



NORME DI PROGETTO

Gruppo Cyber13 - Progetto P2PCS

Informazioni sul documento

| | |
|-------------------------|--|
| Versione | 1.0.0 |
| Data Redazione | 09/03/2019 |
| Responsabili | Matteo Squeri Andrea Casagrande |
| Redazione | Andrea Casagrande Elena Pontecchiani Ilaria Rizzo |
| Verifica | Fabio Garavello Matteo Squeri Daniel Mirel Bira |
| Approvazione | Andrea Casagrande |
| Uso | Interno |
| Destinatari | Cyber13 Prof. Tullio Vardanega Prof. Riccardo Cardin |
| Mail di contatto | swe.cyber13@gmail.com |

Diario delle modifiche

| Versione | Data | Descrizione | Autore | Ruolo |
|----------|------------|---|-------------------|--------------|
| 1.0.0 | 04/04/2019 | Approvazione del documento per il rilascio in RR | Andrea Casagrande | Responsabile |
| 0.3.0 | 04/04/2019 | Verifica generale con esito positivo per il rilascio del documento in RR | Daniel Mirel Bira | Verificatore |
| 0.2.1 | 09/03/2019 | Approvazione delle norme per l'inizio della stesura degli altri documenti | Matteo Squeri | Responsabile |
| 0.2.0 | 09/03/2019 | Verifica con esito positivo per il rilascio del documento in RR | Fabio Garavello | Verificatore |
| 0.1.2 | 08/03/2019 | Contenuti rielaborati seguendo le direttive riguardanti la documentazione esposti nelle norme | Andrea Casagrande | Analista |

| | | | | |
|-------|------------|---|--|----------------|
| 0.1.1 | 07/03/2019 | Correzione errori di ortografia e di forma | Ilaria Rizzo | Analista |
| 0.1.0 | 07/03/2019 | Verifica con esito positivo per quanto riguarda i contenuti del documento | Fabio Garavello, Matteo Squeri | Verificatori |
| 0.05 | 06/03/2019 | Stesura sezione riguardanti i processi di organizzativi (sezione 4) | Ilaria Rizzo, Andrea Casagrande, | Amministratori |
| 0.04 | 05/03/2019 | Stesura sezione riguardanti i processi di supporto (sezione 3) | Elena Pontecchiani, Ilaria Rizzo, Andrea Casagrande, | Amministratori |
| 0.03 | 04/03/2019 | Stesura sezione riguardanti i processi primari (sezione 2) | Elena Pontecchiani, Ilaria Rizzo, Andrea Casagrande, | Amministratori |

| | | | | |
|------|------------|---|-----------------------|----------------|
| 0.02 | 04/03/2019 | Stesura sezione di introduzione (sezione 1) e sezione riguardanti i processi primari | Elena Pontecchiani | Amministratore |
| 0.01 | 04/03/2019 | Creazione scheletro documento | Elena Pontecchiani | Amministratore |

Indice

| | | |
|-----------|---|-----------|
| 1 | Introduzione | 9 |
| 1.1 | Scopo del documento | 9 |
| 1.2 | Scopo del prodotto | 9 |
| 1.3 | Glossario | 9 |
| 1.4 | Riferimenti | 10 |
| 1.4.1 | Riferimenti normativi | 10 |
| 1.4.2 | Riferimenti informativi | 10 |
| 2 | Processi primari | 11 |
| 2.1 | Processo di fornitura | 11 |
| 2.1.1 | Scopo | 11 |
| 2.1.2 | Aspettative | 11 |
| 2.1.3 | Attività | 11 |
| 2.1.3.1 | Studio di fattibilità | 11 |
| 2.1.3.2 | Piano di progetto | 12 |
| 2.1.3.3 | Piano di qualifica | 12 |
| 2.2 | Processo di sviluppo | 13 |
| 2.2.1 | Scopo | 13 |
| 2.2.2 | Aspettative | 13 |
| 2.2.3 | Attività | 13 |
| 2.2.3.1 | Analisi dei requisiti | 13 |
| 2.2.3.1.1 | Scopo | 13 |
| 2.2.3.1.2 | Aspettative | 13 |
| 2.2.3.1.3 | Classificazione dei requisiti | 14 |
| 2.2.3.1.4 | Classificazione dei casi d'uso | 14 |
| 2.2.3.2 | Progettazione | 15 |
| 2.2.3.2.1 | Scopo | 15 |
| 2.2.3.2.2 | Aspettative | 16 |
| 2.2.3.2.3 | Diagrammi | 16 |
| 2.2.3.3 | Codifica | 17 |
| 2.2.3.3.1 | Scopo | 17 |
| 2.2.3.3.2 | Aspettative | 17 |
| 2.2.3.3.3 | Stile di codifica | 17 |
| 2.2.3.3.4 | Convenzioni per la documentazione | 18 |
| 2.2.3.4 | Strumenti utilizzati | 19 |
| 3 | Processi di supporto | 20 |
| 3.1 | Documentazione | 20 |
| 3.1.1 | Scopo | 20 |
| 3.1.2 | Aspettative | 20 |
| 3.1.3 | Ciclo di vita della documentazione | 20 |
| 3.1.4 | Separazione tra documenti interni e esterni | 22 |
| 3.1.5 | Nomenclatura dei documenti | 22 |
| 3.1.5.1 | Versioni di un documento | 22 |
| 3.1.5.2 | Formato dei file | 22 |

| | | |
|-----------|---|----|
| 3.1.6 | Documenti correnti | 23 |
| 3.1.7 | Formattazione dei documenti | 23 |
| 3.1.7.1 | Strutturazione dei file | 23 |
| 3.1.7.2 | Norme tipografiche | 25 |
| 3.1.8 | Ambiente | 26 |
| 3.2 | Qualità | 26 |
| 3.2.1 | Scopo | 26 |
| 3.2.2 | Aspettative | 26 |
| 3.2.3 | Classificazione dei processi | 26 |
| 3.2.4 | Classificazione delle metriche | 26 |
| 3.2.5 | Procedure | 27 |
| 3.2.6 | Metriche per la qualità di processo | 27 |
| 3.2.6.1 | M[PROC][0001] - Schedule Variance | 27 |
| 3.2.6.2 | M[PROC][0002] - Budget Variance | 28 |
| 3.2.6.3 | M[PROC][0003] - Rischi non individuati | 28 |
| 3.2.6.4 | M[PROC][0004]: Numero campi dati per classe | 28 |
| 3.2.6.5 | M[PROC][0005] - Metodi per classe | 28 |
| 3.2.6.6 | M[PROC][0006] - Parametri per metodo | 28 |
| 3.2.6.7 | M[PROC][0007]: Grado di instabilità | 28 |
| 3.2.6.8 | M[PROC][0008] - Complessità Ciclomantica | 28 |
| 3.2.6.9 | M[PROC][0009] - Linee di codice per linee di comando | 29 |
| 3.2.6.10 | M[PROC][0010] - Halstead Difficulty per-function | 29 |
| 3.2.6.11 | M[PROC][0011] - Halstead Volume per-function | 29 |
| 3.2.6.12 | M[PROC][0012] - Halstead Effort per-function | 29 |
| 3.2.6.13 | M[PROC][0013] - Indice di manutenibilità | 29 |
| 3.2.6.14 | M[PROC][TM][0001] - Percentuale di codice coperto da test | 29 |
| 3.2.6.15 | M[PROC][TM][0002] - Percentuale di test passati | 30 |
| 3.2.6.16 | M[PROC][TM][0003] - Percentuale di test non passati | 30 |
| 3.2.7 | Metriche per la qualità di prodotto | 30 |
| 3.2.7.1 | Metriche per i documenti | 30 |
| 3.2.7.1.1 | M[PROD][D][0001]: Indice di <i>Gulpease</i> _g | 30 |
| 3.2.7.1.2 | M[PROD][D][0002]: Errori ortografici | 31 |
| 3.2.7.2 | Metriche per il Software | 31 |
| 3.2.7.2.1 | M[PROD][S][0001]: Copertura requisiti ob- bligatori | 31 |
| 3.2.7.2.2 | M[PROD][S][0002]: Copertura requisiti ac- cettati | 31 |
| 3.2.7.2.3 | M[PROD][S][0003]: Accuratezza rispetto al- le attese | 31 |
| 3.2.7.2.4 | M[PROD][S][0004]: Densità di <i>failure</i> _g | 32 |
| 3.2.7.2.5 | M[PROD][S][0005]: Blocco di operazioni non corrette | 32 |
| 3.2.7.2.6 | M[PROD][S][0006]: Tempo di risposta | 32 |
| 3.2.7.2.7 | M[PROD][S][0007]: Comprensibilità delle fun- zioni offerte | 32 |

| | | |
|------------|---|-----------|
| 3.2.7.2.8 | M[PROD][S][0008]: Facilità di apprendimen- to delle funzionalità | 33 |
| 3.2.7.2.9 | M[PROD][S][0009]: Utilizzo effettivo delle funzionalità | 33 |
| 3.2.7.2.10 | M[PROD][S][0010]: Capacità di analisi di failure | 33 |
| 3.2.7.2.11 | M[PROD][S][0011]: Impatto delle modifiche | 33 |
| 3.2.7.2.12 | M[PROD][S][0012]: Rapporto linee di com- mento su linee di codice | 33 |
| 3.2.7.2.13 | M[PROD][S][0013]: Versioni di Android sup- portate | 34 |
| 3.2.7.2.14 | M[PROD][S][0014]: Copertura del framework <i>Octalysis_g</i> | 34 |
| 3.3 | Configurazione | 34 |
| 3.3.1 | Scopo | 34 |
| 3.3.2 | Aspettative | 34 |
| 3.3.3 | Controllo di versione | 34 |
| 3.3.3.1 | Descrizione | 34 |
| 3.3.3.2 | Struttura delle repository | 34 |
| 3.3.3.3 | Branch | 35 |
| 3.3.3.4 | Aggiornamento della repository | 35 |
| 3.3.4 | Strumenti | 36 |
| 3.4 | Verifica | 36 |
| 3.4.1 | Scopo | 36 |
| 3.4.2 | Aspettative | 36 |
| 3.4.3 | Descrizione | 36 |
| 3.4.4 | Attività | 36 |
| 3.4.4.1 | Analisi statica | 36 |
| 3.4.4.2 | Analisi dinamica | 37 |
| 3.4.4.2.1 | Test | 37 |
| 3.4.5 | Strumenti | 37 |
| 3.5 | Validazione | 37 |
| 3.5.1 | Scopo | 37 |
| 3.5.2 | Aspettative | 38 |
| 3.5.3 | Procedure | 38 |
| 4 | Processi organizzativi | 39 |
| 4.1 | Processi di coordinamento | 39 |
| 4.1.1 | Scopo | 39 |
| 4.1.2 | Aspettative | 39 |
| 4.1.3 | Comunicazione | 39 |
| 4.1.3.1 | Comunicazioni interne | 39 |
| 4.1.3.2 | Comunicazioni esterne | 39 |
| 4.1.4 | Riunioni | 40 |
| 4.1.4.1 | Riunioni interne | 40 |
| 4.1.4.2 | Riunioni esterne | 40 |

| | | | |
|-----|---------|---|----|
| | 4.1.4.3 | Verbale di riunione | 41 |
| 4.2 | | Processi di pianificazione | 42 |
| | 4.2.1 | Scopo | 42 |
| | 4.2.2 | Aspettative | 42 |
| | 4.2.3 | Ruoli di progetto | 42 |
| | 4.2.3.1 | Responsabile di progetto | 42 |
| | 4.2.3.2 | Amministratore | 42 |
| | 4.2.3.3 | Analista | 43 |
| | 4.2.3.4 | Progettista | 43 |
| | 4.2.3.5 | Programmatore | 43 |
| | 4.2.3.6 | Verificatore | 43 |
| | 4.2.3.7 | Rotazione dei ruoli | 43 |
| | 4.2.4 | Ticketing | 44 |
| | 4.2.4.1 | Task list | 44 |
| | 4.2.4.2 | Task | 44 |
| | 4.2.4.3 | Ticket | 44 |
| 4.3 | | Procedure | 45 |
| | 4.3.1 | Creazione e gestione dei task | 45 |
| | 4.3.2 | Gestione dei ticket | 45 |
| | 4.3.3 | Stesura del consuntivo | 47 |
| 4.4 | | Strumenti | 47 |
| | 4.4.1 | Pianificazione | 47 |
| | 4.4.2 | Comunicazione | 47 |
| | 4.4.3 | Creazione diagrammi di Gantt | 48 |
| | 4.4.4 | Calcolo del consuntivo | 48 |
| 4.5 | | Formazione | 48 |

Elenco delle figure

| | | |
|---|--|----|
| 1 | Diagramma del ciclo di vita della documentazione | 21 |
| 2 | Diagramma del ciclo di vita di un ticket | 46 |

1 Introduzione

1.1 Scopo del documento

Il documento ha lo scopo di fissare precise norme che regoleranno l'intero svolgimento del progetto.

