

SiVoDiM



Norme di Progetto

| | |
|-------------------------------|---|
| Versione | 1.0.0 |
| Redattori | <Redattore> |
| Verificatori | <Verificatore> |
| Responsabili | <Responsabile> |
| Uso | <Uso> |
| Lista di distribuzione | <i>Stark Labs</i> Prof. Vardanega Tullio, Dr. Cardin Riccardo |

Documento riguardante l'insieme di norme stabilite dal gruppo StarkLabs per la realizzazione di SiVoDiM.

Indice

| | | |
|----------|---|----------|
| 1 | Introduzione | 1 |
| 1.1 | Scopo del documento | 1 |
| 1.2 | Scopo del progetto | 1 |
| 1.3 | Glossario | 1 |
| 1.4 | Riferimenti | 1 |
| 1.4.1 | Normativi | 1 |
| 1.4.2 | Informativi | 1 |
| 2 | Processi primari | 2 |
| 2.1 | Processo di sviluppo | 2 |
| 2.1.1 | Analisi dei requisiti | 2 |
| 2.1.1.1 | Studio di fattibilit  | 2 |
| 2.1.1.2 | Requisiti | 2 |
| 2.1.1.3 | Documentazione dei requisiti | 2 |
| 2.1.1.4 | Casi d'uso | 2 |
| 2.1.1.5 | Tracciamento dei requisiti | 2 |
| 2.1.2 | Progettazione | 2 |
| 2.1.2.1 | Diagrammi UML | 2 |
| 2.1.2.2 | Design pattern | 2 |
| 2.1.2.3 | Classi di verifica | 2 |
| 2.1.2.4 | Stile di progettazione | 2 |
| 2.1.3 | Progettazione architetture | 2 |
| 2.1.4 | Progettazione architetture di dettaglio | 2 |
| 2.1.5 | Codifica e convenzioni | 2 |
| 2.1.5.1 | Linguaggi di codifica | 2 |
| 2.1.5.2 | Framework e librerie | 2 |
| 2.1.5.3 | Convenzioni di codifica | 2 |
| 2.1.5.4 | File HTML | 2 |
| 2.1.5.5 | Nomenclatura | 2 |
| 2.1.5.6 | Intestazione di un file | 2 |
| 2.1.5.7 | Commenti | 2 |
| 2.1.5.8 | Esecuzione dei test di unit  | 2 |
| 3 | Processi di supporto | 3 |
| 3.1 | Processo di documentazione | 3 |
| 3.1.1 | Template | 3 |
| 3.1.2 | Versionamento | 3 |
| 3.1.3 | Struttura dei documenti | 3 |
| 3.1.3.1 | Prima pagina | 3 |
| 3.1.3.2 | Registro delle modifiche | 3 |
| 3.1.3.3 | Indice | 3 |
| 3.1.3.4 | Formattazione delle pagine | 3 |
| 3.1.4 | Classificazione dei documenti | 3 |
| 3.1.4.1 | Documenti informali | 3 |
| 3.1.4.2 | Documenti formali | 3 |
| 3.1.4.3 | Verbale | 3 |
| 3.1.5 | Norme tipografiche | 3 |
| 3.1.5.1 | Punteggiatura | 3 |
| 3.1.5.2 | Stile del testo | 3 |
| 3.1.5.3 | Composizione del testo | 3 |
| 3.1.5.4 | Formati ricorrenti | 3 |
| 3.1.5.5 | Sigle | 3 |
| 3.1.6 | Componenti grafiche | 3 |

