



AlphaSix

AlphaSix

Norme di Progetto

Informazioni sul documento

Nome Documento	NormeDiProgetto v1.0.0.pdf
Versione	1.0.0
Data di Creazione	26 novembre 2018
Data ultima modifica	10 gennaio 2018
Stato	Approvato
Redazione	Samuele Gardin
Verifica	Laura Cameran Matteo Marchiori
Approvazione	Nicola Carlesso
Uso	Interno
Distribuzione	AlphaSix
Destinato a	Prof. Tullio Vardanega, Prof. Riccardo Cardin, AlphaSix
Email di riferimento	alpha.six.unipd@gmail.com

Descrizione

Questo documento descrive le regole, gli strumenti e le convenzioni adottate da AlphaSix durante la realizzazione del progetto Butterfly.

Registro delle modifiche

Versione	Descrizione	Ruolo	Nominativo	Data
1.0.0	Approvazione	Responsabile	Nicola Carlesso	10-01-2019
0.4.0	Verifica finale	Verificatore	Matteo Marchiori	08-01-2019
0.3.4	Aggiunti link alla documentazione nella sezione Formazione, completato paragrafo	Verificatore	Laura Cameran	06-01-2018
0.3.3	Revisionati processi primari	Verificatore	Laura Cameran	06-01-2018
0.3.2	Completamento della sezione Gestione di progetto	Amministratore	Samuele Gardin	05-01-2018
0.3.1	Aggiornamento gestione di progetto	Verificatore	Laura Cameran	05-01-2018
0.3.0	Verifica documento	Verificatore	Ciprian Voinea	29-12-2018
0.2.6	Agginto processo di supporto "Verifica"	Verificatore	Nicola Carlesso	10-12-2018
0.2.5	Inserimento analisi statica e dinamica	Verificatore	Nicola Carlesso	08-12-2018
0.2.4	Aggiunta sottosezione "Manutenimento"	Amministratore	Timoty Granziero	06-12-2018
0.2.3	Completato paragrafo sui casi d'uso	Analista	Laura Cameran	05-12-2018
0.2.2	Redatto paragrafo "Processi di Sviluppo"	Amministratore	Timoty Granziero	04-12-2018
0.2.1	Completata sezione sui diagrammi UML	Analista	Laura Cameran	04-12-2018
0.2.0	Verifica dei contenuti presenti nel documento	Amministratore	Samuele Gardin	04-12-2018
0.1.6	Aggiunte sottosezioni template e ciclo di vita	Amministratore	Timoty Granziero	03-12-2018
0.1.5	Aggiunto "Struttura del documento"	Amministratore	Timoty Granziero	02-12-2018
0.1.4	Aggiunti riferimenti normativi e informativi	Amministratore	Laura Cameran	01-12-2018
0.1.3	Inserimento metriche	Verificatore	Nicola Carlesso	01-12-2018
0.1.2	Aggiunti "Strumenti di pianificazione e coordinamento"	Verificatore	Timoty Granziero	30-11-2018
0.1.1	Aggiunta pianificazione della qualità	Verificatore	Nicola Carlesso	30-11-2018
0.1.0	Verifica documento	Verificatore	Timoty Granziero	29-11-2018
0.0.8	Completata sezione sui processi di fornitura	Amministratore	Laura Cameran	28-11-2018



Versione	Descrizione	Ruolo	Nominativo	Data
0.0.7	Aggiunti paragrafi “Strumenti di supporto” e “Tabelle”	Verificatore	Timoty Granziero	27-11-2018
0.0.6	Migliorata sezione sull’ambiente di sviluppo	Amministratore	Laura Cameran	27-11-2018
0.0.5	Completata sezione design contenente stile del testo, elenchi puntati, figure, ecc	Amministratore	Laura Cameran	26-11-2018
0.0.4	Completata sezione sulla suddivisione dei documenti	Amministratore	Laura Cameran	26-11-2018
0.0.3	Completata introduzione	Amministratore	Laura Cameran	25-11-2018
0.0.2	Aggiunto contenuto alla sezione di formazione	Amministratore	Laura Cameran	23-11-2018
0.0.1	Creazione template	Redattore	Timoty Granziero	22-11-2018

Indice

1	Introduzione	2
1.1	Premessa	2
1.2	Scopo del documento	2
1.3	Scopo del prodotto	2
1.4	Glossario e documenti esterni	2
1.5	Riferimenti	2
1.5.1	Normativi	2
1.5.2	Informativi	3
2	Processi Primari	4
2.1	Processo di fornitura	4
2.1.1	Scopo	4
2.1.2	Ricerca delle tecnologie	4
2.1.3	Studio di Fattibilità	4
2.1.4	Preparazione in vista della revisione	4
2.2	Processo di sviluppo	4
2.2.1	Scopo	4
2.2.2	Analisi dei Requisiti	5
2.2.2.1	Denominazione dei requisiti	5
2.2.2.2	Casi d'uso	5
2.2.3	Diagrammi UML	6
2.2.3.1	Diagrammi dei casi d'uso	6
2.2.4	Strumenti di base	6
2.2.4.1	Ambiente di sviluppo	6
2.2.4.1.1	Sistema operativo	6
3	Processi Organizzativi	7
3.1	Gestione del progetto	7
3.1.1	Scopo	7
3.1.2	Pianificazione di attività	7
3.1.2.1	Scopo	7
3.1.2.2	Obiettivo	7
3.1.2.3	Ordine di esecuzione	8
3.1.2.4	Classificazione dei rischi	8
3.1.2.5	Ruoli di progetto	9
3.1.3	Pianificazione della qualità	9
3.1.3.1	Classificazione dei processi	10
3.1.3.2	Classificazione degli obiettivi	10
3.1.3.3	Classificazione delle metriche	10
3.1.4	Controllo del progetto	11
3.1.4.1	Monitoraggio del progetto	11
3.1.4.1.1	Monitoraggio dell'esecuzione dei processi	11
3.1.4.1.2	Procedure di comunicazione	11
3.1.4.2	Gestione dei problemi emersi	12
3.1.5	Chiusura dei processi	12
3.1.5.1	Archiviazione dei prodotti	12
3.1.5.2	Archiviazione delle misurazioni	12
3.1.6	Strumenti di pianificazione e coordinamento	12
3.1.6.1	Slack	12