Tutti i membri del team sono obbligati a visionare tale documento e a sottostare alle norme ivi descritte. Le norme contenute nel documento si occupano di regolamentare i diversi processi all'interno del progetto. La regolamentazione tramite norme permette di sviluppare prodotti coerenti e consistenti tra i vari componenti del gruppo e facilitare l'esecuzione delle operazioni di verifica. In particolare le norme si occuperanno di regolamentare:

- Le modalità di interazione tra i membri del team e le persone esterne al team;
- L'organizzazione del team: verranno definiti e assegnati ruoli ai membri del team. Per ogni ruolo verranno identificate determinate mansioni;
- Le convenzioni tipografiche e le modalità di stesura delle documentazioni;
- Gli ambienti di sviluppo, il *repository*_[g] e il *ticketing*_[g];

Nel caso in cui ve ne sia necessità, ogni membro del team potrà contattare il *Project Manager*_[g] per suggerire eventuali cambiamenti

1.2 Scopo del prodotto

Lo scopo del prodotto è quella di realizzare un'applicazione *Android*_[g] che implementi un servizio di car sharing *peer-to-peer*_[g].

1.3 Glossario

Onde evitare ambiguità o incomprensioni di natura lessicale, si allega il *Glossario v1.0.0*. All'interno del documento saranno presenti parole di ambito specifico, uso raro che potrebbero creare incomprensioni. Per una maggiore leggibilità tali parole sono riconoscibili all'interno dei vari documenti in quanto scritte in corsivo e con un 'g' a pedice tra barre orizzontali (per esempio *Glossario*_[g]). Nel caso esistano più ripetizioni di una stessa parola del glossario all'interno di uno stesso paragrafo, la dicitura con la lettera 'g' a pedice sarà inserita solo la prima volta che la parola comparirà. Il comando LaTeX da utilizzare per contrassegnare un termine da glossario all'interno dei documenti è `\citgl{..}`.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- *Norme di Progetto v1.0.0*;

1.4.2 Riferimenti informativi

- *Analisi Dei Requisiti v1.0.0*;
- *Piano di Progetto v1.0.0*;
- *Piano di Qualifica v1.0.0*;
- *Studio di Fattibilità v1.0.0*;
- Sito del corso di Ingegneria del Software:
<https://www.math.unipd.it/~tullio/IS-1/2018/>
- Software Engineering - Ian Sommerville - 10th Edition;
- Materiale suggerito dalla proponente:
 - Materiale $MVP_{|g|}$:
<https://model-view-presenter-android-guidelines>
 - Materiale Testing e *Refactoring*_{|g|}:
https://www.youtube.com/watch?v=_NnElP05BU0

2 Processi primari

2.1 Processo di fornitura

2.1.1 Scopo

In questa sezione vengono sancite precise norme per quanto riguarda il rapporto di fornitura con la proponente *GaiaGo_[g]* e dei committenti Prof. Tullio Vardanega e Prof. Riccardo Cardin nell'ambito della progettazione, sviluppo e consegna del prodotto P2PCS.

Le norme ivi trattate sono tassative e i membri del gruppo Cyber13 sono tenuti a rispettarle al fine di proporsi e diventare fornitori.

2.1.2 Aspettative

Il gruppo intende mantenere un costante dialogo con il proponente per avere un riscontro efficace sul lavoro svolto.

Nel corso dell'intero progetto il team Cyber13 si auspica di instaurare con la committente GaiaGo, nel particolare con il referente Filippo Pretto, un rapporto di collaborazione al fine di:

- Determinare come adempire appieno ai requisiti fissati dal proponente;
- Determinare vincoli sui processi e i requisiti;
- Eseguire una stima dei costi;
- Concordare la qualifica del prodotto.

2.1.3 Attività

2.1.3.1 Studio di fattibilità

In seguito alla formazione ufficiale dei gruppi del secondo lotto, avvenuta in data 4 Marzo 2019, è stata convocata una riunione interna al gruppo per discutere in merito ai vari capitolati disponibili per lo svolgimento del progetto, presentati in data 16 Novembre 2018.

Una volta stabilita la scelta del capitolato per il quale proporsi come fornitori, gli analisti hanno condotto un'ulteriore e approfondita attività di analisi dei rischi e delle opportunità culminata con la redazione del documento *Studio di Fattibilità v1.0.0*. Tale documento include le motivazioni che hanno portato il gruppo Cyber13 a proporsi come fornitore per il prodotto indicato e riporta per ciascun capitolato:

- **Informazioni sul capitolato:** In cui vengono riportati nome del progetto, azienda proponente e i committenti.
- **Descrizione:** All'interno della quale viene descritto brevemente cosa il progetto richiede.

- **Studio del dominio:** Nel quale vengono indicati i domini applicativi e tecnologici del capitolato.
- **Esito finale:** Dove sono riportati aspetti positivi, fattori di rischio e conclusione finale del gruppo in merito al capitolato.

2.1.3.2 Piano di progetto

Il responsabile avrà il compito di redigere il documento *Piano di Progetto v1.0.0*. Tale documento formale descrive:

- Gli obiettivi del progetto.
- Le attività da svolgere per adempiere agli obiettivi rispettando i requisiti fissati. Attraverso l'attività di pianificazione si organizzano le diverse attività da svolgere con precise tempistiche da rispettare.
- Le risorse coinvolte (in termini di personale e di tecnologie impiegate).
- I costi previsti, che vengono stabiliti attraverso il preventivo e il consuntivo. Sulla base della pianificazione si stima la quantità di lavoro necessaria per ogni fase, proponendo così un preventivo per il costo totale del progetto. Alla fine di ogni attività si redige inoltre un consuntivo di periodo per tracciare l'andamento rispetto a quanto preventivato;
- I rischi potenziali attraverso l'analisi dei rischi. Si analizzano nel dettaglio i rischi che potrebbero insorgere nel corso del progetto e i modi per affrontarli, capendo la probabilità che essi accadano e il livello di gravità ad essi associato.

2.1.3.3 Piano di qualifica I verificatori dovranno scegliere una strategia da adottare per la *verifica_[g]* e la *validazione_[g]* del materiale prodotto dal gruppo. Il documento dovrà contenere:

- **Visione generale delle strategie di verifica:** Si stabiliscono le procedure di controllo sulla qualità di processo e di prodotto, tenendo in considerazione le risorse a disposizione.
- **Misure e metriche:** Si devono stabilire delle metriche oggettive per i documenti, i processi e il software.
- **Gestione della revisione:** Si devono stabilire le modalità di comunicazione delle anomalie e le procedure di controllo per la qualità di processo.
- **Pianificazione del collaudo:** Si devono definire nel dettaglio le metodologie di collaudo del prodotto realizzato.
- **Resoconto delle attività di verifica:** Alla fine di ogni attività si devono riportare le metriche calcolate e un resoconto sulla verifica di tale attività.

2.2 Processo di sviluppo

2.2.1 Scopo

Il processo di sviluppo contiene tutte le attività di analisi, design, codifica, integrazione ed installazione relative al prodotto da sviluppare. Di seguito sono raccolte le linee guida che verranno utilizzate dai membri del gruppo nelle principali attività di questo processo.

Pertanto il processo si compone delle seguenti attività:

- Analisi dei requisiti;
- Progettazione;
- Codifica.

2.2.2 Aspettative

Gli obiettivi perseguiti in fase di sviluppo sono i seguenti:

- Realizzare un prodotto finale conforme alle richieste del proponente;
- Realizzare un prodotto finale soddisfacente i test di verifica e di validazione.

2.2.3 Attività

2.2.3.1 Analisi dei requisiti

2.2.3.1.1 Scopo

L'analisi dei requisiti è un documento ad uso esterno, il cui compito è fornire una descrizione completa di tutti i *requisiti*_[g] individuati in fase di analisi dagli analisti e dei casi d'uso riguardanti il progetto *P2PCS*_[g].

Le informazioni estrapolate sono state ottenute dall'esposizione del capitolato e soprattutto attraverso incontri con il referente dell'azienda proponente.

In seguito sono elencate le norme che riguardano i requisiti esposti nel documento *Analisi Dei Requisiti v1.0.0*, al fine di evitare ambiguità o difformità di natura formale.

La tecnica utilizzata per l'analisi e la ricerca dei requisiti è quella dei casi d'uso.

2.2.3.1.2 Aspettative

L'obiettivo dell'attività è la creazione della documentazione formale contenente tutti i requisiti richiesti dal proponente.

2.2.3.1.3 Classificazione dei requisiti

Ad ogni requisito individuato in fase di analisi viene identificato un codice univoco, espresso nel seguente formato:

$$R[\text{Priorità}][\text{Tipo}][\text{Codice}]$$

Dove:

- **R:** abbreviazione di requisito.
- **Priorità:** Indica l'importanza di un requisito, e può assumere i seguenti valori (mutuamente esclusivi ed espressi in ordine decrescente di importanza):
 - C: Dall'inglese 'compulsory', che significa obbligatorio, indica un requisito irrinunciabile per il committente.
 - D: Dall'inglese 'desirable', desiderabile. Requisito auspicabile ma non strettamente necessario.
- **Tipo:** Indica la classificazione del vincolo. Può assumere i seguenti valori:
 - F: Indica un requisito funzionale, ovvero la definizione di una funzione/caratteristica che deve essere implementata in un sistema. Questa tipologia di requisiti descrivono quindi come il software reagisce a situazioni ed input particolari;
 - Q: Indica un requisito di qualità. Includono i requisiti di *efficacia*_{|g|}, *efficienza*_{|g|} e i requisiti per garantire la qualità del prodotto;
 - V: Indica requisiti di vincolo imposti dalla proponente GaiaGo.
- **Codice:** Numero intero univoco e incrementale di tre cifre associato al particolare requisito sulla base dell'ordine in cui compare nella struttura del documento.