| | | |
|-----------|--|----------|
| 3.1.6.1 | Immagini | 3 |
| 3.1.6.2 | Tabelle | 3 |
| 3.1.7 | Procedure di avanzamento di un documento | 3 |
| 3.2 | Processo di verifica | 3 |
| 3.2.1 | Tecniche di analisi | 3 |
| 3.2.1.1 | Analisi statica | 3 |
| 3.2.1.1.1 | Walkthrough | 3 |
| 3.2.1.1.2 | Inspection | 3 |
| 3.2.1.2 | Analisi dinamica | 3 |
| 3.2.1.3 | Test | 4 |
| 3.2.1.3.1 | Test di unità | 4 |
| 3.2.1.3.2 | Test di integrazione | 4 |
| 3.2.1.3.3 | Test di sistema | 4 |
| 3.2.1.3.4 | Test di regrassione | 4 |
| 3.2.1.3.5 | Test di accettazione | 4 |
| 3.2.2 | Verifica dei documenti | 4 |
| 3.2.3 | Verifica dei diagrammi | 4 |
| 3.2.4 | Verifica del codice | 4 |
| 3.2.4.1 | Analisi statica | 4 |
| 3.2.4.2 | Analisi dinamica | 4 |
| 4 | Processi organizzativi | 4 |
| 4.1 | Gestione dei processi | 4 |
| 4.1.1 | Attività | 4 |
| 4.1.1.1 | Gestione delle comunicazioni | 4 |
| 4.1.1.1.1 | Comunicazione interna | 4 |
| 4.1.1.1.2 | Comunicazione esterna | 4 |
| 4.1.1.1.3 | Composizione email | 4 |
| 4.1.1.2 | Gestione delle riunioni | 5 |
| 4.1.1.2.1 | Riunioni interne | 5 |
| 4.1.1.2.2 | Riunioni esterne | 6 |
| 4.1.1.3 | Gestione del sistema dei task _G | 6 |
| 4.1.1.4 | Gestione delle milestone _G | 6 |
| 4.1.1.5 | Gestione dei task _G | 6 |
| 4.1.1.6 | Gestione dello svolgimento dei task _G | 6 |
| 4.1.2 | Procedure | 7 |
| 4.1.2.1 | Generazione di una milestone | 7 |
| 4.1.2.2 | Assegnazione di un task | 7 |
| 4.1.2.3 | Svolgimento di un task | 7 |
| 4.1.2.4 | Rilevazione dei rischi | 9 |
| 4.1.2.5 | Ruoli di Progetto | 9 |
| 4.1.2.5.1 | Responsabile di Progetto | 9 |
| 4.1.2.5.2 | Amministratore di Progetto | 9 |
| 4.1.2.5.3 | Analista | 9 |
| 4.1.3 | Norme | 9 |
| 4.1.3.0.4 | Progettista | 9 |
| 4.1.3.0.5 | Programmatore | 10 |
| 4.1.3.0.6 | Verificatore | 10 |
| 4.1.4 | Strumenti | 10 |
| 4.1.4.1 | Teamwork | 10 |
| 4.1.4.2 | Astah | 10 |
| 4.1.4.3 | Telegram | 10 |
| 4.1.4.4 | Microsoft Office Power Poin | 10 |
| 4.2 | Gestione delle infrastrutture | 10 |
| 4.2.0.5 | Attività | 10 |
| 4.2.0.5.1 | Gestione del repository | 10 |

| | | |
|-----------|--|----|
| 4.2.0.5.2 | Gestione del messaggio di commit | 10 |
| 4.2.0.6 | Procedure | 10 |
| 4.2.0.6.1 | Installazione di Git | 10 |
| 4.2.0.6.2 | Creazione di una cartella locale di repository | 11 |
| 4.2.0.6.3 | Creazione del branch personale | 11 |
| 4.2.1 | Norme | 11 |
| 4.2.1.1 | Repository | 11 |
| 4.2.1.1.1 | Nomi dei file in SiVoDim | 11 |
| 4.2.1.1.2 | Struttura di SiVoDim | 12 |
| 4.2.1.1.3 | Messaggio di commit | 12 |
| 4.2.2 | Strumenti | 12 |
| 4.2.2.1 | Git | 12 |
| 4.2.2.2 | GitHub | 12 |
| 4.2.2.3 | Dropbox | 12 |
| 4.2.2.4 | Sistema Operativo | 12 |
| 4.3 | Formazione dei membri del gruppo | 12 |

1 Introduzione

1.1 Scopo del documento

Questo documento definisce le norme che i membri del gruppo Stark Labs dovranno rispettare nello svolgimento del progetto "SiVoDiM". Ogni componente del gruppo è tenuto a leggere il documento e seguire le norme per raggiungere il miglior punto di incontro tra efficienza ed efficacia nello svolgimento delle attività. In questo modo viene garantita l'uniformità del materiale prodotto e vengono facilitate le operazioni di verifica. In particolare verranno specificate norme riguardanti:

- Interazioni tra membri del gruppo;
- Comunicazione verso l'esterno;
- Stesura dei documenti e convenzioni tipografiche;
- Organizzazione dell'ambiente di lavoro;
- Modalità di lavoro durante le fasi del progetto;
- Stesura del codice.

1.2 Scopo del progetto

Lo scopo del progetto è lo sviluppo di un applicativo per dimostrare efficacemente le potenzialità del motore di sintesi vocale FA-TTS_G. **//eventualmente completare dopo l'incontro con Giulio**

1.3 Glossario

Al fine di aumentare la comprensione del testo ed evitare eventuali ambiguità, viene fornito un glossario (Glossario v1.0) contenente le definizioni degli acronimi e dei termini tecnici utilizzati nel documento. Ogni vocabolo che ha un riferimento contenuto nel glossario è contrassegnato dal pedice "_G".

1.4 Riferimenti

1.4.1 Normativi

- Glossario v1.0.