3.1.6.2	GitHub	13
3.2	Formazione	13
3.2.1	Piano di formazione	13
4	Processi di supporto	15
4.1	Documentazione	15
4.1.1	Implementazione	15
4.1.1.1	Template	15
4.1.1.2	Ciclo di vita dei documenti	15
4.1.2	Struttura	15
4.1.2.1	Frontespizio	15
4.1.2.2	Storico delle versioni	16
4.1.2.3	Indice	16
4.1.2.4	Contenuto	16
4.1.3	Design	16
4.1.3.1	Norme tipografiche	16
4.1.3.1.1	Stile del testo	17
4.1.3.1.2	Elenchi puntati	17
4.1.3.1.3	Altri formati testuali comuni	17
4.1.3.2	Elementi grafici	17
4.1.3.2.1	Figure	17
4.1.3.2.2	Tabelle	17
4.1.4	Produzione	18
4.1.4.1	Suddivisione dei documenti	18
4.1.4.1.1	Documenti interni	18
4.1.4.1.2	Documenti esterni	18
4.1.4.1.3	Verbali	18
4.1.4.2	Strumenti di supporto	18
4.1.4.2.1	L ^A T _E X	18
4.1.4.2.2	TexStudio/Visual Studio Code	18
4.1.4.2.3	GanttProject	19
4.1.4.2.4	Draw.io	19
4.1.4.2.5	Indice di Gulpease	19
4.1.5	Mantenimento	19
4.1.5.1	Versionamento	19
4.1.5.2	Continuous Integration	19
4.1.5.3	Nomenclatura	20
4.1.5.3.1	Verbali	20
4.1.5.3.2	Documenti vari	20
4.2	Verifica	20
4.2.1	Scopo	20
4.2.2	Descrizione	20
4.3	Walkthrough e Inspection	20
4.3.1	Walkthtough	21
4.3.2	Inspection	21
4.3.3	Metriche	21
4.3.3.1	Metriche per i documenti	21
4.3.3.1.1	MPD001 Indice di Gulpease	21
4.3.3.1.2	MPD002 Correttezza ortografica	22
4.3.3.2	Metriche per i processi	22
4.3.3.2.1	MPR001 Varianza della pianificazione	22
4.3.3.2.2	MPR002 Varianza dei costi	22



4.3.3.2.3	MPR003 Aderenza agli standard	22
4.3.3.2.4	MPR004 Frequenza commit nella repository	23
4.3.3.2.5	MPR005 Requisiti obbligatori non soddisfatti	23
4.3.3.2.6	MPR006 Requisiti desiderabili non soddisfatti	23
4.3.3.2.7	MPR007 Requisiti opzionali non soddisfatti	23
4.3.3.2.8	MPR008 Rischi non previsti avvenuti	23
4.3.3.3	MPR009 Frequenza controllo prodotti	23
4.3.3.4	Analisi statica	24
4.3.3.4.1	Analisi dei documenti	24
4.3.3.4.2	Analisi dei processi	24



Elenco delle tabelle

1	Metrica Indice di Gulpease	11
---	--------------------------------------	----

1 Introduzione

1.1 Premessa

Il DOCUMENTO_G che segue verrà prodotto incrementalmente al presentarsi della necessità di redigere nuove NORME_G. Per questo motivo, non è da considerare al pari di un documento completo (e.g. la parte relativa alla codifica non ci sarà fino al presentarsi di tale necessità).

1.2 Scopo del documento

Il presente documento ha l'obiettivo di mettere in chiaro le norme, le convenzioni e le tecnologie che verranno adottate da AlphaSix durante lo svolgimento del PROGETTO_G. Ogni membro del team è tenuto ad osservarlo rigorosamente, per mantenere consistenza ed omogeneità in ogni aspetto, durante tutta la durata del progetto.

1.3 Scopo del prodotto

Lo scopo del PRODOTTO_G è creare un APPLICATIVO_G per poter gestire i messaggi o le segnalazioni provenienti da diversi prodotti per la realizzazione di software. Queste segnalazioni passano attraverso un BROKER_G che gestisce i canali a loro dedicate per poi distribuirle ad applicazioni di messaggistica.

Il software dovrà inoltre essere in grado di riconoscere il TOPIC_G dei messaggi in input per poterli inviare in determinati canali a cui i destinatari dovranno iscriversi.

È anche richiesto di creare un canale specifico per gestire le particolari esigenze dell'azienda. Dovrà essere in grado, attraverso la lettura di particolari METADATI_G, di reindirizzare i messaggi ricevuti al destinatario più appropriato.

1.4 Glossario e documenti esterni

Al fine di rendere il documento più chiaro possibile, i termini che possono assumere un significato ambiguo o i riferimenti a documenti esterni avranno delle diciture convenzionali:

- **D**: indica che il termine si riferisce al nome di un particolare documento (ad esempio *PianoDiProgetto v1.0.0_D*).
- **G**: indica che il termine si riferisce ad una voce riportata nel Glossario (ad esempio REDMINE_G).

1.5 Riferimenti

1.5.1 Normativi

- ISO 8601
https://it.wikipedia.org/wiki/ISO_8601#0rari
- ISO/IEC 12207
https://en.wikipedia.org/wiki/ISO/IEC_12207
- Formato della data
https://it.wikipedia.org/wiki/Formato_della_data

1.5.2 Informativi

- Descrizione dei ruoli di progetto
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/RO.html>
- Visual Studio Code
<https://code.visualstudio.com/docs>
- TexStudio
<https://www.texstudio.org/>
- GanttProject
<https://www.ganttproject.biz/>
- L^AT_EX
<https://www.latex-project.org/help/documentation/>
- Redmine
<https://www.redmine.org/guide>
- GitLAB_G
<https://docs.gitlab.com/>
- APACHE KAFKA_G
<https://kafka.apache.org/documentation/>
- Telegram
<https://core.telegram.org/>
- DOCKER_G
<https://docs.docker.com/>
- The Twelve-Factor App
<https://12factor.net/it/>

2 Processi Primari

2.1 Processo di fornitura

2.1.1 Scopo

La sezione corrente ha lo scopo di riportare le attività principali che i membri di AlphaSix si propongono di attuare al fine di diventare FORNITORI_G per la proponente Imola Informatica.

2.1.2 Ricerca delle tecnologie

AlphaSix approfondisce la propria conoscenza su tecnologie e FRAMEWORK_G vari, in modo da discutere insieme quali sono quelle di interesse comune e di utilità per il progetto. Questo prevede poi l'auto-formazione di quelle scelte in comune accordo per acquisirne una buona padronanza.

2.1.3 Studio di Fattibilità

In quest'attività viene prodotto il documento *StudioDiFattibilità v1.0.0_D* al fine di analizzare ogni CAPITOLATO_G e scegliere quale contratto accettare. Nello specifico, il documento in ogni sezione contiene:

- **Descrizione generale:** breve descrizione del capitolato.
- **Obiettivo finale:** obiettivo da raggiungere.
- **Tecnologie coinvolte:** elenco delle tecnologie direttamente coinvolte esplicitate nel capitolato.
- **Valutazione conclusiva:** giudizio finale del team di sviluppo.

2.1.4 Preparazione in vista della revisione

In questa parte AlphaSix prepara tutto il materiale necessario al buon superamento della revisione, quali lettera di presentazione, documentazione, verbali e diapositive per l'esposizione.