2.2.3.1.4 Classificazione dei casi d'uso

Gli Analisti hanno inoltre il compito di individuare i diversi *Casi d'uso*_{|g|}, elencandoli attraverso una strategia *Top down*_{|g|}, ovvero partendo dal generale e in seguito scendendo nel particolare.

I casi d'uso hanno lo scopo di descrivere scenari di interazione tra utenti e un sistema. Ciascun caso d'uso presente nel documento di Analisi dei Requisiti sarà corredato delle seguenti informazioni:

- **Codice identificativo e nome:** Ogni caso d'uso è identificato da un codice univoco strutturato nel seguente modo:

$$UC [\text{Codice Padre}][\text{Codice Figlio}]\text{-Nome}$$

Dove:

- UC: Indica "User case", caso d'uso in inglese;
- Codice Padre: Numero intero;

- Codice Figlio: Uno o più numeri interi che specificano l’annidamento del caso;
- Nome del caso d’uso.
- **Attori:** Indica gli attori principali (obbligatoriamente) e secondari (opzionalmente, se esistono) del caso d’uso.
- **Scopo e descrizione:** Riporta una breve descrizione del caso d’uso.
- **Scenario principale:** Descrizione di ciò che il caso d’uso vuole modellare, indicando ogni azione che ne fa parte.
- **Precondizione:** Specifica le condizioni che sono identificate come vere prima del verificarsi degli eventi del caso d’uso.
- **Postcondizione:** Specifica le condizioni che sono identificate come vere dopo il verificarsi degli eventi del caso d’uso.
- **Scenari alternativi (opzionale):** Descrizione di una possibilità alternativa allo scenario principale.
- **Inclusioni (opzionale):** Usate per non descrivere più volte lo stesso flusso di eventi, inserendo il comportamento comune in un caso d’uso a parte.
- **Estensioni (opzionale):** Descrivono i casi d’uso che non fanno parte del flusso principale degli eventi, allo stesso modo di quanto descritto in “Scenario principale”.

Faremo inoltre uso dei *diagrammi dei casi d’uso*_[g], descritti nel linguaggio *UML 2.0*_[g], per mettere in evidenza gli attori ed i servizi del sistema.

2.2.3.2 Progettazione

2.2.3.2.1 Scopo

L’attività di progettazione precede la fase di realizzazione ed è svolta dai progettisti. Essi hanno il compito di sviluppare e documentare una visione architeturale ad alto livello del prodotto identificando componenti chiare, riusabili e coese rimanendo nei costi fissati. Nel processo di descrizione dell’*architettura*_[g] saranno fissate le componenti che andranno a costituire il sistema, con relative dipendenze e interazioni tra le loro istanze. I progettisti definiranno inoltre le interfacce necessarie a consentire l’interazione tra componenti e i *design pattern*_[g] da utilizzare.

Durante questa attività è molto importante che i progettisti si confrontino con l’azienda proponente, allo scopo di ottenere feedback e consigli. L’attività di progettazione è documentata nella *Technology Baseline*_[g] e nella *Product Baseline*_[g] che saranno consegnati rispettivamente alla Revisione di Progettazione e alla Revisione di Qualifica.

L’architettura definita avrà i seguenti obiettivi:

- Soddisfare i requisiti individuati in fase di analisi e riportati nel documento *Analisi Dei Requisiti v1.0.0* e adattarsi in caso questi evolvano o se ne aggiungano di nuovi.
- Essere robusta, in modo tale da essere in grado di gestire situazioni impreviste.
- Ridurre al minimo i tempi di manutenzione.
- Essere sicura, in modo da sapersi difendere in caso di errori o intrusioni indesiderate.
- Essere in grado di rispettare le specifiche nel tempo.
- Essere composta da componenti semplici e che minimizzino il grado di dipendenza l'uno dall'altro.

L'attività di progettazione deve rispettare i requisiti ed i vincoli stabiliti tra il gruppo e il proponente. In questa fase i progettisti pongono i seguenti obiettivi:

- Garantire la correttezza del prodotto sviluppato, perseguendo la correttezza per costruzione.
- Organizzare e ripartire compiti implementativi, riducendo la complessità del problema originale fino alle singole componenti facilitandone la codifica da parte dei singoli programmatori.
- Ottimizzare l'uso delle risorse disponibili.

2.2.3.2.2 Aspettative

L'attività di progettazione consiste nel descrivere una soluzione del problema che sia soddisfacente per tutti gli *stakeholders*_[g].

2.2.3.2.3 Diagrammi

Al fine di rendere più chiare le scelte progettuali adottate e ridurre le possibili ambiguità, si farà ricorso a vari tipi di diagrammi UML 2.0. In particolare:

- **Diagrammi dei casi d'uso:** Dedicati alla descrizione delle funzioni offerte dal sistema. Saranno il tipo di diagrammi utilizzati in fase preliminare alla RR.
- **Diagrammi delle classi:** Dedicati alla descrizione degli oggetti che fanno parte di un sistema e delle loro dipendenze.
- **Diagrammi dei package:** Dedicati alla descrizione della dipendenza tra classi raggruppate in package.
- **Diagrammi di sequenza:** Dedicati a descrivere la collaborazione nel tempo tra un gruppo di oggetti.
- **Diagrammi di attività:** Dedicati a descrivere la logica procedurale.

2.2.3.3 Codifica

2.2.3.3.1 Scopo

Questa attività ha come scopo l'effettiva realizzazione del prodotto software richiesto, rispettando le metriche stabilite nel documento *Piano di Qualifica v1.0.0*. In questa fase si concretizza la soluzione attraverso la programmazione, in modo da ottenere il prodotto software finale.

Durante l'attività di codifica i programmatori devono seguire le linee-guida ivi indicate, al fine di rendere il codice più uniforme e leggibile per favorire le fasi manutenzione, verifica e validazione, e di conseguenza migliorare la qualità del prodotto.

2.2.3.3.2 Aspettative

Obiettivo dell'attività è la creazione di un prodotto software conforme alle richieste prefissate con il proponente.

2.2.3.3.3 Stile di codifica

- **Indentazione:** Si richiede l'utilizzo di esattamente una tabulazione per ogni blocco di istruzioni.
- **Parentesi dei costrutti:** Si richiede inserire la parentesi di apertura di un blocco in linea, mentre quella di chiusura allineata alla prima e in una linea a essa dedicata;
- **Convenzioni per i nomi:** si seguiranno le usuali convenzioni sancite dal linguaggio Java:
 - Nomi di variabili, metodi e funzioni devono avere la prima lettera minuscola. Eventuali parole dopo la prima che dovessero comporre il nome avranno la prima lettera maiuscola.
 - I nomi delle classi devono avere la prima lettera maiuscola. Eventuali parole dopo la prima che dovessero comporre il nome avranno la prima lettera maiuscola.
 - I nomi delle classi devono avere la prima lettera maiuscola.
 - Se possibile alcuni caratteri all'interno dei nomi sono da evitare in quanto facilmente confondibili con i numeri 1 e 0, elencati di seguito:
 - * l: lettera minuscola elle;
 - * O: lettera maiuscola o;
 - * I: lettera maiuscola i.
 - Tutti i nomi devono essere unici ed esplicativi al fine di evitare il più possibile ambiguità e incomprensioni.

- **Commenti:** Il programmatore è invitato ad inserire commenti ogniqualvolta li ritenga utili per la comprensione del codice prodotto.
- **Funzioni pure:** Onde evitare effetti collaterali dovrebbero essere l'unico tipo di procedure da utilizzare.
- **Codice requisito:** Se lo scopo della funzione oppure della classe è soddisfare la richiesta di un requisito, va specificato il suo codice.
- **Lunghezza delle funzioni:** Ogni procedura non dovrebbe superare le 50 righe di codice, approssimativamente equivalenti al numero di righe visualizzabili contemporaneamente su uno schermo.
- **Code tags:** sono delle parole chiavi informali all'interno dei commenti per segnalare dei problemi, note o delle parti di codice ancora da fare. La struttura è la seguente:

```
// <tag>:<descrizione in una singola linea>
```

I tag potranno essere usati all'interno di commenti blocco o inline. Sono supportati da molti IDE ed editor moderni. I tipi di tag sono:

- TODO: qualcosa che va completato.
- NOTE: annotazioni particolari.
- FIXME: qualcosa che va risolto perchè non funzionante.
- HACK: qualcosa che funziona ma andrebbe migliorato.

2.2.3.3.4 Convenzioni per la documentazione

- **Intestazione:** Ogni file contenente codice deve avere la seguente intestazione contenuta in un commento e posta all'inizio del file stesso:

```
File:nome del file;  
Version: versione del file nella forma X.Y descritta  
        in seguito;  
Type: tipo del file;  
Date: data di creazione del file;  
Author: autore del file;  
  
License: tipo licenza del file;  
  
Advice: lista avvertenze e limitazioni legate al file;  
  
Changelog: registro modifiche strutturato come:  
          Author || Data || Description
```

Per quanto riguarda la versione del file, essa sarà rappresentata nella seguente forma: X.Y, dove X, Y sono numeri interi che, rispettivamente, rappresentano l'indice di una versione principale e l'indice di una modifica parziale. L'incremento del valore X rappresenta un avanzamento della versione stabile e implica l'azzeramento dell'indice Y. L'incremento dell'indice Y rappresenta una verifica o una modifica rilevante all'interno del documento (per esempio l'aggiunta o la rimozione di una o più istruzioni).

La versione 1.0 deve rappresentare la prima versione del file completo e stabile, cioè quando le sue funzionalità obbligatorie sono state definite e si considerano funzionanti. Solo dalla versione 1.0 è possibile testare il file, con degli appositi test predefiniti, per validarne la qualità.

2.2.3.4 Strumenti utilizzati Di seguito sono elencati gli strumenti utilizzati dal team Cyber13 in fase di sviluppo. Gli elementi indicati con * sono stati identificati a seguito di un'analisi preliminare. Tali strumenti potrebbero cambiare in caso di necessità differenti da quanto pianificato.

- **Creazione diagrammi UML:** Per la produzione dei diagrammi UML viene utilizzato *Draw.io*_[g], idoneo dato il funzionamento sul cloud e la condivisione diretta su *Google Drive*_[g];
- **IDE:** Si utilizza *Android Studio*_[g] per la codifica in Java e Kotlin.
- **Realizzazione Slide di presentazione:** per preparare le slide di presentazione del progetto il team utilizza lo strumento Presentazioni Google, che realizza il file di presentazione direttamente all'interno del Google Drive associato alla mail del gruppo e permette ad ogni membro del team di apportare modifiche e aggiornamenti anche da remoto.
- **Continuous Integration:** Implementata attraverso la piattaforma *Bitrise*_[g].

3 Processi di supporto

3.1 Documentazione

3.1.1 Scopo

In questa sezione verranno descritti gli standard ai quali i membri del team si attengono per quanto riguarda la stesura, la *verifica_[g]* e l'approvazione dei documenti redatti. Le norme ivi indicate sono tassative per tutti i documenti formali presentati, esplicitamente riportati nella sezione "Documenti correnti".

3.1.2 Aspettative

Il gruppo desidera consegnare documentazione formale coerente. Per conseguire tale fine vengono seguite pedissequamente le norme relative alla documentazione ivi indicate.

