevere *feedback* frequenti dal proponente e di apportare modifiche al prodotto in modo tempestivo. In aggiunta, permette al *team* di risolvere eventuali problemi o dubbi con il proponente in modo dinamico, flessibile e tempestivo. I mini-*sprint* aumentano la frequenza delle retrospettive e mantengono il gruppo focalizzato sul lavoro da svolgere. Permettono di valutare lo svolgimento delle attività durante lo *sprint* e di apportare modifiche al processo di sviluppo adattando il lavoro alle esigenze del progetto.

#### 3.2.2 *Mini-sprint*

Questa iterazione emula il *framework* Scrum, con una cadenza settimanale piuttosto che bisettimanale o mensile. Si tratta di uno *sprint* interno al gruppo: il proponente non

viene coinvolto. Al termine di un mini-*sprint* potrebbe corrispondere un cambio dei ruoli, in base alle esigenze del progetto e del gruppo.

### 3.3 Eventi

#### 3.3.1 SAL

Lo Stato di Avanzamento del Lavoro (SAL) è un incontro con il proponente che avviene ogni due settimane di venerdì. Queste riunioni sono utili per condividere i *feedback* in entrambe le direzioni: il proponente può valutare il lavoro svolto dal gruppo; e SWEnergy può esprimere le proprie opinioni sul prodotto. Di seguito sono riportate le attività principali che avvengono durante un SAL:

- ***Sprint review***: Il gruppo presenta il lavoro svolto durante lo *sprint* e il proponente fornisce dei *feedback* sul prodotto. In aggiunta, il gruppo può porre domande per chiarire eventuali dubbi in merito ai requisiti e alle funzionalità richieste, oppure in merito all'implementazione di queste ultime.
- ***Sprint retrospective***: Il gruppo discute sulle modalità di lavoro. In particolare, valuta se il processo di sviluppo è stato efficace ed efficiente ed in quale modo sia migliorabile. Sono chiesti consigli al proponente in merito all'organizzazione del lavoro; sono riportati eventuali problemi riscontrati durante lo *sprint* e sono proposte soluzioni per risolverli.
- ***Sprint planning***: Il gruppo e il proponente concordano l'incremento del prodotto da sviluppare durante lo *sprint* successivo; ovvero che cosa inserire nello *sprint backlog*.

#### 3.3.2 Stand-up

Il nome *stand-up* è ispirato ai *daily stand-up meeting* del framework Scrum. Si noti che l'incontro con il proponente avviene un venerdì su due; mentre le *stand-up*, gli incontri all'inizio e al termine di un mini-*sprint*, hanno luogo ogni domenica, per dare modo al responsabile di considerare i *feedback* del proponente e di pianificare l'iterazione successiva. L'organizzazione viene poi discussa durante la *stand-up*. Di seguito sono riportate le attività principali che avvengono durante una *stand-up*:

- **Brainstorming:** Il responsabile riassume il lavoro svolto durante la settimana e ciascun membro del gruppo può arricchire la spiegazione con le proprie esperienze, per esempio, descrivendo le difficoltà incontrate e le soluzioni adottate.
- **Retrospettiva:** Il gruppo discute sulle modalità di lavoro. In particolare, valuta se il processo di sviluppo è stato efficace ed efficiente ed in quale modo sia migliorabile. Sono riportati eventuali problemi riscontrati durante il mini-*sprint* e sono proposte soluzioni per risolverli. Eventualmente si prende nota dei problemi per discuterne con il proponente durante il SAL; oppure per domandare consigli al committente.
- **Pianificazione:** Il responsabile mostra la pianificazione del mini-*sprint* successivo. I membri del gruppo possono intervenire per proporre miglioramenti o per chiarire eventuali dubbi. Infine, il responsabile assegna i compiti ai membri del gruppo, secondo le loro disponibilità, capacità e preferenze.

Risulta utile dividere le *stand-up* in due gruppi: le *stand-up* che avvengono la domenica successiva al SAL sono dedicate alla pianificazione del mini-*sprint* successivo; mentre le *stand-up* che avvengono la domenica successiva sono principalmente dedicate alla retrospettiva del mini-*sprint* appena concluso. Queste ultime sono utilizzate anche per organizzare il SAL successivo: il responsabile raccoglie i problemi riscontrati; organizza le domande da porre al proponente; e pianifica l'ordine del giorno del SAL. Il *team* può richiedere qualche modifica all'ordine del giorno. In questo modo, è possibile anticipare eventuali problemi e risolverli prima del SAL, evitando di sprecare tempo prezioso. In aggiunta, il responsabile può condividere l'ordine del giorno con il proponente prima del SAL. Si noti che è prevista una vera e propria *stand-up*, della durata di circa 30 minuti, anche tra il giovedì o il venerdì precedente al SAL, per effettuare un veloce *brainstorming*, raccogliere le ultime domande e modificare l'ordine del giorno in base alle risposte del proponente e alle necessità del gruppo.

### 3.4 Motivazioni

SWEnergy ha deciso di organizzarsi come descritto in precedenza più per necessità che per scelta: durante i corsi di Ingegneria del Software e di Metodi e Tecnologie per lo Sviluppo Software, i membri del gruppo hanno appreso i concetti fondamentali del fra-

*mework* Scrum. Dunque, il *team* ha deciso di metterli in pratica. Come già accennato, il gruppo non ha mai lavorato in un ambito professionale e non ha esperienza con i metodi di lavoro ed organizzativi. In aggiunta, il proponente ha richiesto una pianificazione di almeno due settimane. Con la seguente organizzazione, SWEnergy spera di riuscire a soddisfare le esigenze del proponente mitigando i rischi individuati durante l'analisi dei rischi. Il *framework* Scrum dovrebbe fornire i seguenti vantaggi:

1. **Flessibilità:** Il gruppo può adattare il processo di sviluppo alle esigenze del progetto. In particolare, il gruppo può modificare la pianificazione in base alle esigenze del proponente e del gruppo.
2. **Comunicazione trasparente:** Il gruppo può rilasciare incrementi del prodotto in modo tempestivo. In aggiunta, il proponente può valutare il lavoro svolto dal gruppo e fornire *feedback* in itinere, approfondendo i requisiti.
3. **Miglioramento continuo:** Le retrospettive permettono al gruppo di valutare il processo di sviluppo e di apportare modifiche per migliorarlo. In aggiunta, il gruppo può discutere con il proponente o con il committente per ricevere consigli e suggerimenti per migliorare l'organizzazione.
4. **Monitoraggio costante:** La pianificazione basata sugli *sprint* permette di identificare e affrontare rischi in modo più tempestivo, riducendo la possibilità di ritardi gravi.

## **4 Pianificazione**

### **4.1 RTB**

#### **4.1.1 Prima fase**

#### **4.1.2 Stesura delle bozze**

#### **4.1.3 Progettazione e prototipazione**

#### **4.1.4 Stesura della documentazione**

#### **4.1.5 Creazione del PoC**

#### **4.1.6 Retifica della documentazione**

### **4.2 PB**

### **4.3 CA**

## **5 Consuntivo di periodo**

### **5.1 Prospetto del Preventivo**

### **5.2 Effettivo**