2.2 Processo di sviluppo

2.2.1 Scopo

Lo sviluppo consiste nell'affrontare le attività volte a produrre il software richiesto dal proponente. Per una corretta implementazione di questo PROCESSO_G è necessario:

- Fissare degli obiettivi di sviluppo
- Realizzare un prodotto che sia conforme ai:
 - REQUISITI_G definiti dal proponente
 - Test definiti dalle norme di QUALITÀ_G

Lo standard ISO/IEC 12207:1995¹ definisce il processo di sviluppo quel processo contenente tutte le attività relative al prodotto finale, quali:

- Analisi dei requisiti
- Progettazione
- Codifica
- Integrazione ed installazione

¹Riferirsi alla voce "ISO/IEC 12207" in §1.5.1

2.2.2 Analisi dei Requisiti

Gli Analisti si occupano di redigere l'*AnalisiDeiRequisiti v1.0.0_D*, composta come segue:

- Descrizione generale del prodotto
- Modellazione concettuale del SISTEMA_G tramite la definizione dei vari CASI D'USO_G
- Classificazione e tracciamento dei requisiti

2.2.2.1 Denominazione dei requisiti

Ogni requisito che è stato individuato durante l'analisi presenta il seguente identificativo univoco:

R[Numero] [Tipo] [Priorità]

- **R**: si riferisce a requisito.
- **Numero**: corrisponde ad un numero che cerca di seguire la struttura del documento ed è progressivo. Inizia da 1.
- **Tipo**: segnala la tipologia di requisito che può essere di:
 - **F**: funzionalità, che ha a che vedere con le funzionalità del sistema software.
 - **Q**: qualità, che riguarda tecniche ad hoc.
 - **V**: vincolo, proposto da Imola Informatica.
- **Priorità**: indica il grado di urgenza per il soddisfacimento di un requisito, come:
 - **0**: opzionale, di grado basso e solo marginalmente utile.
 - **1**: desiderabile, di medio livello quindi non strettamente necessario ma che dà valore aggiunto.
 - **2**: obbligatorio, di grado alto quindi irrinunciabile per il COMMITTENTE_G e impossibile da tralasciare.

Esempio: R2Q1 indica il secondo requisito di qualità ed è desiderabile.

2.2.2.2 Casi d'uso

Un caso d'uso è una tecnica che identifica i requisiti funzionali descrivendo le interazioni tra il sistema di riferimento e un utente ad esso esterno.

Ogni caso d'uso che si vuol descrivere presenta:

- **Codice**: per l'identificazione del tipo:
 - UC[Numero]: per un caso principale (e.g. per il primo caso d'uso avremo UC1).
 - UC[Numero].[Numero]: per un sotto caso (e.g. per il primo figlio del primo caso d'uso avremo UC1.1).
- **Titolo**: denominazione del caso d'uso, possibilmente breve.
- **Attori primari**: tutti gli ATTORI_G primari coinvolti.
- **Attori secondari**: opzionale, tutti gli attori secondari coinvolti.
- **Descrizione**: per spiegare più nel dettaglio le azioni che vengono compiute.
- **Precondizione**: per rappresentare lo stato del sistema nell'istante precedente.

- **Postcondizione:** per rappresentare lo stato del sistema l'istante successivo.
- **Scenario principale:** contenente la serie di azioni da compiere numerate nell'ordine in cui vengono compiute.
- **Estensioni:** opzionale, per azioni inerenti a scenari alternativi come d'eccezione o errore.

2.2.3 Diagrammi UML

2.2.3.1 Diagrammi dei casi d'uso

Per i diagrammi dei casi d'uso viene utilizzato lo strumento DRAW.IO_G e il linguaggio UML_G 2.0. Essi descrivono la visione di un utente esterno al sistema e non danno nessun dettaglio implementativo. I COMPONENTI_G contenuti in questo tipo di diagrammi sono:

- ATTORE_G
- Casi d'uso
- Relazioni, che possono essere di:
 - **Associazione:** è sempre presente e rappresenta una comunicazione diretta dell'attore con il caso d'uso.
 - **Inclusione:** è una funzionalità comune che coinvolge più casi d'uso, in cui ogni istanza del primo esegue necessariamente il secondo. Tutte le inclusioni vengono sempre eseguite dall'utente.
 - **Estensione:** aumenta le funzionalità di un caso d'uso coinvolgendone un altro. L'esecuzione di quest'ultimo interrompe il precedente, ma non è necessariamente detto che tutte le estensioni vengano eseguite.
 - **Generalizzazione:** avviene tra casi d'uso o tra attori e rappresenta delle modifiche alle caratteristiche di base.

Per la specifica di un caso d'uso si fa riferimento al paragrafo §2.2.2.2.

2.2.4 Strumenti di base

2.2.4.1 Ambiente di sviluppo

Nei paragrafi successivi vengono riportate le componenti software utilizzate da ogni membro del team per lo sviluppo del progetto.

2.2.4.1.1 Sistema operativo

Il sistema operativo scelto da tutto il team è Ubuntu, in particolare ognuno ha una determinata versione di Ubuntu 18.

3 Processi Organizzativi

3.1 Gestione del progetto

3.1.1 Scopo

L'attività di gestione del progetto consiste nel delineare:

- I processi di progetto
- Le $RISORSE_G$ a essi necessarie
- I costi per la loro esecuzione
- I $TASK_G$ di risorsa umana
- La verifica delle attività di processo

3.1.2 Pianificazione di attività

3.1.2.1 Scopo

Responsabile e Amministratore insieme si occupano di generare il Piano di Progetto che tratta generalmente:

- **Modello di sviluppo:** per la decisione del modello di sviluppo che si intende adottare.
- **Analisi dei rischi:** in cui si identificano i vari possibili rischi e le strategie per evitarli o mitigarli.
- **Partizione del carico di lavoro:** che si occupa di delineare un preciso calendario di avanzamento e la rotazione dei ruoli.
- **Prospetto economico:** per la stima dei costi relativa alle risorse umane.

3.1.2.2 Obiettivo

L'obiettivo che ci si pone di raggiungere tramite il Piano di Progetto è organizzare il lavoro con $EFFICENZA_G$ ed $EFFICACIA_G$, prevedendo anticipatamente le varie attività da svolgere e determinando dei precisi periodi di tempo da dedicare ad ognuna di esse.