1.4.2 Informativi

- Capitolato C1 – Actor base: A NoSQL Based on the Actor model
- Capitolato C1 – Actor base: A NoSQL Based on the Actor model
- Capitolato C1 – Actor base: A NoSQL Based on the Actor model

2 Processi primari

2.1 Processo di sviluppo

2.1.1 Analisi dei requisiti

2.1.1.1 Studio di fattibilità

2.1.1.2 Requisiti

2.1.1.3 Documentazione dei requisiti

2.1.1.4 Casi d'uso

2.1.1.5 Tracciamento dei requisiti

2.1.2 Progettazione

2.1.2.1 Diagrammi UML

2.1.2.2 Design pattern

2.1.2.3 Classi di verifica

2.1.2.4 Stile di progettazione

2.1.3 Progettazione architetture

2.1.4 Progettazione architetture di dettaglio

2.1.5 Codifica e convenzioni

2.1.5.1 Linguaggi di codifica

2.1.5.2 Framework e librerie

2.1.5.3 Convenzioni di codifica

2.1.5.4 File HTML

2.1.5.5 Nomenclatura

2.1.5.6 Intestazione di un file

2.1.5.7 Commenti

2.1.5.8 Esecuzione dei test di unità

3 Processi di supporto

3.1 Processo di documentazione

3.1.1 Template

3.1.2 Versionamento

3.1.3 Struttura dei documenti

3.1.3.1 Prima pagina

3.1.3.2 Registro delle modifiche

3.1.3.3 Indice

3.1.3.4 Formattazione delle pagine

3.1.4 Classificazione dei documenti

3.1.4.1 Documenti informali

3.1.4.2 Documenti formali

3.1.4.3 Verbale

3.1.5 Norme tipografiche

3.1.5.1 Punteggiatura

3.1.5.2 Stile del testo

3.1.5.3 Composizione del testo

3.1.5.4 Formati ricorrenti

3.1.5.5 Sigle

3.1.6 Componenti grafiche

3.1.6.1 Immagini

3.1.6.2 Tabelle

3.1.7 Procedure di avanzamento di un documento

3.2 Processo di verifica

3.2.1 Tecniche di analisi

3.2.1.1 Analisi statica

3.2.1.1.1 Walktrough

3.2.1.1.2 Inspection

3.2.1.2 Analisi dinamica

3.2.1.3 Test

3.2.1.3.1 Test di unità

3.2.1.3.2 Test di integrazione

3.2.1.3.3 Test di sistema

3.2.1.3.4 Test di regrassione

3.2.1.3.5 Test di accettazione

3.2.2 Verifica dei documenti

3.2.3 Verifica dei diagrammi

3.2.4 Verifica del codice

3.2.4.1 Analisi statica

3.2.4.2 Analisi dinamica

4 Processi organizzativi

4.1 Gestione dei processi

4.1.1 Attività

4.1.1.1 Gestione delle comunicazioni

4.1.1.1.1 Comunicazione interna Si utilizza Telegram_G per una comunicazione informale all'interno del gruppo, che inoltre fornisce il vantaggio di essere un'applicazione multi-piattaforma e disponibile anche in versione desktop/web.

4.1.1.1.2 Comunicazione esterna Il Project Manager_G sarà la persona preposta a mantenere i contatti con individui esterni al gruppo; per rappresentare il gruppo è stato creato il seguente indirizzo di posta elettronica:

starklabs.swe@gmail.com

Tutti i componenti del gruppo possono accedere alla casella, tuttavia solo il Project Manager_G si incaricherà di inviare le comunicazioni con questo indirizzo e-mail. Tutte le e-mail ricevute alla casella sopra indicata verranno automaticamente inoltrate ai membri del gruppo.

4.1.1.1.3 Composizione email

- **Destinatario:**

- **Esterno:** i destinatari possono essere il Proponente, Giulio Paci e l'azienda Mivoq, il Prof. Tullio Vardanega o il Prof. Riccardo Cardin.

- **Mittente:**

- **Esterno:** l'unico indirizzo utilizzabile è starklabs.swe@gmail.com e deve essere usato solamente dal Project Manager.

- **Oggetto:** l'oggetto deve contenere [UNIPD-TTS_G] e dovranno essere indirizzate all'attenzione del referente Giulio Paci, così come è stato specificato nel capitolato dell'azienda Proponente. Nel caso il messaggio sia una risposta è necessario aggiungere la particella "Re:" all'inizio dell'oggetto per distinguere il livello di risposta; se si dovesse trattare di un inoltro si deve usare la particella "I:". L'oggetto non va mai cambiato.
- **Corpo:** in caso di risposta da parte dell'azienda Mivoq o del fornitore risulta utile la citazione della frase a cui si intende rispondere. Il modello per citare correttamente deve seguire le seguenti regole. Devono essere presenti data e ora della mail a cui si risponde, il nome del mittente, il suo indirizzo e-mail tra parentesi angolari, ad esempio: <starklabs.swe@gmail.com>, la dicitura "ha scritto:" e infine il testo con una parentesi angolare chiusa prima, ">testo di prova". Se dovessero essere presenti alcune parti con uno o più destinatari specifici, il nome dovrà essere indicato all'inizio del paragrafo tramite la dicitura: @destinatario.
- **Allegati:** qualora vi sia la necessità è data la possibilità di allegare alcuni file al messaggio e-mail. Possono essere usati per allegare il verbale di incontri con proponente e committente o i punti più importanti citati in una comunicazione Telegram_G.