3.1.3 Ciclo di vita della documentazione

Ogni documento formale presentato dovrà superare con esito positivo le seguenti fasi:

- **Sviluppo:** Comprende le decisioni in merito alla strutturazione dei contenuti da comprendere nel documento e la stesura degli stessi. Il documento concluso in questo stadio è non formale.
- **Verifica:** Quando si ritiene conclusa la stesura di un documento non formale, viene chiamato in causa il Responsabile di progetto. È sua responsabilità assegnare il documento ai Verificatori, la cui incombenza è svolgere l'attività di verifica. Quest'ultima consiste nel controllare la correttezza formale del documento. Al termine del controllo gli esiti possibili sono due:
 - Esito negativo da parte dei verificatori: Nel caso in cui i Verificatori individuino degli errori o delle difformità nel documento; sarà loro dovere comunicare le proprie valutazioni al Responsabile di progetto. Quest'ultimo riassegnerà il documento ai redattori, che ripeteranno la stesura del documento o delle sue parti ritenute incorrette. Il ciclo si ripete finché i Verificatori non hanno più segnalazioni.
 - Esito positivo da parte dei verificatori: I verificatori non individuano difformità. Il documento entra in fase di approvazione da parte del Responsabile di progetto.
- **Approvazione:** Segue l'esito positivo della fase di verifica. Successivamente il documento verrà quindi passato al Project Manager, che sarà responsabile dell'approvazione o meno del documento. Gli esiti possibili sono due:

- Se il documento non viene considerato adeguato al rilascio il Responsabile di progetto comunicherà ai redattori le modifiche da apportare. In casi estremi, potrà imporre che la stesura del documento dovrà essere rifatta nella sua totalità.
- Se il documento verrà approvato lo si riterrà un documento formale, e potrà essere distribuito alle persone nominate nella lista di distribuzione.

A seguire è presente un diagramma rappresentante una visione schematica del ciclo di vita sopra descritto:

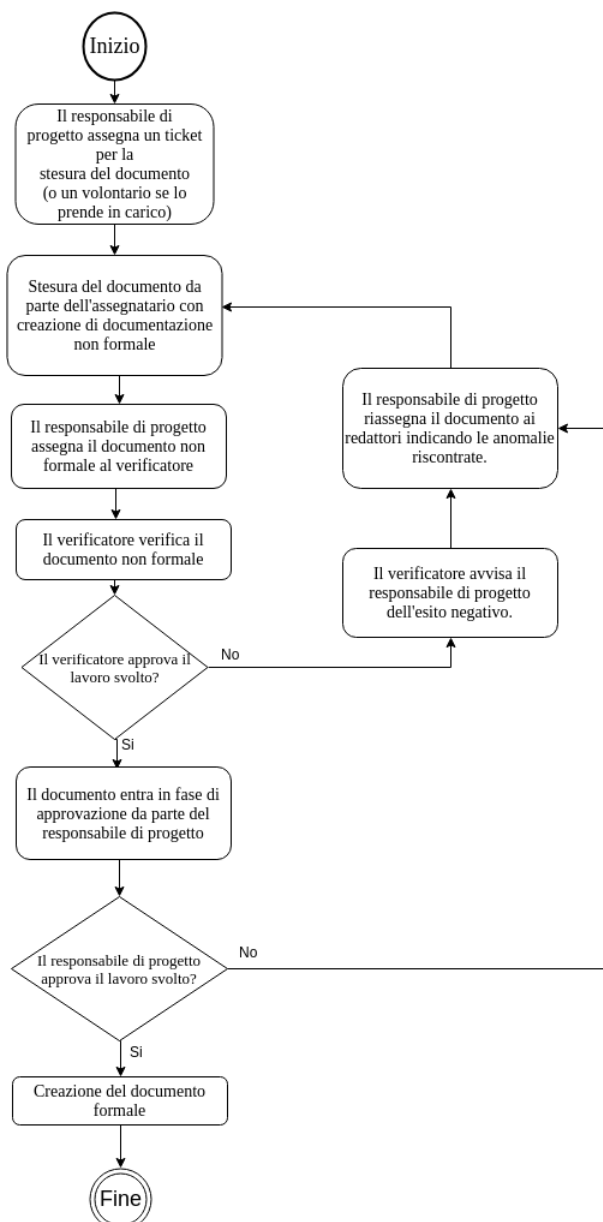


Figura 1: Diagramma del ciclo di vita della documentazione

3.1.4 Separazione tra documenti interni e esterni

I documenti formali redatti dal gruppo Cyber13 rientreranno tassativamente in una delle seguenti categorie (mutuamente esclusive):

- **Documenti interni** : Documenti ad uso esclusivo del team Cyber13, redatti in lingua Italiana.
- **Documenti esterni**: Documenti che verranno condivisi anche con il proponente e i committenti. Nel caso di documenti utili per il *deploy_[g]* del software o per il loro utilizzo da parte degli utenti finali, dovranno essere redatti in lingua inglese.

3.1.5 Nomenclatura dei documenti

Per quanto riguarda il nome dei documenti formali essi verranno nominati attraverso la dicitura "NomeDocumento", senza spazi e con lettere maiuscole all'inizio di ogni parola, seguita dalla versione, nella forma: "NomeDocumento_vX.Y.Z.estensione". Per quanto riguarda i verbali (interni ed esterni) si seguirà una particolare nomenclatura, descritta nella sezione 4.1.4.3.

3.1.5.1 Versioni di un documento

Il numero di versione è indicato tramite il carattere 'v' seguito da un numero, da un punto, da un numero, da un punto e un numero: vX.Y.Z. Ogni volta che si inserisce una modifica nell'omonimo diario, va assegnato un nuovo numero di versione, in modo coerente con le seguenti regole:

- Quando verrà creato un documento avrà necessariamente il numero di versione 'v0.0.1'.
- **X**: Rappresenta il numero di pubblicazioni ufficiali del documento (quindi ogni qualvolta il documento superi con esito positivo la fase di approvazione). Ad ogni nuova pubblicazione del documento i valori Y e Z vengono azzerati e quello di X incrementato di uno.
- **Y**: Rappresenta il numero di verifiche eseguite sul documento. L'incremento dell'indice Y implica l'azzeramento del valore dell'indice Z;
- **Z**: Rappresenta il numero di modifiche effettuate al documento durante il suo sviluppo.

3.1.5.2 Formato dei file

I documenti sono redatti attraverso lo strumento Latex, pertanto ogni documento si trova nel formato .tex durante il suo sviluppo. Dopo il superamento della fase di approvazione del documento da parte del Responsabile viene il documento viene esportato in formato PDF.

3.1.6 Documenti correnti

Di seguito si presentano i documenti formali, classificati per appartenenza (interno o esterno) consegnati:

- **Analisi dei Requisiti:** Uso esterno, sigla (AdR).
Documento per esporre e scomporre i requisiti del progetto contenente i *casi d'uso*_[g] relativi al prodotto e diagrammi di interazione con l'utente. Viene scritto dagli Analisti dopo aver analizzato il capitolato e interagendo con il Proponente in riunioni esterne.
- **Glossario:** Uso esterno, sigla (GL).
Documento per raccogliere le definizioni dei termini o concetti che saranno usati nei documenti formali per facilitarne la comprensione.
- **Piano di Progetto:** Uso esterno, sigla (PdP).
Documento per l'analisi e la pianificazione della gestione delle risorse di tempo e umane.
- **Piano di Qualifica:** Uso esterno, sigla (PdQ).
Documento per descrivere standard e obiettivi che il gruppo dovrà raggiungere per garantire la qualità di prodotto e processo.
- **Studio di Fattibilità:** Uso interno, sigla (SdF).
Documento per indicare le riflessioni, punti di forza e caratteristiche sfavorevoli per ogni capitolato proposto sulla base dei quali il gruppo ha fatto la sua scelta.
- **Norme di Progetto:** Uso interno, sigla (NdP).
Documento per mostrare le direttive e gli standard utilizzati all'interno del gruppo di lavoro Cyber13 per lo sviluppo del progetto.

3.1.7 Formattazione dei documenti

3.1.7.1 Strutturazione dei file

Ogni documento formale si deve attenere alla seguente strutturazione dei contenuti:

- **Frontespizio:** Si tratta della prima pagina dei documenti formali, conterrà le seguenti informazioni (in ordine dall'alto verso il basso):
 - Intestazione: "Università degli Studi di Padova";
 - Logo del gruppo: Centrato;
 - Titolo del documento: Centrato;
 - Nome del gruppo - Nome progetto: In corsivo.
- **Sezione di informazione del documento:** Si trova sempre nella prima pagina del documento e contiene le seguenti informazioni:
 - Versione: Attenendosi alle norme della sezione 3.1.4.2;

- Data Redazione: Seguendo il formato indicato al punto 3.1.6.2 "Formati ricorrenti";
 - Responsabile: Nome del responsabile che ha supervisionato il documento;
 - Redazione: Nomi dei redattori del documento;
 - Verifica: Nomi dei verificatori del documento;
 - Approvazione: Nome del responsabile che ha approvato il documento;
 - Uso: Interno o esterno;
 - Destinatari: A cui è indirizzato il documento;
 - Mail di contatto: Mail del gruppo Cyber13.
- **Diario delle modifiche:** Tabella inserita nella seconda pagina del documento. Ha lo scopo di riepilogare l'elenco delle modifiche che sono state apportate al documento nel corso del processo di redazione. L'assegnamento delle versioni si attiene alle norme indicate nella sezione 3.1.4.2 "Versioni del documento". Il diario è una tabella ordinata in modo decrescente secondo la data della modifica e, di conseguenza, il numero di versione. Si adotta questo stile per focalizzare gli ultimi cambiamenti poiché trattati nelle prime righe della tabella. Ogni riga del diario dovrà contenere i seguenti elementi nell'esatto ordine in cui sono indicati:
 - Versione del documento;
 - Data della versione;
 - Descrizione delle modifiche apportate;
 - Autore delle modifiche apportate;
 - Ruolo dell'autore delle modifiche;
 - **Indice delle sezioni:** Gli indici hanno lo scopo di riepilogare e dare una visione macroscopica della struttura del documento. Permettono quindi di rintracciare i contenuti tramite una gerarchia. Ogni documento, esclusi i verbali, dovrà essere corredato dall'indice dei contenuti. Esso sarà posizionato dopo il diario delle modifiche. Se sono presenti tabelle o immagini all'interno del documento, l'indice dei contenuti sarà seguito dalla lista delle tabelle e poi dalla lista delle figure.
 - **Elenco delle tabelle:** Vengono indicate tutte le tabelle inserite all'interno del documento.
 - **Elenco delle figure:** Vengono indicate tutte le figure inserite all'interno del documento.
 - **Introduzione:** Sezione sempre presente nei documenti che ne chiarifica:
 - Scopo del documento;
 - Scopo del prodotto;

- Riferimenti: normativi e informativi.
- **Contenuto del documento:** Le restanti pagine del documento sono interamente occupate dal contenuto dello stesso. In ogni pagina, esclusa il frontespizio, devono comparire i seguenti elementi:
 - Logo del team: Dovrà comparire, nell'intestazione della pagina, il logo del gruppo Cyber13. Tale logo sarà posizionato a sinistra.
 - Sezione corrente: Il numero e il nome della sezione in cui ci si trova dovranno essere presenti nell'intestazione. Tali elementi saranno posizionati a destra;
 - Nome del documento: A piè di pagina, a sinistra, deve apparire il nome del documento completo di versione.
 - Numero di pagina: A piè di pagina, a destra, deve comparire il numero della pagina espresso nella seguente notazione: "Pagina x di n", dove x è la pagina corrente e n il numero di pagine totali.