Più nello specifico, il *PianoDiProgetto v1.0.0_D* contiene:

- Introduzione e scopo
- Ciclo di vita con esposizione del modello di sviluppo
- Analisi dei rischi con annessa valutazione e classificazione
- $PIANIFICAZIONE_G$ delle attività in una visione temporale
- Prospetto economico del personale durante le attività
- $PREVENTIVO_G$ di ore e costi
- $ORGANIGRAMMA_G$ comprendente i componenti del team di sviluppo

3.1.2.3 Ordine di esecuzione

L'insieme di azioni che indicano in quale modo procedere sono riportate qui di seguito in ordine:

1. Individuazione di tutte le attività da svolgere basandosi sulle varie revisione da affrontare
2. Riconoscimento dei vari problemi in cui è possibile incorrere nel corso del progetto con conseguente analisi approfondita di ognuno essi. Questo comporta l'identificazione di:
 - La probabilità di verificarsi
 - La gravità della situazione
 - Le strategie da seguire per l'accertamento
 - Le contromisure da adottare per la risoluzione
3. Ordinamento delle attività tramite diagrammi di GANTT_G creati secondo alcuni fattori importanti da tenere in considerazione, quali:
 - Dipendenze tra attività
 - Sequenzialità
 - Possibilità di parallelismo
 - Andamento del progresso
 - Margine di SLACK TIME_G per consentire l'ammortizzamento di possibili rallentamenti
4. Stima delle risorse secondo la METRICA_G tempo/persona e le MILESTONE_G nel tempo, pianificando all'indietro
5. Assegnazione delle risorse umane alle attività secondo una precisa rotazione dei ruoli

3.1.2.4 Classificazione dei rischi

Le tipologie in cui suddividere i rischi sono state prese dal libro Software Engineering² e sono:

- **Organizzativo:** dovuto alla gestione di persone che hanno diverse responsabilità all'interno del progetto.
- **Personale:** riguarda le conoscenze, i tempi e la FORMAZIONE_G personale.
- **Requisiti:** ha a che fare con il numero di requisiti che può variare nel corso dello sviluppo del progetto.
- **Strumentale:** per l'utilizzo e la performance degli strumenti hardware.
- **Tecnologico:** per problemi riguardanti l'utilizzo e le funzionalità degli strumenti software.

A ciascun rischio viene assegnato un codice identificativo in modo da essere facilmente riconoscibile e per comprenderne le generalità (classe, probabilità e severità), per poi non doverle cercare nelle tabelle che comprendono tutti i rischi del progetto analizzati.

Questo codice è composto da:

[Tipologia] [ID] - [Gravità] [Probabilità] [Classe]

²Software Engineering, 10° edizione, Pagina 648, Capitolo 22: Project management

I valori che possono assumere sono:

- **Tipologia:**
 - **O:** organizzativo.
 - **P:** personale.
 - **R:** requisiti.
 - **S:** strumentale.
 - **T:** tecnologico.
- **ID:** numero progressivo di tre cifre (001 - 999);
- **Gravità:**
 - **0:** accettabile.
 - **1:** tollerabile.
 - **2:** inaccettabile.
- **Probabilità:**
 - **0:** bassa.
 - **1:** media.
 - **2:** alta.
- **Classe:** ci si riferisce ai livelli di rischio.
 - **0:** basso.
 - **1:** medio.
 - **2:** alto.

Ad esempio con P001-021 si può intuitivamente capire che si tratta del primo rischio relativo al personale, di gravità accettabile, probabilità alta e un valore di classe medio.

3.1.2.5 Ruoli di progetto

I ruoli³ adoperati per lo sviluppo del progetto sono:

- Analista
- Progettista
- Responsabile
- Amministratore
- Programmatore
- Verificatore

3.1.3 Pianificazione della qualità

Per la ricerca della qualità nei prodotti è necessario che ne vengano misurate diverse caratteristiche associate. Per ogni caratteristica devono essere fissati un livello minimo e un livello ottimale che questa deve avere. In tale sezione, viene descritto il modo di utilizzo sia delle metriche che degli strumenti utili ad ottenere qualità nei processi, nei documenti e nei prodotti. Gli strumenti utilizzati possono essere software, descritti nelle *NormeDiProgetto v1.0.0_D*, o standard di qualità, descritti nel *PianoDiQualifica v1.0.0_D*.

Gli obiettivi che ci si pone di raggiungere sono descritti nel *PianoDiQualifica v1.0.0_D*.

³Riferirsi alla voce “Descrizione dei ruoli di progetto” in §1.5.2.

3.1.3.1 Classificazione dei processi

I processi analizzati nel *PianoDiQualifica v1.0.0_D* sono classificati nel seguente modo:

PROC[ID] [Nome]

- **ID**: un numero incrementale per indicare in modo univoco il processo.
- **Nome**: una breve frase per indicare la funzione del processo.

3.1.3.2 Classificazione degli obiettivi

Gli obiettivi per la qualità (Q) concordati dall'Amministratore e dal Verificatore, descritti nel Piano di Qualifica, avranno il seguente codice identificativo:

Q[Tipo] [ID] [Nome]

- **Tipo**: indica la tipologia dell'oggetto di cui viene valutata la qualità, e questa può essere:
 - PR: processi.
 - PD: prodotti di documentazione.
 - PS: prodotti software.
- **ID**: ogni tipo di obiettivo possiede una lista ordinata attraverso un numero incrementale di tre cifre.
- **Nome**: offre un'informazione più chiara dell'obiettivo attraverso una breve frase.

Ad esempio:

- QPD001 Leggibilità del testo

3.1.3.3 Classificazione delle metriche

Ogni obiettivo della qualità deve, per quanto possibile, essere collegato ad una metrica, anch'essa scelta dall'Amministratore e dal Verificatore. Questo per valutare quantitativamente il raggiungimento o meno degli obiettivi stabiliti.

Le metriche (M) verranno classificate nel seguente modo:

M[Tipo] [ID] [Nome]

- **Tipo**: indica la tipologia di quello su cui viene applicata la metrica, può essere:
 - PR: processi.
 - PD: prodotti di documentazione.
 - PS: prodotti software.
- **ID**: ogni tipo di obiettivo possiede una lista ordinata attraverso un numero incrementale di tre cifre.
- **Nome**: offre un'informazione più chiara dell'obiettivo attraverso una breve frase.

Ad esempio:

- MPD001 Indice Gulpease

Quando sarà possibile, la metrica e l'obiettivo di qualità collegati fra loro avranno lo stesso Tipo e ID.

Ad esempio:

Obiettivo	Metrica	Valore desiderato
QPD001 Leggibilità del testo	MPD001 Indice Gulpease	50-60
Descrizione: quanto il testo è leggibile e comprensibile a livello sintattico lo stabilisce l'INDICE DI GULPEASE _G e una verifica da parte del Verificatore.		

Tabella 1: Metrica dell'Indice di Gulpease

3.1.4 Controllo del progetto

3.1.4.1 Monitoraggio del progetto

3.1.4.1.1 Monitoraggio dell'esecuzione dei processi

Ogni processo verrà controllato periodicamente durante tutta la sua esecuzione in modo da non intralciare il WAY OF WORKING_G, mediante misurazioni associate a metriche descritte dai processi di verifica, controllate tramite l'utilizzo di strumenti automatizzati. Le metriche adottate sono, per il processo considerato, indicatori di efficacia dei prodotti rispetto ai requisiti di funzionalità e qualità, stabiliti nel *PianoDiQualifica v1.0.0_D* e nell'*AnalisiDeiRequisiti v1.0.0_D*.