4.1.1.2 Gestione delle riunioni

4.1.1.2.1 Riunioni interne

- **Frequenza:** le riunioni del gruppo di lavoro avranno una frequenza settimanale;
- **Convocazione:** Il Responsabile di Progetto ha il compito di convocare le riunioni generali, a cui dovranno partecipare tutti i membri del gruppo. Su decisione del Responsabile di Progetto le riunioni possono coinvolgere anche solo specifici componenti del gruppo, a seconda del ruolo che si ritiene più utile in una data fase del progetto. Al termine di ogni riunione viene redatto un verbale. Il responsabile deve convocare l'assemblea, con almeno un giorno di preavviso, attraverso l'invio di una mail a starklabs.swe@gmail.com contenente:
 - **Oggetto:** convocazione riunione n. X, dove X indica il numero crescente di riunioni effettuate.
 - **Corpo:**
 - * **Data:** data e ora prevista;
 - * **Luogo:** luogo previsto;
 - * **Tipo:** ordinaria/straordinaria;
 - * **Ordine del giorno:** elenco numerato delle voci da esaminare.

Ogni componente del gruppo deve rispondere al messaggio in modo più celere possibile, confermando la presenza o giustificando un'eventuale assenza. Il Responsabile di Progetto, in mancata risposta di uno o più membri nel tempo utile, ha il compito di contattarli telefonicamente. Una volta ricevute le risposte e verificata l'assenza o presenza dei membri richiesti, il Responsabile di Progetto ha la possibilità di decidere se confermare, o posticipare la riunione per permettere la presenza di tutti i membri convocati; tutte le eventuali modifiche dovranno essere notificate tramite e-mail.

- **Verbale:** il verbale di riunione interna si presenta in forma di documento interno informale, per fissare i punti principali trattati e le soluzioni proposte. Verrà redatto come documento testuale utilizzando la funzione Notebooks_G di TeamWork_G permettendo così la sua condivisione, tra tutti i membri del gruppo, di un documento sempre aggiornato all'ultima modifica, dal segretario della riunione, ruolo scelto a rotazione tra i presenti. Sarà inoltre compito del segretario annotare ogni argomento trattato e controllare che venga seguito l'ordine del giorno.

4.1.1.2 Riunioni esterne

- **Convocazione:** vengono seguite le stesse dinamiche esposte per la comunicazione delle riunioni interne, per quanto sia auspicabile una riunione plenaria, eventuali assenze dei componenti del gruppo non causeranno posticipazioni o spostamenti delle date di incontro, dovendo ovviamente considerare gli impegni dell'azienda Proponente.
- **Verbale:** in caso di riunione con il committente od il proponente, il verbale è un documento che assume carattere ufficiale, e quindi redatto secondo uno schema specifico. Per agevolare la scrittura di tale documento viene utilizzato un template $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, per definire la struttura e organizzare i contenuti. Tale documento dovrà essere redatto e inviato come allegato in risposta all'e-mail di convocazione dell'assemblea e al Proponente Giulio Paci dal segretario scelto tra i membri presenti.

4.1.1.3 Gestione del sistema dei task_G Il sistema scelto per la gestione dei task_G è Teamwork_G, un'applicazione web di project management. Le viste presenti sono:

- **Dashboard:** dove vengono visualizzati i progetti attivi e le ultime notizie relative ad essi;
- **Everything:** che consente di visualizzare i task, le *milestone_G*, i file e filtrarli per data;
- **Project:** permette di visualizzare la lista di tutti i propri progetti suddivisi per categoria e ne permette l'accesso;
- **Calendar:** mostra un calendario per la gestione degli impegni e delle scadenze;
- **Statuses:** consente di verificare gli stati dei propri collaboratori al progetto;
- **People:** permette di visualizzare l'elenco dei singoli elementi del gruppo di lavoro e di accedere al loro profilo.

Le funzionalità principali e più importanti si hanno una volta avuto l'accesso al progetto desiderato e sono le seguenti:

- Aggiunta di nuovi task_G, ed eventualmente di *sub-task_G*, da associare ad uno o più membri del *team_G*;
- Assegnazione a ciascun task_G di una data di inizio e di termine di fattibilità;
- Aggiunta di nuove *milestone_G* e relativi dettagli come responsabile, descrizione e data di scadenza;
- *Upload_G* di file potenzialmente utili al gruppo di lavoro;
- Utilizzo di un blocco di note.

4.1.1.4 Gestione delle *milestone_G* Il *Responsabile di Progetto* dovrà pianificare i punti di controllo che il *team_G* deve raggiungere e assicurarsi che ogni task_G necessario al suo soddisfacimento venga terminato entro la data stabilita.

4.1.1.5 Gestione dei task_G Sarà compito del *Responsabile di Progetto* individuare ogni singolo task_G e, al seguito di un' accurata valutazione, assegnarlo al membro del gruppo più adatto. Dovrà inoltre assegnare una data di inizio e di scadenza di fattibilità. Tutte queste attività possono essere facilmente effettuate mediante l'interfaccia grafica di Teamwork_G.