3.1.7.2 Norme tipografiche

- **Virgolette:** Alte singole ' ' per singolo carattere, alte doppie " " per racchiudere stringhe mentre parentesi angolari « » per racchiudere citazioni.
- **Parentesi:** Tonde per descrivere esempi e fornire sinonimi o precisazioni, quadre per rappresentare uno *standard*_[g] *ISO*_[g], uno stato relativo a un *ticket*_[g] o un riferimento ad un codice definito all'interno del documento stesso.
- **Punteggiatura:** Ogni segno di punteggiatura deve essere seguito da uno spazio e non avere spazi precedenti al segno stesso;
- **Stile del testo:**
 - Corsivo: Per dare enfasi ad una parola, un concetto o per indicare il nome di un termini di glossario e nomi di documenti.
 - Grassetto: Per i titoli, sottotitoli ed elementi di elenchi e liste di definizione di primo livello.
 - Azzurro: Per indicare dei collegamenti ipertestuali.
- **Elenchi:** La prima parola di ogni punto appartenete a un elenco inizierà con la lettera maiuscola. In caso di elenchi di definizioni, al termine da definire seguiranno i 'due punti' (:) successivamente vi sarà la sua descrizione. Al termine della descrizione dell'elemento si inserirà il carattere 'punto e virgola' (;) nel caso di definizione semplice, mentre il carattere 'punto' (.) al termine di una definizione complessa (più di un periodo). Per l'ultimo elemento della lista si userà sempre il carattere 'punto' (.);
- **Formati ricorrenti:**

- Date: Scritte con lo standard YYYY-MM-DD dove YYYY indica l'anno, MM il mese e DD il giorno;
- Orari: Scritti nel formato 24h.

- **Componenti grafiche:**

- Immagini: I formati ammessi sono PNG o JPG;
- Tabelle: Devono rispettare lo stile del *template_{|g|} Latex_{|g|}* realizzato.

3.1.8 Ambiente

La stesura dei documenti deve essere effettuata utilizzando il *linguaggio di markup_{|g|} LaTeX_{|g|}* e l'ambiente *Overleaf_{|g|}* con dizionario italiano ed inglese installati.

3.2 Qualità

3.2.1 Scopo

Ivi sono descritte le norme alle quali i componenti del gruppo devono sottostare in sede di redazione del *Piano di Qualifica v1.0.0*.

3.2.2 Aspettative

Il gruppo desidera redarre il documento *Piano di Qualifica v1.0.0* in modo coerente con quanto riportato nelle norme ivi indicate.

3.2.3 Classificazione dei processi

La qualità del lavoro è garantita dalla suddivisione di quest'ultimo in vari processi, riportati all'interno del Piano di Qualifica.

Ad occuparsi di tale suddivisione sono gli Amministratori, che dovranno rispettare la seguente notazione:

PROC[num]

Dove num è un numero positivo, intero di quattro cifre che identifica univocamente il determinato processo. La numerazione parte da "0001".

3.2.4 Classificazione delle metriche

La rilevazione della qualità del lavoro è effettuata dagli amministratori attraverso metriche riportate nel *Piano di Qualifica v1.0.0*. Onde evitare ambiguità le metriche dovranno seguire la seguente notazione:

$$M[\text{categ}][\text{sottocateg}][\text{num}]$$

All'interno della quale:

- **categ**: Indica la categoria della metrica riferendosi a prodotti, processi o test. Può assumere i seguenti valori:
 - PROC: Per indicare i processi;
 - PROD: Per indicare i prodotti;
- **sottocateg**: Indica la sotto-categoria della metrica, se esiste, in caso contrario non è presente.
 - Per quanto riguarda la categoria PROC, solo nel caso in cui si tratti di processi di test, possono assumere i seguenti valori:
 - * TA: per indicare tutti i tipi di test
 - * TM: per indicare i test di modulo
 - * TH: per indicare i test ad alto livello
 - Per quanto riguarda la categoria PROD, possono assumere i seguenti valori:
 - * D: per indicare i documenti;
 - * S: per indicare il software.
- **num**: Numero positivo, intero di quattro cifre che identifica univocamente la determinata metrica. La numerazione parte da "0001".

3.2.5 Procedure

L'implementazione di metriche tramite procedure verrà studiato con l'avanzare del progetto.

3.2.6 Metriche per la qualità di processo

Di seguito sono spiegate nel dettaglio le metriche utilizzate per valutare la qualità dei processi.

3.2.6.1 M[PROC][0001] - Schedule Variance

Indica a che livello di avanzamento del progetto rispetto alla pianificazione delle attività.

$$SV = BCWP - BCWS$$

dove BCWP indica le attività svolte e BCWS le attività che dovrebbero essere state svolte finora.

3.2.6.2 M[PROC][0002] - Budget Variance

Indica se attualmente si è speso meno o più di quanto previsto.

$$BV = BCWS - ACWP$$

dove BCWS indica il costo pianificato delle attività svolte ad una certa data e ACWP indica il costo effettivo delle attività svolte a tale data.

3.2.6.3 M[PROC][0003] - Rischi non individuati

Indice del numero di rischi non individuati nella fase di analisi: è indicato con un numero intero incrementato partendo da zero per ogni rischio rilevato che non fosse stato individuato precedentemente in fase di analisi dei rischi. Viene resettato all'inizio di ogni fase di progetto.

3.2.6.4 M[PROC][0004]: Numero campi dati per classe

Indice del numero di campi definiti in una classe. Un numero eccessivo di campi dati rischia di rendere la classe poco specializzata e indica una cattiva progettazione.

3.2.6.5 M[PROC][0005] - Metodi per classe

Indice del numero di metodi definiti in una classe.

3.2.6.6 M[PROC][0006] - Parametri per metodo

Indice del numero di parametri definiti in un metodo. Un numero eccessivo di parametri per metodo potrebbe sovraccaricare lo *stack*_{|g|} e comprometterne la funzionalità.

3.2.6.7 M[PROC][0007]: Grado di instabilità

È un modo per misurare l'instabilità delle componenti di un sistema, in particolare la possibilità di effettuare modifiche ad un elemento del sistema senza influenzarne altri all'interno dell'applicazione. Questo risultato dipende dall'indice afferente (numero di classi esterne dipendenti da quelle interne) ed efferente (numero di classi interne dipendenti da quelle esterne).

$$I = \left(\frac{C_e}{C_a + C_e} \right) * 100$$

dove Ce è indice efferente e Ca è indice afferente.

3.2.6.8 M[PROC][0008] - Complessità Ciclomantica

Indica la complessità di funzioni, moduli, metodi o classi di un programma contando il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso.

3.2.6.9 M[PROC][0009] - Linee di codice per linee di comando

Indica la percentuale di linee di commento presenti all'interno del codice sorgente.

$$P = \left(\frac{N_c}{N_{sloc}} \right) * 100$$

dove N_c è il numero di linee di commento e N_{sloc} è il numero di linee di codice prodotte.

3.2.6.10 M[PROC][0010] - Halstead Difficulty per-function

Misura il livello di complessità di una funzione.

$$DIF = \left(\frac{UOP}{2} \right) * \frac{OD}{UOD}$$

dove UOP è il numero di operatori distinti, OD è il numero totale di operandi e UOD è il numero di operandi distinti.

3.2.6.11 M[PROC][0011] - Halstead Volume per-function

Indica la dimensione dell'implementazione di un algoritmo basandosi sul numero di operazioni eseguite e sugli operandi di una funzione.

$$VOL = (OP+OD) * \log_2(UOP + UOD)$$

dove OP è il numero totale di operatori, OD è il numero totale di operandi, UOP è il numero di operatori distinti e UOD è il numero di operandi distinti.

3.2.6.12 M[PROC][0012] - Halstead Effort per-function

Rappresenta il costo necessario per scrivere il codice di una funzione.

$$E = DIF * VOL$$

dove DIF indica l'Halstead Difficulty e VOL è l'Halstead Volume.

3.2.6.13 M[PROC][0013] - Indice di manutenibilità

Permette di stabilire quanto sarà semplice mantenere il codice prodotto.

$$MI = 171 - 3.42 * \ln(\text{aveE}) - 0.23 * \ln(\text{aveV}) - 1616.2 * \ln(\text{aveLOC})$$

dove aveE è l'Halstead Effort medio per modulo, aveV è la complessità ciclomatica media per modulo, e aveLOC è il numero medio di linee di codice per modulo.

3.2.6.14 M[PROC][TM][0001] - Percentuale di codice coperto da test

Indica il rapporto tra linee di codice per le quali è previsto un test di verifica su linee di codice totale in percentuale.

$$PL_{cc} = \left(\frac{L_{cc}}{L_{ct}}\right) * 100$$

dove L_{cc} indica linee codice coperte e L_{ct} le linee totali di codice.

3.2.6.15 M[PROC][TM][0002] - Percentuale di test passati

Indica il rapporto tra i test passati e i test totali nella fase di sviluppo.

$$PT_p = \left(\frac{T_p}{T_t}\right) * 100$$

dove T_p indica i test passati e T_t i test totali.

3.2.6.16 M[PROC][TM][0003] - Percentuale di test non passati

Indica il rapporto tra i test non passati e i test totali nella fase di sviluppo

$$PT_{np} = \left(\frac{T_{np}}{T_t}\right) * 100$$

dove T_{np} indica i test non passati mentre T_t sono i test totali.

3.2.7 Metriche per la qualità di prodotto

Di seguito sono spiegate nel dettaglio le metriche utilizzate per valutare la qualità di prodotto.

3.2.7.1 Metriche per i documenti

3.2.7.1.1 M[PROD][D][0001]: Indice di *Gulpease*_{|g|}

È un indice di leggibilità per la lingua italiana, con il vantaggio di contare le singole lettere e non le sillabe per definire la lunghezza delle parole. L'indice prende in considerazione la lunghezza delle parole e la lunghezza delle frasi rispetto al numero totale delle lettere. Come descritto nella seguente formula, restituisce poi un valore che indica l'indice di difficoltà della leggibilità del testo.

$$89 + \frac{300 * (\text{numero delle frasi}) - 10 * (\text{numero delle lettere})}{\text{numero delle parole}}$$

Risultati:

- Minore di 80: difficile da leggere per chi ha licenza elementare;
- Minore di 60: difficile da leggere per chi ha licenza media;
- Minore di 40: difficile da leggere per chi ha un diploma superiore.

Ai fini di rendere il testo il più comprensibile possibile, si è deciso di porre importanza anche ad un'analisi diretta da parte di una persona, in quanto l'indice di leggibilità fornito dalla formula sopra riportata non garantisce un buon testo sotto tutti gli aspetti ritenuti importanti. Uno dei motivi che ha spinto il gruppo a questa scelta è dato dal fatto che un risultato considerato non buono secondo l'indice di Gulpease, potrebbe essere conseguenza di un testo con frasi complesse utilizzate per non ricadere in contenuti poco formali; inoltre il risultato non dà informazioni sulla logica del contenuto.

Per calcolare l'indice di Gulpease il gruppo ha tenuto conto solo dei contenuti testuali informativi, tralasciando contenuti come indice, intestazione e tabelle.

3.2.7.1.2 M[PROD][D][0002]: Errori ortografici

Overleaf_{|g|}, l'editor utilizzato per redigere la documentazione, fa uso di un correttore ortografico per mettere in evidenza gli errori di ortografia. Sarà compito del Verificatore correggerli attraverso l'uso di tale strumento.