Risultano validi indicatori per la valutazione di:

- Aderenza al way of working
- Stato di avanzamento del processo rispetto alla pianificazione
- Identificazione dei problemi
- Eventuali ripianificazioni

3.1.4.1.2 Procedure di comunicazione

Per la coordinazione del team e le comunicazioni con il committente sullo stato del progetto, sono state stabilite le seguenti norme:

- Per le comunicazioni interne verranno utilizzati gli strumenti segnalati in §3.1.6, in particolare con l'utilizzo di SLACK_G.
- All'occorrenza, verranno fissate riunioni per le questioni più importanti, concordando:
 - Data, ora e luogo
 - Un ordine del giorno da discutere
 - La persona incaricata ad appuntare il contenuto della riunione per poter poi redigere formalmente il verbale dell'incontro
- Per le comunicazioni esterne verrà utilizzata prevalentemente la comunicazione via email e in caso di necessità si concorderà con l'azienda per riunioni via Skype o in un luogo prestabilito.

3.1.4.2 Gestione dei problemi emersi

Problemi emersi durante l'esecuzione dei processi verranno segnalati tramite `TICKETG`, utilizzando il sistema integrato di `GITHUBG` come descritto in dettaglio in §3.1.6.2. Questo permette di tracciare i problemi insorti durante le attività e di pianificare la risoluzione degli stessi. Per segnalazioni minori, verranno spesso utilizzati dei tag commentati nei `SORGENTIG LATEX`. Saranno nella forma:

[TAG]: [Descrizione]

I tipi di tag utilizzati dal team sono:

- **TODO**: per segnalare la presenza di un lavoro da fare o lasciato a metà.
- **FIXME**: per segnalare la presenza di una correzione da effettuare su del lavoro già svolto.

3.1.5 Chiusura dei processi

3.1.5.1 Archiviazione dei prodotti

I prodotti che, una volta terminate le attività e i processi per completarli, soddisfano le aspettative in termini di qualità, verranno archiviati in apposite `REPOSITORYG`. Alla versione *1.0.0* del presente documento, gli unici prodotti archiviati saranno i file `LATEX` per la generazione dei relativi PDF.

3.1.5.2 Archiviazione delle misurazioni

Oltre ai prodotti vengono archiviate le metriche, definite e classificate nel presente documento, con i limiti di accettazione riportati nel *PianoDiQualifica v1.0.0_D*. Il calcolo delle metriche verrà effettuato su ciascun prodotto nel momento del suo inserimento nella repository, ove possibile; tali risultati o anomalie verranno opportunamente archiviati.

3.1.6 Strumenti di pianificazione e coordinamento

3.1.6.1 Slack

Slack è uno strumento di collaborazione nato appositamente per coordinare il lavoro tra i team, permettendo la comunicazione in tempo reale e mettendo a disposizione molte altre utilità indispensabili. Tra queste, c'è la possibilità di dividere il `WORKSPACEG` in vari canali specifici, marcare parole o frasi importanti con diversi stili (grassetto, corsivo, codice, barrato), fissare i messaggi importanti, aprire una discussione sotto ogni messaggio per non creare troppi messaggi sul canale, ecc . . .

Il workspace usato dal team è stato suddiviso in diversi canali in base alle esigenze del periodo di lavoro. Nel periodo relativo alla stesura della versione attuale del documento, i vari canali sono:

- **# general**: canale in cui verranno discusse tematiche generali riguardanti il progetto e le sue attività.
- **# Analisi dei Requisiti**: per discussioni inerenti alla stesura del documento Analisi dei Requisiti. Nello specifico verranno discussi i requisiti e i casi d'uso.
- **# Piano di Progetto**: per discussioni inerenti la pianificazione, la suddivisione del lavoro, la valutazione dei rischi, la consuntivazione e altre attività volte alla redazione del Piano di Progetto.
- **# Piano di Qualifica**: per discussioni inerenti principalmente la qualità di processo e di prodotto, riportate nel Piano di Qualifica.

- **# Norme di Progetto:** per discussioni riguardanti le norme che il team di sviluppo adotterà e che ogni membro sarà tenuto a rispettare. Esse saranno riportate nelle Norme di Progetto.
- **# Studio di Fattibilità:** canale in cui verrà discussa la fattibilità di ogni capitolato proposto, convergenti nello Studio di Fattibilità.
- **# git:** canale in cui un BOT_G correttamente configurato manderà una notifica ogni volta che verrà effettuato un COMMIT_G, oppure alla chiusura o apertura di una ISSUE_G nella repository principale.
- **# latex:** per discussioni riguardanti gli aspetti tecnici di L^AT_EX.
- **# random:** per discussioni off-topic del team.

3.1.6.2 GitHub

GitHub fornisce un ISSUE TRACKING SYSTEM_G per ogni repository che ospita. Questo strumento offre più funzionalità di quanto potrebbe sembrare, poiché permette anche di trattare le issue come delle task, marcandole con una opportuna label. Infatti, questo sistema offre le seguenti funzionalità:

- Aprire una issue, attribuendone un titolo e una descrizione più dettagliata. Ogni issue è legata a un ID generato automaticamente in maniera incrementale (e.g. #32), al quale sarà possibile fare riferimento nei messaggi di commit.
- Creare delle milestone, e aggregare ad esse le issue.
- Assegnare delle label alle issue, per marcarne la natura. Ad esempio, una issue marcata come BUG_G sarà un problema da risolvere, una issue marcata come ENHANCEMENT_G sarà un miglioramento o una task. È possibile inoltre creare delle label personalizzate.
- Assegnare una issue a sé stessi o a un collaboratore, che sarà tenuta a correggerla.
- Supporto di MARKDOWN_G esteso, che permette di formattare le pagine delle issue in modo estremamente funzionale.

3.2 Formazione

La formazione di ogni membro di AlphaSix avviene tramite studio autonomo dei FRAMEWORK_G menzionati da Imola Informatica durante la presentazione del capitolato e incontri con AlphaSix.

3.2.1 Piano di formazione

Il piano di formazione prevede:

- Lo studio della documentazione⁴ di tecnologie e metodologie quali:
 - L^AT_EX
 - Redmine
 - GitLab
 - Apache Kafka
 - Telegram
 - Docker

⁴Vedere §1.5.2 per i link alla documentazione.



- API Rest
- The Twelve-Factor App
- Lo studio delle possibilità d'integrazione tra di esse

4 Processi di supporto

4.1 Documentazione

4.1.1 Implementazione

4.1.1.1 Template

Prima di iniziare a redigere i documenti, è stato creato un `TEMPLATEG` per `LATEX` (§4.1.4.2.1) contenente tutte le impostazioni grafiche condivise tra questi, per sfruttare il riutilizzo del codice e semplificare enormemente la manutenzione dei sorgenti.