4.1.1.6 Gestione dello svolgimento dei task_G Ogni membro del gruppo di lavoro è tenuto ad accettare il task_G assegnatigli dal *Responsabile di Progetto* e fare quanto possibile per portarlo a termine entro la data di scadenza. Nel caso in cui l'assegnatario non fosse in grado di adempiere al suo compito dovrà renderlo noto al *Responsabile di Progetto* entro 24 ore dall'assegnazione del task_G, altrimenti quest'ultimo verrà considerato come accettato; solo dopo un'accurata valutazione delle motivazioni riportate, il *Responsabile di Progetto* dovrà provvedere a trovare un nuovo destinatario del task.

4.1.2 Procedure

4.1.2.1 Generazione di una milestone Il *Responsabile di Progetto* dovrà eseguire i seguenti passi per generare una *milestone_G* dopo aver avuto accesso al progetto a Teamwork_G:

- Cliccare sul pulsante *Add a milestone*;
- Definire il titolo della *milestone_G*;
- Definire la data di scadenza;
- Assegnare un responsabile.

4.1.2.2 Assegnazione di un task Il *Responsabile di Progetto* dovrà eseguire i seguenti passi, riassunti in Figura 1, per generare una *task_G* dopo aver avuto accesso al progetto a Teamwork_G:

- Cliccare sul pulsante *Add a task*;
- Definire il titolo del *task_G*;
- Definire la data di inizio;
- Definire la data di scadenza;
- Assegnarlo ad uno o più membri del gruppo.

4.1.2.3 Svolgimento di un task Il membro assegnatario del *task_G*, ricevuta la notifica e non avendo alcun impedimento, dovrà procedere secondo le seguenti direttive che vengono anche riportate in figura 2:

- Se il *task_G* ricevuto ha una scadenza più immediata rispetto a quello su cui sta lavorando, dovrà sospendere lo svolgimento di quest'ultimo, metterlo in coda e dedicarsi al *task_G* appena notificato;
- Se, dopo aver iniziato lo svolgimento del *task_G*, si riceve la notifica di uno nuovo con scadenza più immediata si procederà come riportato nel punto precedente;
- Se si dovesse superare la data di scadenza prevista, si dovrà impostare il tag Delay dal sistema di Teamwork_G.

Questa situazione si può verificare se:

- Il tempo assegnato dal *Responsabile di progetto* non era sufficiente al completamento del *task_G*;
- Il *task_G* in ritardo sta alle dipendenze di un altro non ancora terminato;
- L'assegnatario ha rallentamenti esterni non resi noti al *Responsabile di Progetto*;
- Il committente non ha a disposizione tutte le conoscenze necessarie al corretto svolgimento del *task_G*.

È compito del *Responsabile di Progetto* fare in modo che i primi due casi non si verifichino.

- Al completamento del lavoro l'assegnatario dovrà spuntare il *task_G* dalla lista presente su Teamwork_G;
- Proseguire con lo svolgimento dei *task_G* rimanenti seguendo la procedura dal suo inizio.