3.2.7.2 Metriche per il Software

3.2.7.2.1 M[PROD][S][0001]: Copertura requisiti obbligatori

Una volta eseguita l'analisi dei requisiti, essi vengono organizzati in unità di dimensioni gestibili e quelli obbligatori messi in evidenza. Tale metrica indica la percentuale di requisiti obbligatori implementati fino a un dato momento. La formula che la misura è dunque:

$$CRo = \frac{NRoS}{NRoT} * 100$$

dove NRoS è il numero dei requisiti obbligatori soddisfatti e NRoT è il numero dei requisiti obbligatori totali

3.2.7.2.2 M[PROD][S][0002]: Copertura requisiti accettati

Una volta eseguita l'analisi dei requisiti, essi vengono organizzati in unità di dimensioni gestibili e una parte di essi viene identificata come accettati ma non obbligatori. Tale metrica indica la percentuale di implementazione di questi requisiti fino a un dato momento. La formula che la misura è dunque:

$$CRa = \frac{NRaS}{NRaT} * 100$$

dove NRaS è il numero dei requisiti accettati che sono stati soddisfatti e NRaT è il numero totale dei requisiti accettati.

3.2.7.2.3 M[PROD][S][0003]: Accuratezza rispetto alle attese

In fase di testing tale metrica misura la percentuale dei test che hanno restituito i risultati attesi. La formula è la seguente:

$$AC = (1 - \frac{NTf}{NTt}) * 100$$

dove NTf è il numero di test case falliti e NTt è il numero di test case totali.

3.2.7.2.4 M[PROD][S][0004]: Densità di *failure*_{|g|}

Rappresenta un dato opposto a quello dell'Accuratezza rispetto alle attese. Indica infatti la percentuale di test conclusi con un esito di failure, quindi negativo, rispetto al totale di test eseguiti.

$$DF = (\frac{Nfr}{Nte}) * 100$$

dove Nfr indica il numero di failure rilevati e Nte il numero di test case eseguiti.

3.2.7.2.5 M[PROD][S][0005]: Blocco di operazioni non corrette

È un valore espresso in percentuale che indica il livello di funzionalità con cui si gestiscono correttamente i fault.

$$BNC = (\frac{Nfe}{Non}) * 100$$

Nfe: numero di failure evitati durante i test; Non: numero di test case eseguiti che prevedono l'esecuzione di operazioni non corrette.

3.2.7.2.6 M[PROD][S][0006]: Tempo di risposta

Consiste nel tempo che intercorre tra la domanda al software per una precisa funzionalità e il risultato restituito all'utente.

$$TR = \frac{\sum_{k=1}^N Tk}{n}$$

dove a partire dalla richiesta di n funzionalità, Tk è il tempo trascorso tra una richiesta k per una determinata funzionalità e il termine delle operazioni necessarie a dare seguito a tale richiesta.

3.2.7.2.7 M[PROD][S][0007]: Comprensibilità delle funzioni offerte

Consiste nella percentuale di operazioni comprese nell'immediato senza il bisogno di consultare il manuale.

$$CFC = (\frac{Nfc}{Nfo}) * 100$$

dove Nfc è il numero di funzionalità comprese dall'utente e Nfo è il numero di funzionalità offerte ad esso offerte.

3.2.7.2.8 M[PROD][S][0008]: Facilità di apprendimento delle funzionalità

Indica il tempo, misurato in minuti, utilizzato dall'utente per apprendere nello specifico una determinata funzionalità e il modo di utilizzarla correttamente.

3.2.7.2.9 M[PROD][S][0009]: Utilizzo effettivo delle funzionalità

Percentuale che indica il livello di apprezzamento dell'utente rispetto alle funzionalità offerte dall'applicazione.

$$PA = \left(\frac{Nfu}{Nfo} \right) * 100$$

dove Nfu è il numero funzionalità utilizzate dall'utente e Nfo è il numero funzionalità offerte dall'app.

3.2.7.2.10 M[PROD][S][0010]: Capacità di analisi di failure

Rapporto, espresso in percentuale, delle failure di cui si conosce la causa rispetto al numero totale di failure attualmente rilevate.

$$CF = \left(\frac{Nfi}{Nft} \right) * 100$$

dove Nfi è il numero di failure di cui si è identificata la causa e Nft è il numero di failure totali trovate.

3.2.7.2.11 M[PROD][S][0011]: Impatto delle modifiche

Dato il numero di failure risolte, questo indice calcola, in percentuale, il rapporto tra quante di esse hanno generato nuove failure in seguito alla loro risoluzione e il loro numero totale.

$$IM = \left(\frac{Nfg}{Nfr} \right) * 100$$

dove Nfg è il numero di failure generanti (che generano altre failure) e Nfr è il totale di failure risolte.

3.2.7.2.12 M[PROD][S][0012]: Rapporto linee di commento su linee di codice

Si considera di grande importanza la presenza di commenti che descrivono il codice, in quanto aumentano la leggibilità e l'intelligibilità dello stesso e ne favoriscono la manutenzione, soprattutto in previsione di interventi di più persone.

$$RLC = \left(\frac{Nlcom}{Nlcod} \right) * 100$$

dove Nlcom è il numero di linee di commenti e Nlcod è il numero di linee di codice

3.2.7.2.13 M[PROD][S][0013]: Versioni di Android supportate

È considerato prioritario il supporto delle versioni più recenti del sistema operativo Android, tuttavia è considerato migliorativo un numero maggiore di versioni supportate dall'applicazione.

3.2.7.2.14 M[PROD][S][0014]: Copertura del framework *Octalysis*_{|g|}

Il prodotto include funzionalità che seguono i principi del framework di *gamification*_{|g|} Octalysis.

Per misurare la distribuzione di tali funzionalità vengono contate per ogni *Core Drive*_{|g|} quante appartengono ad ognuno di essi, e riportati i valori su una scala da 0 a 10. Nel *Piano di Qualifica v1.0.0* sono definiti i valori accettabili per ciascun Core Drive. Per la visualizzazione grafica della distribuzione delle funzionalità viene utilizzato il software online Octalysis Tool.

Per una breve panoramica sulla gamification e sul framework Octalysis fare riferimento al paragrafo 3.3.3 del *Piano di Qualifica v1.0.0*.

3.3 Configurazione**3.3.1 Scopo**

Nella seguente sezione sono descritte le direttive riguardanti la configurazione degli strumenti di condivisione attraverso i quali sviluppare il materiale da consegnare.

3.3.2 Aspettative

Il gruppo si attende alle seguenti norme al fine di condividere tra i vari membri del team il materiale da consegnare in modo ordinato e tracciabile.

3.3.3 Controllo di versione**3.3.3.1 Descrizione**

Si è scelto l'utilizzo della tecnologia Git per la raccolta e la condivisione dei file di documentazione e file non formali, opportunamente organizzati in *repository*_{|g|} separati.

3.3.3.2 Struttura delle repository

Durante la fase di RR son state create due cartelle, una contenente i file non formali chiamata "non formali" e una contenente la documentazione denominata "Documentazione" dove al suo interno è stata creata la cartella "RR" contenente i file di questa fase, organizzati secondo la seguente struttura:

- Documenti interni

- Studio Di Fattibilità;
- Norme Di Progetto;
- Verbali interni.

- **Documenti esterni**

- Piano di Progetto;
- Piano di Qualifica;
- Analisi dei Requisiti
- Glossario;
- Verbali esterni.

All'interno delle cartelle son presenti i file finali in formato PDF mentre i file *LaTex*_{|g|} divisi in sezioni con relativo diario modifiche restano salvati su un account condiviso da tutti i membri del gruppo sulla piattaforma *Overleaf*_{|g|}.

3.3.3.3 Branch

Per il momento si è deciso di non fare uso dei *branch*_{|g|} in quanto per lo sviluppo della documentazione da presentare in fase RR si è stabilito di contenere tutto nel master branch. Successivamente verrà creato un branch per ogni membro del gruppo per la gestione del codice in parallelo in uno stesso documento.

3.3.3.4 Aggiornamento della repository

- Dare il comando "git pull" per aggiornare il repository locale sulla base delle modifiche presente in remoto.
- Nel caso in cui si verifichino dei conflitti:
 - Dare il comando "git stash" per accantonare momentaneamente le modifiche apportate.
 - Dare il comando "git pull" per ottenere ed applicare i *commit*_{|g|} mancanti.
 - Dare il comando "git stash apply" per ripristinare le modifiche.
- Dare il comando "git add [files]" , che aggiungerà i file modificati e quelli nuovi specificati;
- Dare il comando "git commit" e successivamente riassumere le modifiche effettuate, in caso sia utile si può aggiungere un messaggio esteso di descrizione (aggiungendo al comando la dicitura -m "messaggio").
- Dare il comando "git push" per completare l'operazione e fornire le modifiche agli altri membri del gruppo.

3.3.4 Strumenti

- **Client git:** Secondo le preferenze dei membri del gruppo vengono usati principalmente due client *GitKraken*_[g] e *Git Bash*_[g].
- **Server git:** Vista la conoscenza preliminare e l'affidabilità di *Github*_[g] si è scelto questo server.

3.4 Verifica

3.4.1 Scopo

Attraverso il processo di verifica si vuole evidenziare ed eliminare la presenza di errori nell'esecuzione degli altri processi per l'intero sviluppo del prodotto, in particolare alla conclusione di ogni incremento. Questa sezione norma gli strumenti e i metodi che verranno usati per la verifica del codice e dei documenti. Per quanto riguarda questa fase antecedente alla RR la verifica verterà essenzialmente su documenti e diagrammi.

3.4.2 Aspettative

Un'implementazione del processo di verifica permette di ottenere i seguenti effetti:

- Ottenere un sistema di controllo degli errori non invasivo, efficiente ed efficace;
- Stabilire i criteri necessari per la verifica del prodotto;
- Sottolineare le problematiche trovate col fine di risolverle.

3.4.3 Descrizione

Il processo si divide nei seguenti momenti:

- **Controllo:** Attraverso le tecniche di analisi statica e analisi dinamica (descritte nelle sezioni successive) si riesce ad effettuare un controllo approfondito del codice sorgente e della sua corretta esecuzione;
- **Test:** Esecuzione test sul software.

3.4.4 Attività

3.4.4.1 Analisi statica

L'analisi statica è una tecnica di analisi applicabile sia alla documentazione che al codice e permette di effettuare la verifica di quanto prodotto individuando errori ed anomalie.

Vi sono due metodologie per effettuarla:

- **Walkthrough:** tecnica applicata quando non si sanno le tipologie di errori o problemi che si stanno cercando e quindi prevede una lettura da cima a fondo del codice o documento per trovare anomalie di qualsiasi tipo. Questo metodo risulta essere il più dispendioso in termini di efficienza ma si tratta anche del più semplice da imparare, diventando inevitabilmente il più utilizzato nelle prime fasi del progetto.
- **Inspection:** tecnica da applicare quando si ha idea delle possibili problematiche che si stanno cercando e si attua leggendo in modo mirato il documento o il codice. Questo procedimento risulta molto più veloce del precedente in quanto, utilizzando la lista di controllo degli errori e dalle misurazioni effettuate, permette un'analisi più efficace dei punti critici, tralasciando invece le parti senza problematiche.

3.4.4.2 Analisi dinamica

Il processo di analisi dinamica consiste nella realizzazione ed esecuzione di una serie di test di vario tipo sul codice del software. Questa tecnica non è applicabile per trovare errori nella documentazione.