Nello specifico, è presente un file per ognuna delle seguenti utilità:

- `LAYOUTG` delle pagine
- `MACROG` personalizzate volte a semplificare l'utilizzo di strutture o comandi ricorrenti
- Codice per la generazione della prima pagina (struttura definita in §4.1.2.1)
- Diario delle modifiche

4.1.1.2 Ciclo di vita dei documenti

Durante il suo ciclo di vita, ogni documento potrà trovarsi in una delle seguenti fasi:

- **Redazione:** fase che inizia con la creazione del documento e dura fino alla sua ultima approvazione. Il Responsabile assegna ai `REDATTORIG` le varie sezioni di ogni documento da redigere, i quali aggiorneranno la versione nel diario delle modifiche come normato in §4.1.5.1.
- **Verifica:** il documento entra in questa fase nel momento in cui i Redattori hanno terminato la stesura del lavoro loro assegnato, segnalandolo al Responsabile, che a sua volta assegnerà ai Verificatori la verifica della qualità del prodotto, secondo quanto riportato nelle norme di verifica. Essi potranno approvare il documento oppure notificare il Responsabile su eventuali errori o incongruenze emerse durante la fase di verifica, che provvederà a riassegnare il lavoro.
- **Approvazione:** fase che inizia dall'accettazione del documento da parte dei Verificatori nella fase di verifica. Spetta al Responsabile l'approvazione ufficiale del documento, seguita dal rilascio di una `MAJOR RELEASEG`.

4.1.2 Struttura

4.1.2.1 Frontespizio

La prima pagina di ogni documento, sarà caratterizzata da:

- Logo e nome del gruppo
- Titolo del documento
- Informazioni sul documento:
 - Versione documento
 - Data di creazione e ultima modifica
 - Nominativo dei Redattori
 - Nominativo dei Verificatori

- Nominativo del Responsabile
- Destinazione d'uso
- Destinatari del documento
- Contatto del gruppo
- Breve descrizione del documento

4.1.2.2 Storico delle versioni

La pagina che segue il frontespizio contiene lo storico delle versioni del documento, in cui ogni aggiunta o modifica significativa ha comportato un incremento di versione. Ogni riga contiene, a partire da sinistra:

- Il numero della versione nel formato espresso in §4.1.5.1
- Una breve descrizione delle modifiche apportate
- Il ruolo dell'autore che ha apportato la modifica
- Il nominativo dell'autore
- La data di modifica

La chiave primaria della tabella è il numero di versione ordinata in senso decrescente, in modo che la versione più vecchia sia l'ultima riga della tabella.

4.1.2.3 Indice

In ogni documento, esclusi i verbali, è presente un indice contenente tutte le sezioni, sottosezioni e paragrafi. I numeri di sezioni, sottosezioni, e paragrafi sottostanti saranno separati da un punto (e.g. 1.4.1).

Saranno eventualmente presenti un indice delle figure e un indice delle tabelle, assenti in caso non ci siano tabelle o figure nel documento.

I valori degli indici partono da 1.

4.1.2.4 Contenuto

La struttura di ogni pagina presenta:

- Intestazione con:
 - A sinistra, logo di AlphaSix
 - A destra, nome del capitolato e documento corrente
- Piè di pagina con:
 - A sinistra, nome e mail di riferimento del gruppo
 - A destra, numero della pagina corrente

4.1.3 Design

4.1.3.1 Norme tipografiche

Le norme tipografiche qui di seguito elencate sono state decise in modo che ogni membro di AlphaSix concorra a mantenere una forma coerente e univoca per tutti i documenti redatti.

4.1.3.1.1 Stile del testo

- **Corsivo:** solo per i nomi dei documenti citati.
- **Maiuscolo:** la prima lettera per
 - Tutte le parole appartenenti ai nomi dei documenti tranne gli articoli
 - Nomi dei ruoli
 - Prima parola degli elenchi puntati

4.1.3.1.2 Elenchi puntati

- **Simboli di livello:** un pallino nero per il primo livello, un trattino per il secondo livello.
- **Punteggiatura:** nessuna punteggiatura alla fine di una frase, tranne nel caso in cui sia presente una descrizione. In quel caso la descrizione è preceduta dai due punti “:” e termina con un punto “.”.
- **Grassetto:** solo se è presente una descrizione, allora sono in grassetto tutte le parole prima dei due punti “:”.

4.1.3.1.3 Altri formati testuali comuni

- **Orari:** HH:MM secondo la norma ISO 8601⁵ nel formato 24 ore dove:
 - HH indica le ore, da 00 a 23
 - MM i minuti, da 00 a 59
- **Date:** DD-MM-YYYY formato adottato in Europa dove:
 - DD indica il numero del giorno, da 01 a 31
 - MM il mese, da 01 a 12
 - YYYY l’anno
- **Nota a piè di pagina:** serve ad inserire elementi aggiuntivi, come osservazioni o riferimenti a parti interne al documento, utili alla comprensione del testo, ma se inseriti all’interno del discorso, ne interromperebbero la lettura, rendendola meno scorrevole.

4.1.3.2 Elementi grafici

4.1.3.2.1 Figure

Ogni immagine inserita nei documenti deve sempre essere centrata rispetto al foglio e adeguatamente separata dal testo. Deve inoltre essere accompagnata da una breve CAPTION_G che permetta al lettore di capire esattamente che cosa sta guardando.

È presente nell’indice l’elenco delle figure che raccoglie la lista di tutte le immagini presenti.

4.1.3.2.2 Tabelle

Come per le figure, ogni tabella sarà accompagnata da una caption e sarà della dimensione del testo, o se più piccola, centrata. Tutte le tabelle saranno raccolte nell’elenco delle tabelle.

Saranno presenti due tipologie di tabelle:

- **Semplici:** tabelle standard senza uno stile particolare, in cui le celle sono separate da bordi neri (evitare, ove non risulta necessario, le righe verticali).

⁵Riferirsi alla voce “ISO 8601” in §1.5.1

- **Complesse:** tabelle con un'alternanza di colori tra le righe delle celle (grigio e bianco) e senza bordi verticali. Le celle sono separate orizzontalmente da una corretta spaziatura e allineamento e verticalmente dall'alternanza dei due colori. La riga dell'`HEADERG` può essere bianca o di un grigio più scuro in base al contesto, con il testo che può essere in grassetto.

4.1.4 Produzione

4.1.4.1 Suddivisione dei documenti

4.1.4.1.1 Documenti interni

Sono considerati interni documenti quali Studio di Fattibilità e Norme di Progetto, che non sono visibili ad entità esterne, ma solo ad AlphaSix.

4.1.4.1.2 Documenti esterni

Sono considerati esterni, invece, documenti quali Piano di Progetto, Piano di Qualifica e Analisi dei Requisiti che, al contrario dei precedenti, sono visibili al committente.

4.1.4.1.3 Verballi

Questi documenti vengono redatti successivamente alle riunioni tenute da AlphaSix o in caso di incontri con `STAKEHOLDERG` esterni, per esempio con Imola Informatica. Una singola persona ha il compito di stendere la relazione relativa al verbale, presentando le seguenti sezioni:

- **Informazioni incontro:** lista delle informazioni principali riguardanti la riunione quali luogo, data, orario, ordine del giorno, ecc. . .
- **Argomenti:** lista dei principali argomenti trattati con descrizione di cosa si è discusso nel dettaglio.