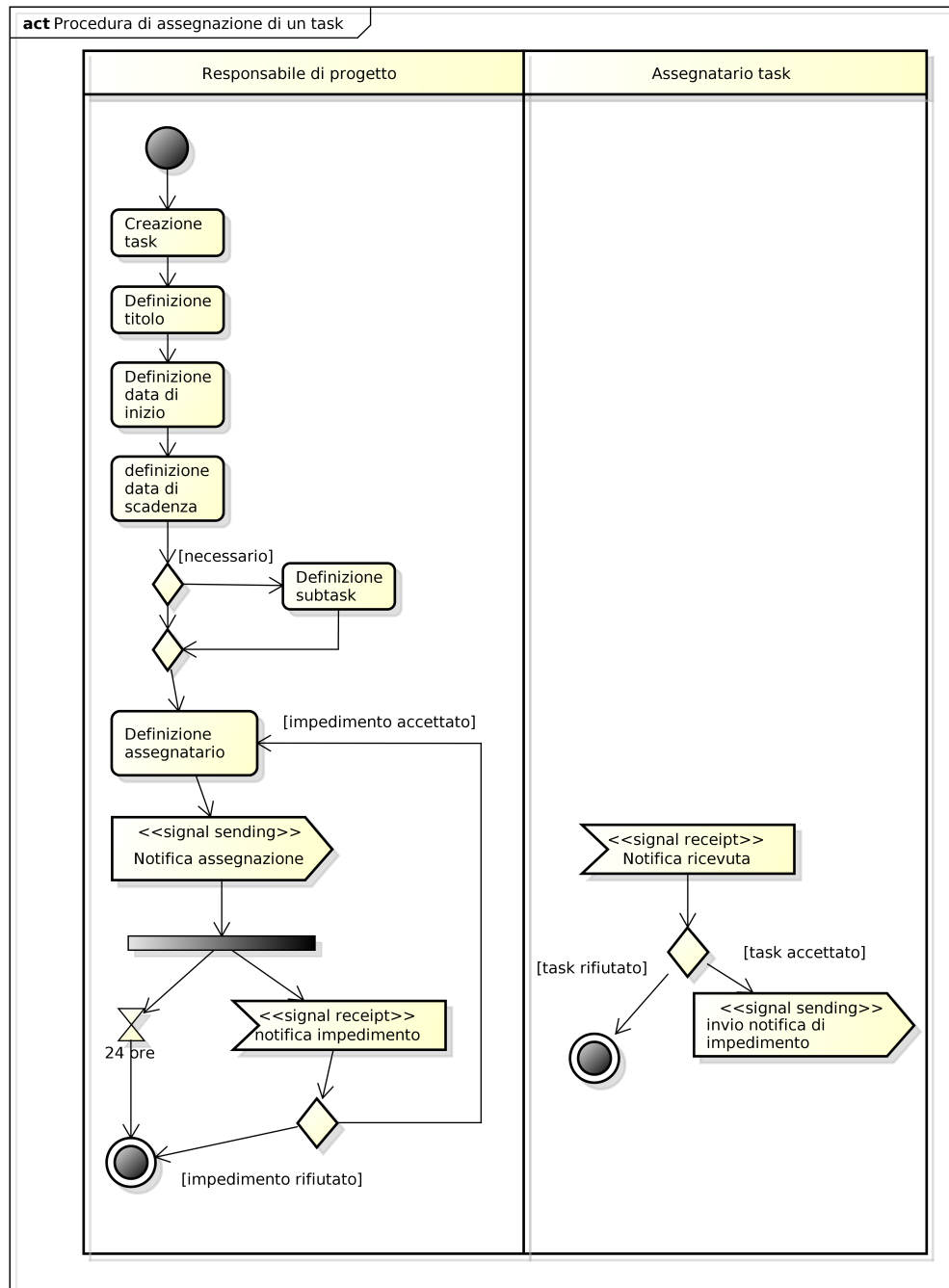


Figura 1: Diagramma di attività - Procedura di assegnazione di un task

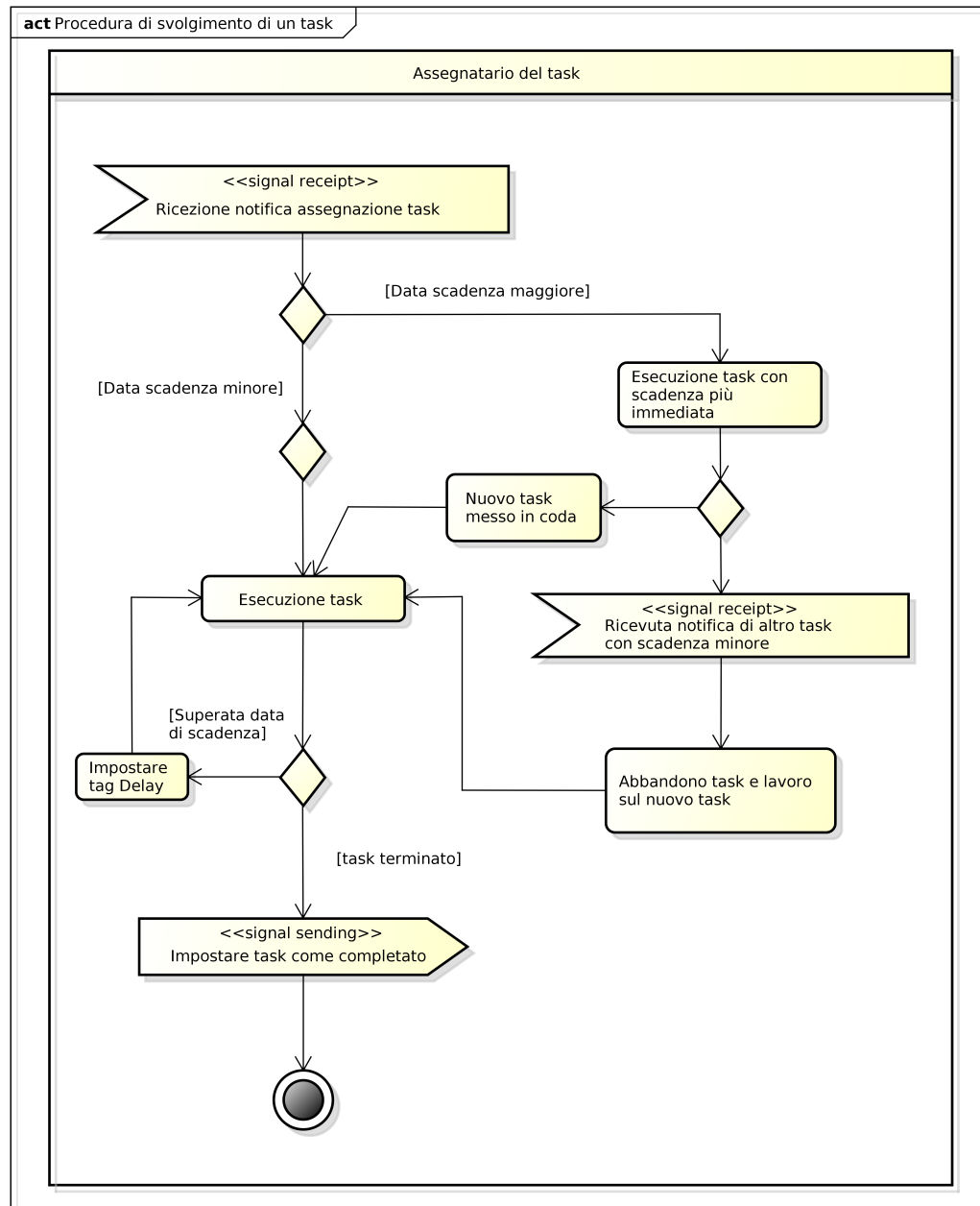


Figura 2: Diagramma di attività - Procedura di svolgimento di un task

4.1.2.4 Rilevazione dei rischi Sarà compito del *Responsabile di Progetto* individuare i rischi trovati nel *Piano di Progetto v1.0.0*.

Questa attività necessita di un continuo monitoraggio, in quanto è plausibile che insorgano nuovi rischi in seguito a quelli rilevati nella fase preliminare. In tal caso il *Responsabile di Progetto* dovrà agire come segue:

- Registrare il resoconto effettivo dei rischi nel *Piano di Progetto v1.0.0*;
- Pianificare per gestire i nuovi rischi;
- Aggiornare le metodologie per far fronte alla nuova pianificazione;
- Monitorare i nuovi rischi riscontrati durante lo sviluppo del progetto.

4.1.2.5 Ruoli di Progetto Ogni componente del gruppo *Stark Labs* dovrà ricoprire almeno una volta ciascuno dei ruoli necessari allo sviluppo del progetto.

Vengono ora presentate le diverse cariche, delineando per ciascuna le mansioni e le responsabilità.

4.1.2.5.1 Responsabile di Progetto Il *Responsabile di Progetto* rappresenta il *team_G* e il progetto verso il committente e il proponente; accentra le responsabilità di scelta e approvazione.

Detiene le seguenti responsabilità:

- Pianificazione e coordinamento delle attività;
- Gestione e controllo delle risorse;
- Analisi e gestione dei rischi;
- Approvazione dei documenti;
- assicurarsi che tutte le attività svolte siano conformi alle *Norme di Progetto v1.0.0* e rispettino la pianificazione effettuata nel *Piano di Progetto v1.0.0*.

4.1.2.5.2 Amministratore di Progetto L' *Amministratore di Progetto* deve svolgere i seguenti compiti:

- Assicurarsi che tutte le risorse siano presenti e operanti;
- Deve garantire un'infrastruttura funzionale;