3.4.4.2.1 Test

Poiché la fase di RR prevede la sola preparazione di documentazione, le norme che regolano l'analisi dinamica verranno definite in seguito.

3.4.5 Strumenti

- **Documenti:** Per quanto riguarda la verifica ortografica ci si affida allo strumento integrato di Overleaf, attraverso il quale le parole sbagliate vengono segnate in rosso, permettendo un controllo rapido ed efficace.
- **Indice Gulpease:** Per il calcolo dell'Indice di Gulpease il team ha utilizzato lo strumento disponibile online chiamato *Farfalla Project*_[g];
- **Octalysis Tool:** Per una valutazione della qualità degli aspetti relativi alla *Gamification*_[g] si è scelto di utilizzare lo strumento online Octalysis Tool disponibile sul sito del creatore del framework *Octalysis*_[g] Yu-kai Chou.
- **Gestione processi e feedback:** Anche per la verifica si è stato scelto di utilizzare il sistema di *issues*_[g] integrato in GitHub.

3.5 Validazione

3.5.1 Scopo

Il processo di validazione ha l'obiettivo di verificare che il prodotto sia conforme a quanto pianificato e sia abile nel gestire e minimizzare gli effetti degli errori.

3.5.2 Aspettative

In questa sezione il gruppo intende sviluppare direttive utili per eseguire correttamente il processo di validazione.

3.5.3 Procedure

I passi per compiere l'attività di validazione sono i seguenti:

- **Tracciamento:** Tracciamento dei test con rendiconto eseguito dai verificatori;
- **Analisi risultati:** In seguito a seguire il Responsabile di progetto controlla e analizza i report ricevuti sui test decidendo se:
 - Accettare la validazione e consegnare i risultati al proponente, informandolo sulle modalità di esecuzione della validazione;
 - Chiedere la ripetizione di alcuni o tutti i test con ulteriori nuove indicazioni.

4 Processi organizzativi

4.1 Processi di coordinamento

4.1.1 Scopo

In questa sezione vengono elencate direttive, linee guida e strumenti che il gruppo dovrà utilizzare a fini organizzativi all'interno dello svolgimento del lavoro.

4.1.2 Aspettative

Il gruppo si attende alle seguenti direttive al fine di organizzare, evitando incomprensioni di varia natura, gli incontri tra i componenti.

4.1.3 Comunicazione

In questa sezione vengono illustrati i metodi di comunicazione sia tra i membri all'interno del gruppo Cyber13 sia tra il gruppo e le entità esterne, come Committenti e Proponenti.

4.1.3.1 Comunicazioni interne

Le comunicazioni interne al gruppo avvengono tramite *Slack_[g]*, strumento di messaggistica nel quale sono stati predisposti dei canali tematici, suddivisi per argomento:

- **general:** Canale di carattere generale, per coordinare luoghi e orari di ritrovo, scambiare opinioni sul progetto e in generale inviare comunicati che non trovano spazio in alcuno degli altri canali specifici.
- **documentirr:** Canale predisposto per la coordinazione e l'aggiornamento dei lavori del team su tutti i documenti da approntare in vista della RR di Aprile 2019 (si procederà allo stesso modo anche per le fasi successive).

4.1.3.2 Comunicazioni esterne

Alcune norme regolano anche le comunicazioni con le parti esterne al gruppo Cyber13, nello specifico:

- La proponente GaiaGo rappresentata da Filippo De Pretto, con la collaborazione del quale si definiranno i *requisiti_[g]* che porteranno alla realizzazione del prodotto;
- I committenti Prof. Tullio Vardanega e Prof. Riccardo Cardin, ai quali verrà fornita la documentazione richiesta in ciascuna revisione di progetto.

Comunicazioni esterne scritte

Le comunicazioni esterne scritte vengono effettuate utilizzando l'indirizzo e-mail del gruppo swe.cyber13@gmail.com a cui tutti i membri hanno accesso. Per comunicare con il Proponente viene utilizzato l'indirizzo e-mail f.depreto@gaiago.com. Anche le comunicazioni con i committenti avvengono esclusivamente tramite l'indirizzo e-mail del gruppo: ogni messaggio deve essere definito da un oggetto chiaro e conciso, così come deve essere il suo contenuto, e rivolgersi ai committenti in modo formale e cortese.

4.1.4 Riunioni

Le riunioni interne ed esterne al team Cyber13 sono un momento fondamentale per il corretto avanzamento e conclusione del progetto. Alcune regole ne normano quindi l'organizzazione e lo svolgimento.

Per ogni riunione viene nominato un Segretario tra i membri del gruppo, il quale garantirà il rispetto dell'ordine del giorno, si occuperà di annotare i punti più importanti discussi e da tali appunti redigerà infine un Verbale di Riunione. Quest'ultimo non è un ruolo fisso, quindi il membro del team che ne ricoprirà la carica potrà variare da riunione a riunione se necessario.

Le riunioni che non potranno avere un luogo fisico di ritrovo avverranno in remoto attraverso il programma *Skype*_[g] o lo strumento *Google Meet*_[g], secondo quanto concordato preventivamente prima dell'appuntamento.

4.1.4.1 Riunioni interne

La partecipazione è ammessa ai soli membri del team Cyber13.

Ordine del giorno, data, ora e luogo della riunione vengono decisi dal Responsabile di progetto, che si occupa anche di approvare il Verbale di Riunione stilato dal Segretario.

I membri del team hanno il dovere di presentarsi in orario all'appuntamento, comunicando eventuali ritardi e partecipando attivamente e costruttivamente alla discussione.

La riunione è considerata valida se almeno 5 dei 6 membri del team sono presenti. In circostanze eccezionali qualora un membro fosse impossibilitato ad essere presente di persona ma la sua presenza dovesse essere fondamentale per la discussione dell'ordine del giorno, esso potrà prendere parte alla riunione da remoto attraverso uno strumento di videochiamata come Skype o Google Meet.

4.1.4.2 Riunioni esterne

Prendono parte a queste riunioni i membri del team Cyber13 e la Proponente.

Essendo la sede di GaiaGo a Milano, le riunioni esterne avranno il più delle volte luogo da remoto, attraverso uno strumento quale Skype o Google Meet, deciso di volta in volta prima dell'appuntamento.

Le riunioni esterne cui la Proponente prenderà parte di persona saranno invece effettuate presso la Torre Tullio Levi Civita (ex Torre Archimede), previa disponibilità

dei locali.

Come per le riunioni interne, il Verbale di Riunione sarà redatto da un Segretario eletto di volta in volta e sarà approvato in seguito dal Responsabile di Progetto. Data l'elevata importanza delle riunioni esterne, esse vengono considerate valide anche alla presenza di un singolo membro del team Cyber13 e della Proponente.

4.1.4.3 Verbale di riunione

Il Segretario eletto di volta in volta si occupa di stilarlo secondo il seguente schema:

- **Verbale TIPO del DATA:** Il frontespizio del Verbale di Riunione permette di identificarne il TIPO (Interno/Esterno) e la DATA in cui la riunione ha avuto luogo.
- **Diario delle modifiche:** Come definito nella sezione 3.1.6.1.
- **Indice:** Semplice indice dei contenuti del Verbale.
- **Informazioni sulla riunione:** Deve contenere obbligatoriamente le seguenti informazioni:
 - **Luogo e data della riunione:** ad esempio Padova, 15 Marzo 2019;
 - **Partecipanti alla riunione:** a partire dai Proponenti/Committenti in caso di Riunione Esterna, seguiti dai nomi dei membri del team Cyber13 presenti elencati in ordine alfabetico.

Possono invece considerarsi informazioni facoltative ma utili:

- **Ora di inizio e di fine:** scritte nel formato 24h;
- **Motivo della riunione:** Illustra in forma riassuntiva i motivi generali della riunione;
- **Ordine del giorno:** Indica attraverso un elenco numerato gli argomenti da discutere così come definiti dal Responsabile di Progetto;
- **Resoconto:** riassunto di quanto discusso in riunione, redatto dal Segretario. Indicherà chiaramente quali punti dell'Ordine del Giorno sono stati effettivamente affrontati, eventuali argomenti discussi che non fossero presenti nell'Ordine del Giorno e le relative decisioni prese.

Il tracciamento delle decisioni prese avverrà attraverso un sistema a codici del tipo VER-DATA.X dove VER significa Verbale, DATA è sostituito con la data della riunione e X è un numero sequenziale che identificherà univocamente la decisione cui si sta facendo riferimento.

Nomenclatura e archiviazione dei verbali: I Verbali di Riunione saranno archiviati con la nomenclatura VER-DATA dove DATA è la data della riunione nel formato YYYY-MM-DD per permetterne un posizionamento semplice e cronologico all'interno della cartella assegnata.

Trattandosi di documenti esterni e ufficiali, verranno archiviati nella cartella di Repository dedicata e saranno redatti in LaTeX.

4.2 Processi di pianificazione

4.2.1 Scopo

In questa sezione vengono elencate le norme relative ai processi di pianificazione interni al gruppo.

4.2.2 Aspettative

Il gruppo si aspetta di organizzare in maniera ottimale ed evitando incomprensioni di varia natura la divisione del lavoro da svolgere. Per fare vengono stabiliti ruoli precisi all'interno del progetto.

4.2.3 Ruoli di progetto

Nel corso del progetto ogni componente del team avrà modo di ricoprire a rotazione il ruolo di ognuna delle figure aziendali tipicamente coinvolte nella realizzazione di un software.

Tale assegnazione cercherà di distribuire nel modo più omogeneo possibile il tempo che ciascun membro investirà in ogni ruolo, dando in ogni caso priorità alla continuità delle attività già in corso.

Il Responsabile di Progetto di turno dovrà inoltre garantire che non si creino situazioni di conflitto di interesse che potrebbero compromettere la qualità del prodotto finale.

4.2.3.1 Responsabile di progetto

Il "Responsabile di progetto", o "Project Manager", accentra le responsabilità di scelta e approvazione del gruppo, del quale è anche il rappresentante presso proponenti e committenti.

Si occupa quindi di gestire le risorse umane e di coordinare i membri del team, dell'approvazione dei documenti, della pianificazione e delle relazioni esterne.

4.2.3.2 Amministratore

L'amministratore è una figura di supporto che si occupa di fornire al team gli strumenti per lavorare in maniera rigorosa e regolamentata.

Tali strumenti consistono nella redazione e attuazione di piani e procedure di Gestione della Qualità, la redazione delle Norme di Progetto, il versionamento e la configurazione dei prodotti.

Pur non essendo sotto sua diretta responsabilità, collabora alla redazione del Piano di Progetto e si assicura che la documentazione sia corretta e verificata.

4.2.3.3 Analista

L'analista si occupa di analizzare e capire appieno il dominio del problema per identificarne i requisiti espliciti ed impliciti, e per riconoscere ed evitare gravi problemi di progettazione successivi.

Essendo colui che all'interno del team meglio conosce il problema e i requisiti che la sua realizzazione richiede, esso è coinvolto anche nella definizione degli accordi contrattuali, nella verifica delle implicazioni di costo e qualità, nella realizzazione dello Studio di Fattibilità e dell'Analisi dei Requisiti.

4.2.3.4 Progettista

Il progettista si occupa della definizione di una soluzione soddisfacente per tutti gli *stakeholder*_[g], perseguendo efficienza ed efficacia nella soddisfazione dei requisiti a partire dal lavoro dell'Analista.