4.1.4.2 Strumenti di supporto

4.1.4.2.1 \LaTeX

Per la stesura della documentazione è stato scelto di usare il linguaggio di `MARKUPG` \LaTeX perché presenta molti vantaggi, tra i quali:

- Supporta nativamente il `VERSIONAMENTOG`, essendo un linguaggio compilato
- Supporta la modularità, rendendo più facile organizzare un documento dividendone logicamente i vari moduli
- Permette il riutilizzo del codice tramite l'uso di macro già pronte o personalizzate, oppure includendo lo stesso `SORGENTEG` in punti diversi
- Gestisce automaticamente indici e riferimenti

4.1.4.2.2 TexStudio/Visual Studio Code

TexStudio e Visual Studio Code sono i due ambienti di sviluppo scelti da AlphaSix per stilare la documentazione. TexStudio è un IDE_G nativo per l'utilizzo di \LaTeX . Visual Studio Code è un editor intelligente moderno (alla pari di un IDE_G) che, tramite estensioni, permette il supporto di praticamente ogni linguaggio. Entrambi permettono una rapida compilazione e un'istantanea visualizzazione dell'anteprima del PDF prodotto, oltre agli altri vantaggi che ogni IDE offre, tra

cui: suggerimenti e completamenti automatici delle parole chiave, ricerca intelligente (eventualmente tramite `REGEXPG`) controllo ortografico della lingua italiana o inglese e così via.

Più informazioni sono reperibili sui rispettivi siti ufficiali, i cui link sono presenti in §1.5.2.

4.1.4.2.3 GanttProject

GanttProject è un programma gratuito dedicato alla formazione dei diagrammi di Gantt. Permette di creare task e milestone, organizzare le task in lavoro strutturato a interruzioni, disegnare i vincoli di dipendenza tra di esse e molte altre utilità, generando automaticamente il relativo diagramma. Per maggiori informazioni, si rimanda alla fonte ufficiale (consultare §1.5.2).

4.1.4.2.4 Draw.io

Draw.io è un'applicazione web in grado di creare diagrammi UML, di Entità-Relazionale, di flusso e molto altro. Il motivo che ha portato AlphaSix a scegliere questo strumento è la sua perfetta integrazione con `GOOGLE DRIVEG`, oltre al suo alto livello di intuitività. Questo vantaggio permette poter condividere i diagrammi creati tra tutti i collaboratori in ogni momento e in modo automatico. Per maggiori informazioni, visualizzare la fonte ufficiale (§1.5.2).

4.1.4.2.5 Indice di Gulpease

Per il calcolo dell'indice di Gulpease, è stato creato uno script ad hoc che, preso in input un file PDF, produce in output l'indice di Gulpease.

4.1.5 Mantenimento

4.1.5.1 Versionamento

Tutti i documenti redatti supportano il versionamento, in modo da essere univoci e per rendere disponibile la possibilità di consultare versioni precedenti in qualsiasi fase del loro `CICLO DI VITAG`. Il modello di versionamento adottato segue lo schema `CHANGE SIGNIFICANCEG`.

La versione di un file è espressa secondo la notazione

$$X.Y.Z$$

dove:

- **X** indica il numero di versione principale. Inizia da 0 e viene incrementato ogni volta che il Responsabile approva il documento, determinando una major release
- **Y** indica il numero di versione secondario, contatore delle fasi di verifica effettuate dal Verificatore superate positivamente. Inizia da 0. Viene azzerato ad ogni incremento della **X**
- **Z** è l'indice di modifica minore, incrementato ogni volta che viene effettuato un aggiornamento inferiore, come l'aggiunta di una sezione o paragrafo. Viene azzerato ad ogni incremento della **Y** o della **X**

4.1.5.2 Continuous Integration

Per quanto riguarda la stesura dei documenti, verrà adottato il principio di `CONTINUOUS INTEGRATIONG`, che sarà tuttavia limitato in questo periodo a sincronizzarsi il prima possibile con la repository remota (non essendoci una vera e propria build o dei test da effettuare), sia per quanto riguarda il `FETCHG` che per il `PUSHG`.

Questo serve a rendere più remota possibile la probabilità di incappare nell'`INTEGRATION HELLG`.

4.1.5.3 Nomenclatura

4.1.5.3.1 Verball

I Verball possono essere interni oppure esterni, nel caso in cui il team incontri gli esponenti di Imola Informatica. Il nominativo del file in cui sono formalizzati è il seguente:

- VI_dd-mm-yyyy.pdf per i verball interni
- VE_dd-mm-yyyy.pdf per i verball esterni

dove dd-mm-yyyy è la data in cui sono stati tenuti, nel formato descritto nel paragrafo §4.1.3.1.3.

4.1.5.3.2 Documenti vari

Saranno presenti due tipologie di file: file interni ad AlphaSix e file esterni.

- La prima categoria include moduli di L^AT_EX contenenti le varie sezioni e che non verranno mai esposti esternamente. Questi file verranno denominati usando la convenzione `snake_case.tex`, dove `snake_case` è il nome della sezione o modulo
- La seconda categoria include i file `.tex` principali che produrranno i PDF da consegnare al committente. Essi verranno denominati con la convenzione `CamelCase vX.Y.Z.pdf`, dove `CamelCase` sarà il nome del documento generico mentre `vX.Y.Z` sarà la versione che identifica univocamente il documento come descritto in §4.1.5.1

4.2 Verifica

4.2.1 Scopo

Questa sezione vuole descrivere come il team di sviluppo esegue la fase di verifica, per capire se i prodotti e i processi sono conformi a quanto ci si attende. La fase di verifica serve per stabilire se è opportuno o meno procedere alla fase successiva del progetto, se ciò non fosse possibile sarà necessario il ritorno ad una fase stabile del progetto per poi ripartire da lì, prendendo in considerazione i risultati della precedente verifica.

4.2.2 Descrizione

Ogni processo e prodotto deve essere valutato in modo quantificabile attraverso metriche apposite, quando possibile, e stabilendo il risultato che si vuole raggiungere.

Come indicato dal CICLO DI DEMING_G nel *PianoDiQualifica v1.0.0_D*, nel momento in cui tale risultato sarà raggiunto, se esso non è il migliore, servirà come “base” per alzare il livello di qualità di quel processo o prodotto.

I risultati ottenuti nella fase di verifica sono riportati nel Piano di Qualifica, in questo modo, confrontandoli con gli esiti attesi, è possibile valutare un miglioramento per i vari processi e prodotti sempre nel Piano di Qualifica.

4.3 Walkthrough e Inspection

Due modi di effettuare la verifica sono attraverso WALKTHROUGH_G e Inspection. Il team di sviluppo li adotta entrambi, ma non contemporaneamente, per i vantaggi che comporta ogni metodo.