- Fornire procedure che servono a garantire la qualità del prodotto uscente da un determinato compito.

4.1.2.5.3 Analista L' *Analista* deve:

- Tradurre il bisogno del cliente in una specifica utile a trovare una soluzione;
- Comprendere la complessità del problema;
- Capire il dominio nel quale lavora il cliente;
- Analizzare il dominio applicativo e le specifiche per poi produrre i documenti di analisi.

4.1.3 Norme

4.1.3.0.4 Progettista Il *Progettista* ha il compito di:

- Individuare la tecnologia più idonea per risolvere il problema indicato dall' *Analista*;
- Descrivere il funzionamento interno del sistema a diversi livelli di dettaglio;
- Produrre una soluzione comprensibile e attuabile.

4.1.3.0.5 Programmatore Il *Programmatore* ha responsabilità sulle attività di codifica perciò deve:

- Scrivere codice documentato, versionato e manutenibile;
- Implementare le soluzioni descritte dal *Progettista*;
- Implementare i test sul codice prodotto.

4.1.3.0.6 Verificatore Il *Verificatore* è il responsabile delle attività di verifica, ha quindi i seguenti compiti:

- Controllare che vengano rispettate le norme di progetto;
- Assicurarsi la conformità di ogni stadio del ciclo di vita del prodotto.

4.1.4 Strumenti

4.1.4.1 Teamwork *Teamwork_G* è l'applicazione web scelta per la gestione dei *task_G*; permette anche di gestire un calendario dove inserire note o fissare appuntamenti e o traguardi importanti.

4.1.4.2 Astah *Astah_G* è l'applicativo scelto per la creazione di grafici UML_G. La versione adottata è quella Professional, resa disponibile gratuitamente per un utilizzo da parte di studenti.

4.1.4.3 Telegram Si utilizza *Telegram_G* per una comunicazione informale all'interno del gruppo. Inoltre *Telegram_G* fornisce il vantaggio di essere un'applicazione multi-piattaforma e disponibile anche in versione desktop/web.