In tal senso il progettista conosce e applica soluzioni di lavoro note e ottimizzate (*Best practice*_[g]), e si occupa di organizzare e suddividere il sistema in parti di complessità trattabile, per rendere il lavoro di codifica realizzabile e verificabile.

4.2.3.5 Programmatore

Il programmatore si occupa della codifica del progetto, realizzando quindi il prodotto finale attraverso l'implementazione dell'architettura definita dal Progettista.

Il codice prodotto deve essere documentato, versionato e mantenibile secondo le norme fissate.

Si occupa anche della scrittura del Manuale Utente e di realizzare le componenti necessarie alla verifica e alla validazione del codice.

4.2.3.6 Verificatore

Il verificatore è coinvolto in ogni fase del progetto. Il suo compito è appunto quello di verificare la conformità dei prodotti ai requisiti fissati di funzionalità e qualità.

Il suo coinvolgimento è richiesto in ogni fase poiché deve accertarsi che ad ogni attività di processo eseguita non siano stati introdotti errori.

Attraverso la redazione di un Piano di Qualifica si occupa di illustrare le verifiche effettuate sul progetto e i relativi esiti, secondo quanto previsto dal Piano di Progetto.

4.2.3.7 Rotazione dei ruoli

Per assicurare un buon esito del progetto e allo stesso tempo assicurare una rotazione dei ruoli assegnati a ciascun membro del gruppo durante la sua realizzazione, sono necessarie alcune semplici regole:

- Considerare impegni e interessi di ogni singolo membro del gruppo;
- Evitare conflitti di interesse, in particolare evitare che un ruolo assegnato ad un membro del gruppo comporti la verifica di lavoro svolto in precedenza dallo stesso mentre ricopriva un altro ruolo all'interno della stessa fase dello sviluppo.

- Garantire che al cambio di ruolo il nuovo assegnatario venga posto nelle migliori condizioni di lavoro possibile. In tal senso il precedente assegnatario deve assicurarsi di non lasciare al nuovo criticità da lui non gestibili. Inoltre si occuperà di lasciargli al nuovo una lista di consigli e procedure utili in un documento interno informale.
- Garantire che ciascun membro ricoprirà di volta in volta il ruolo a lui assegnato in modo esclusivo: ciascuno dovrà occuparsi di quanto è di sua competenza, senza interferire o incrociare incarichi e responsabilità.

4.2.4 Ticketing

4.2.4.1 Task list

Lo svolgimento del progetto prevede la suddivisione del modello di sviluppo in varie fasi. Le attività da svolgere in una fase sono contenute in una *task board*_[g]. Il responsabile ha il compito di creare le task board per ogni fase del modello su *GitHub*_[g], utilizzando il nome "X", dove X identifica in modo significativo la sezione di progetto cui si riferisce. Per esempio, per quanto riguarda i documenti, X segue la nomenclatura indicata nella sezione 3.1.4.1.

4.2.4.2 Task

Ogni singola *task*_[g] viene creata o approvata dal Responsabile di progetto ed è caratterizzata da un titolo significativo per la componente di progetto cui si riferisce.

4.2.4.3 Ticket

Il *ticket*_[g] sono lo strumento attraverso cui viene gestito il ciclo di vita del task, dalla sua definizione, alla sua assegnazione ad uno o più membri del gruppo, alla sua implementazione, verifica, approvazione e chiusura. L'assegnazione del *ticket*_[g] ad uno specifico membro del gruppo può avvenire secondo due modalità:

- **Direttamente:** L'assegnazione del task è già noto al momento della creazione dello stesso. In questo caso il *ticket*_[g] viene assegnato direttamente allo specifico componente del gruppo da parte del Responsabile di progetto.
- **Indirettamente:** Nel caso in cui un membro non è presente agli incontri oppure il lavoro che richiede una *task*_[g] è impegnativo, può non essere possibile assegnare direttamente un ticket ad un componente del gruppo. In questo caso il membro non presente è obbligato a controllare la sezione *issues*_[g] presente su *GitHub*_[g] dove troverà il lavoro da svolgere. Questi tasks inoltre possono essere assegnati qualora il componente finisse i *ticket*_[g] a suo carico.

4.3 Procedure

4.3.1 Creazione e gestione dei task

Per la creazione di un task si procede nel seguente modo:

- Accesso alla task list relativa al progetto di interesse identificata con il nome dello stesso.
- Creazione di un nuovo task.
- Il Responsabile di Progetto può decidere se suddividerlo in sotto-task da assegnare a membri del gruppo differenti.

4.3.2 Gestione dei ticket

La vita di ogni ticket viene presentata nel seguente diagramma:

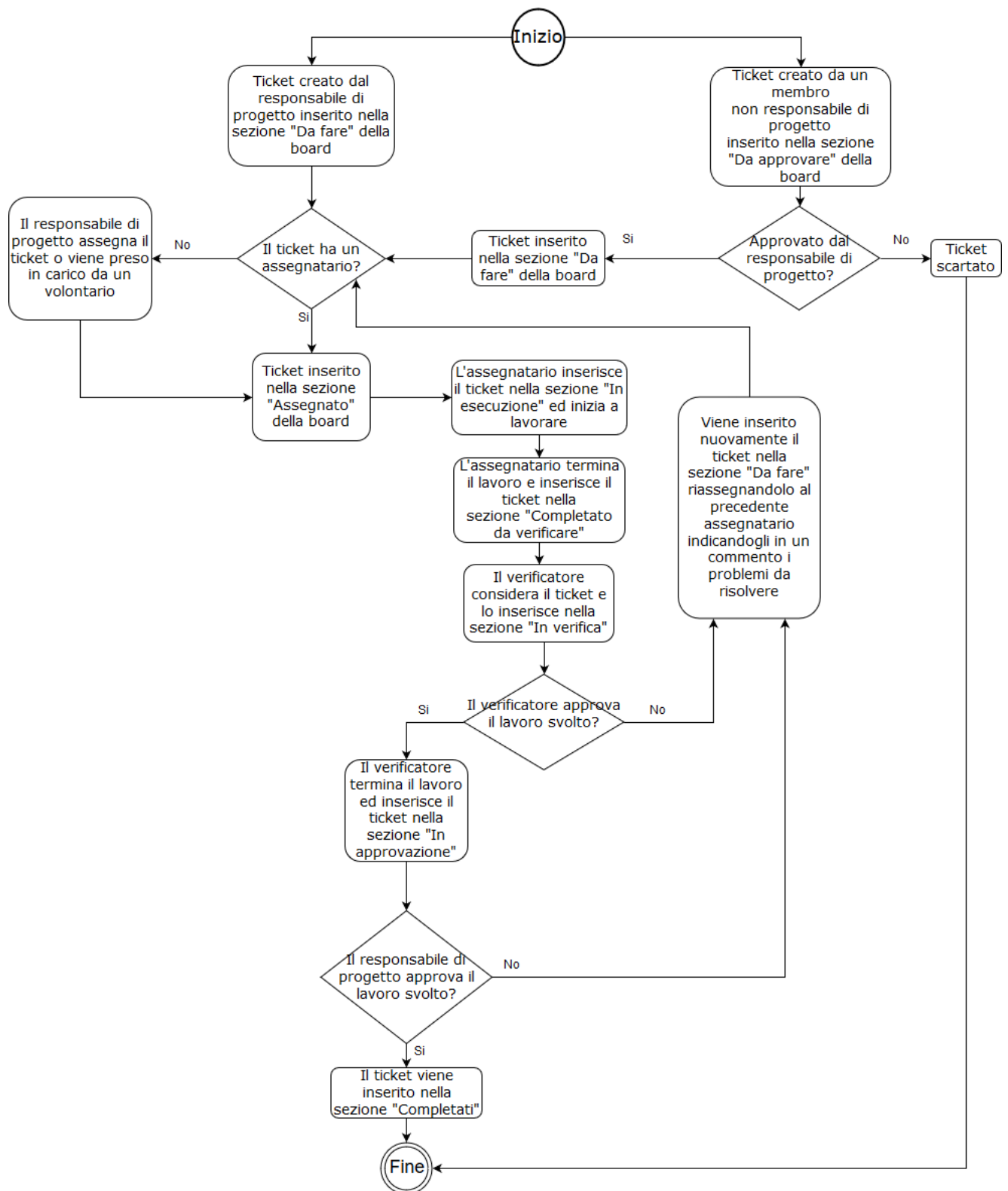


Figura 2: Diagramma del ciclo di vita di un ticket

4.3.3 Stesura del consuntivo

L'operazione di stesura del consuntivo viene eseguita dal Responsabile di Progetto nel seguente modo:

- Esporta da *Gantt project*_[g] un file in formato *Microsoft Excel*_[g] che indica le ore rendicontate nella fase corrente;
- Accede al *Foglio di Google*_[g] presente nel *Google Drive*_[g] associata alla mail del gruppo con relativo template e ne fa una duplicazione;
- Inserisce le ore rendicontate nelle apposite celle ottenendo la differenza con quelle preventivate;
- Riporta i valori ottenuti nel Piano di Progetto;
- Crea una tabella che, elaborando i dati derivanti dalla differenza tra le ore rendicontate e quelle preventivate, trova il budget effettivo rispetto a quello previsto;
- Infine con l'elaborazione di questi dati crea una valutazione complessiva del lavoro.

Per il periodo che precede la revisione RR ci si ferma al punto 3 senza ricavarne la differenza in quanto le ore preventivate non sono presenti. Successivamente ad ogni revisione sarà aggiornato il Piano di Progetto con i dati ottenuti.

4.4 Strumenti

4.4.1 Pianificazione

Per la pianificazione la scelta è ricaduta sulla piattaforma GitHub perchè offre la possibilità di organizzare le task di progetto in ticket, da assegnare a uno o più membri ed infine organizzarli in colonne. A tal proposito il gruppo ha deciso di creare varie colonne una per ogni fase del ciclo di vita del ticket in modo da avere un'idea chiara sul punto in cui sono i vari membri del gruppo con i relativi compiti.

4.4.2 Comunicazione

Per la comunicazione ci si è orientati per l'applicazione di messaggistica *Slack*_[g] appositamente per gruppi di lavoro, che offre la possibilità di realizzare canali telematici, in modo da creare comunicazioni specifiche per argomento. Per le comunicazioni interne al gruppo finora non si ha avuto la necessità di utilizzare strumenti per videochiamate in quanto ci si è organizzati in modo efficiente sugli incontri settimanali. In caso di necessità futura ne si prevede di utilizzare *Skype*_[g] e *Google Meet*_[g] che permettono videochiamate di gruppo.

4.4.3 Creazione diagrammi di Gantt

Per la realizzazione di diagrammi di *Gantt*_[g] si è scelto lo strumento *open source*_[g] e *multiplatforma*_[g] GanttProject.

4.4.4 Calcolo del consuntivo

Gli strumenti a disposizione dal Responsabile di Progetto sono i seguenti:

- Fogli Google
- GanttProject

4.5 Formazione

La formazione del personale è da svolgersi autonomamente. I membri del team Cyber13 sono tenuti a studiare individualmente le tecnologie che verranno utilizzate durante lo svolgimento del progetto.

La documentazione di riferimento, oltre al materiale precedentemente citato nella sezione 1.4.2 Riferimenti informativi, comprende:

- Documentazione LaTeX:
<https://www.latex-project.org>
- Documentazione GitHub:
<https://github.com>
- Documentazione UML:
https://www.math.unipd.it/~tullio/IS-2/Guida_Notazioni_UML-14.pdf