4.1.4.4 Microsoft Office Power Point *Power Point_G* è l'applicativo utilizzato per creare presentazioni.

4.2 Gestione delle infrastrutture

4.2.0.5 Attività

4.2.0.5.1 Gestione del repository Il gruppo ha deciso di utilizzare un repository_G utile a svolgere funzioni diverse, ma necessarie, allo sviluppo del sistema finale. Una volta iscritto ciascun membro avrà la possibilità di creare il suo *branch_G* personale contenente una copia dei file originali del *branch_G* master in modo da poterci lavorare in locale.

4.2.0.5.2 Gestione del messaggio di commit Per mantenere l'ambiente di lavoro il meno ambiguo possibile, è stato deciso di adottare un formato standard per andare a scrivere il messaggio della commit_G.

4.2.0.6 Procedure

4.2.0.6.1 Installazione di Git La procedura di installazione varia a seconda del sistema operativo utilizzato.

Per i sistemi Linux_G occorre seguire la seguente procedura:

- Aprire il terminale;
- Dare il comando `sudo apt-get update`;
- Dare il comando `apt-get install git`;

Per sistemi OS X_G :

- Recarsi nella sezione dedicata ai *download* <https://git-scm.com/download/mac>;
- Scaricare il file in formato DMG;
- Aprire il file appena scaricato;
- Lanciare l'installazione cliccando su `git.pkg`.

Infine per i sistemi Windows $_G$ fare quanto segue:

- Accedere al sito ufficiale <https://git-for-windows.github.io/>
- Scaricare l'eseguibile;
- Lanciare l'eseguibile;
- Seguire la procedura riportata dalla finestra di dialogo.

4.2.0.6.2 Creazione di una cartella locale di repository Seguire la seguente procedura:

- Creare una nuova cartella;
- Aprire il terminale;
- Collocarsi all'interno della cartella appena create;
- Eseguire il comando `git init`.
- Dare il comando `git clone <indirizzo>`, sostituendo con `<indirizzo>` l'URL del progetto su GitHub $_G$.

4.2.0.6.3 Creazione del branch personale Per creare il *branch* personale occorre seguire i seguenti passi:

- Muoversi nella cartella di repository $_G$.
- Accedere alla cartella di progetto;
- Eseguire il comando `git branch <nome>`, sostituendo a `<nome>` il nome del nuovo *branch* $_G$ da creare.

4.2.1 Norme

4.2.1.1 Repository

4.2.1.1.1 Nomi dei file in SiVoDim I file e le cartelle presenti nel repository $_G$ devono essere conformi al seguente formalismo tratto dallo Standard ISO $_G$ 9660:1999 (Level 2):

- I caratteri usati sono solo quelli minuscoli a-z, 0-9, l'underscore e il punto;
- Non sono ammessi caratteri accentati;
- I nomi non possono includere spazi o finire con un punto (.);
- I nomi non devono contenere più di un punto (.) ad eccezione di quelli che fanno riferimento ad una specifica versione;
- I nomi non devono essere più lunghi di 21 caratteri esclusi i 3 destinati all'estensione.

4.2.1.1.2 Struttura di SiVoDim Le cartelle nel repository_G verranno organizzate nel seguente modo a partire dalla root:

- **Documenti** nella quale sono presenti le seguenti cartelle corrispondenti alle varie fasi del processo di sviluppo:
 - **RR**: contenente i documenti e i file necessari alla revisione dei requisiti;
 - **RP**: contenente i documenti e i file necessari alla revisione di progettazione;
 - **RQ**: contenente i documenti e i file necessari alla revisione di qualifica;
 - **RA**: contenente i documenti e i file necessari alla revisione dei accettazione;

4.2.1.1.3 Messaggio di commit Il messaggio di commit_G dovrà essere conforme alla seguente notazione:

Desc:

Data:

Note:

dove:

- **Desc** fornisce una descrizione esaustiva dell'attività svolta;
- **Data** fornisce la data in cui si è apportata la modifica;
- **Note** aiuta a specificare lo stato del lavoro, nello specifico si adotteranno le seguenti notazioni:
 - C se il lavoro è stato completato;
 - NC se il lavoro non è stato completato;
 - V se il lavoro necessita de verifica;

4.2.2 Strumenti

4.2.2.1 Git Git_G è il sistema di controllo di versione utilizzato per il repository_G del team_G.

4.2.2.2 GitHub GitHub_G è il servizio web di hosting adottato per tenere una copia del repository_G del progetto.

4.2.2.3 Dropbox Dropbox_G è lo strumento di cloud_G che si è scelto di utilizzare per gestire file che non necessitano di essere sottoposti a controllo di versione.

4.2.2.4 Sistema Operativo I membri del gruppo operano su tre diversi sistemi operativi:

- Ubuntu_G;
- Windows10_G;
- Mac OS;

4.3 Formazione dei membri del gruppo

I membri del gruppo, per soddisfare le richieste assegnate dal *Responsabile di Progetto* al quale non sanno fare fronte con le conoscenze attuali in loro possesso, dovranno documentarsi adeguatamente durante ore esterne a quelle di lavoro, non imputabili perciò come costi al proponente_G.