4.3.1 Walkthrough

Metodo di verifica che effettua un controllo ad ampio spettro senza l'assunzione di presupposti. Dato che per mettere in atto tale metodo ci si deve dividere in gruppi dove ognuno ha un ruolo ben distinto, Walkthrough è ideale per le verifiche effettuate all'inizio dei vari periodi, dove i membri del team di sviluppo non possiedono le conoscenze adeguate per una verifica efficiente.

Walkthrough possiede delle fasi ben specifiche:

1. **Pianificazione:** viene pianificato in gruppo come effettuare la verifica dei prodotti.
2. **Lettura:** viene effettuata la lettura del prodotto.
3. **Discussione:** vengono discusse le possibili correzioni.
4. **Correzione di difetti:** si attuano i risultati ottenuti dalla fase di discussione.

4.3.2 Inspection

Metodo di verifica dove si esegue una lettura mirata dei prodotti, frutto di un'analisi dei risultati dei precedenti test. Questo metodo dunque, a differenza di Walkthrough, prevede l'esecuzione con dei presupposti.

Le fasi di Inspection sono:

1. **Pianificazione:** viene sempre pianificato in gruppo come effettuare la pianificazione.
2. **Definizione di una lista di controllo:** dato che le parti da verificare sono specificate, queste possono essere riportate in una lista in modo da velocizzare il processo di verifica.
3. **Correzione dei difetti:** attuazione dei punti della lista di controllo.

Per la natura di Inspection, questa non può essere applicata fin dall'inizio, dunque nel momento in cui si presentano nuove tipologie di prodotti e processi da verificare viene effettuato Walkthrough, nel momento in cui il metodo di verifica è ben consolidato da tutti i membri del team di sviluppo si passa ad effettuare verifica secondo Inspection.

4.3.3 Metriche

La denominazione delle metriche è già stata descritta in §3.1.3.3, qui viene descritto il loro funzionamento.

4.3.3.1 Metriche per i documenti

4.3.3.1.1 MPD001 Indice di Gulpease

Si tratta di un indice di leggibilità dei documenti in lingua italiana. A differenza degli indici per le altre lingue, questo si basa sulla lunghezza delle parole in lettere e non in sillabe. L'indice si calcola:

$$89 + \frac{300 \times n_{\text{frasi}} - 10 \times n_{\text{lettere}}}{n_{\text{parole}}}$$

Metrica: il risultato della formula è interpretato nel seguente modo

- **<80:** documento difficile da leggere per chi ha la licenza elementare.
- **<60:** documento difficile da leggere per chi ha la licenza media.
- **<40:** documento difficile da leggere per chi ha un diploma superiore.

Nel momento in cui avviene un commit all'interno di repository, in automatico si avvia uno script che analizza tutti i documenti in PDF per valutarne l'indice di Gulpease. I risultati vengono poi riportati in un apposito file di testo per verificarne la qualità ed un possibile miglioramento.

4.3.3.1.2 MPD002 Correttezza ortografica

Gli errori ortografici possono essere segnalati dallo strumento di Controllo Ortografico presente in *TexStudio*.

Metrica: il numero di errori ortografici presenti nel documento.

4.3.3.2 Metriche per i processi

4.3.3.2.1 MPR001 Varianza della pianificazione

Nel documento *PianoDiProgetto v1.0.0_D*, sono stabilite le $BASELINE_G$ e le scadenze di consegna dei vari prodotti. Nonostante il tempo di slack che ogni fase possiede, è possibile che delle date non vengano rispettate causa incidenti di vario tipo.

$$\frac{\sum_{i=1}^n x_i}{n} \quad x_i = \text{numero di giorni di ritardo per la } i\text{-esima scadenza.}$$

Metrica: viene indicata la media di giorni di ritardo nel rispettare una scadenza durante il macro-periodo. Se un'attività viene conclusa con in d'anticipo, il numero di giorni di ritardo non è negativo ma nullo.

4.3.3.2.2 MPR002 Varianza dei costi

All'interno del *PianoDiProgetto v1.0.0_D* è indicato il costo approssimativo del progetto. In corso d'opera possono presentarsi dei problemi che richiedono un'aggiunta di costo in termini di $\frac{\text{tempo}}{\text{persona}}$. Lo scopo del preventivo è infatti fare una stima non definitiva dei costi. Per verificare se il preventivo è rispettato viene fatto periodicamente un consuntivo di periodo.

Metrica: viene misurata la differenza conteggiata in € tra il costo finale e il preventivo. Viene seguito tale schema per la tariffa oraria dei vari ruoli del team di sviluppo:

- **Responsabile:** € 30
- **Amministratore:** € 20
- **Analista:** € 25
- **Progettista:** € 22
- **Programmatore:** € 15
- **Verificatore:** € 15

4.3.3.2.3 MPR003 Aderenza agli standard

Per misurare e verificare i processi sono stati scelti alcuni specifici standard di qualità descritti nel Piano di Qualifica che possono offrire una valutazione quantitativa. Gli standard scelti sono:

- **ISO/IEC 15504:** ogni processo attivato verrà classificato e valutato secondo gli attributi assegnati ai vari livelli di qualità. Per ogni attributo verrà infine indicata una percentuale di quanto il processo rispetti l'attributo, potendo infine capire nel complesso quanto quel processo riesca a superare un dato livello di maturità.
- **Ciclo di Deming:** nella fase migliorativa del processo sarà data particolare attenzione nel non iniziare una fase del Ciclo di Deming senza aver finito completamente le fasi precedenti.

Metrica: il livello di maturità è descritto nell'ISO/IEC 15504 nel *PianoDiQualifica v1.0.0_D*.

4.3.3.2.4 MPR004 Frequenza commit nella repository

Per mantenere aggiornate le versioni dei prodotti è necessario che ogni membro di AlphaSix effettui un commit ad ogni sua modifica significativa. Così facendo, in caso di errori, è possibile tornare ad una versione stabile del progetto. Misurare quanti commit sono stati effettuati inoltre serve per monitorare l'attività del team di sviluppo.

$$\frac{\sum_{i=1}^n x_i}{n} \quad x_i = \text{numero di commit alla } i\text{-esima settimana del macro periodo.}$$

Metrica: numero minimo di commit da effettuare in media ogni settimana lavorativa durante un macro-periodo.

Metriche sui requisiti e i rischi

Dato che non è fisso il numero dei requisiti di un progetto, sono state scelte una serie di metriche dove il valore ottimale da raggiungere è sempre uguale, lo zero. È stato scelto di contare il numero di requisiti non soddisfatti invece che il contrario. Lo stesso ragionamento è valido per quanto riguarda i rischi che possono verificarsi nel corso del progetto.

Gli obiettivi che si vogliono raggiungere attraverso tali metriche possono essere stabiliti solo a progetto concluso.

4.3.3.2.5 MPR005 Requisiti obbligatori non soddisfatti

Per adempiere completamente alla richiesta del cliente, serve individuare tutti i requisiti presenti nella sua richiesta, impliciti, espliciti, diretti e derivati. Alcuni sono imprescindibili, detti obbligatori, e il loro soddisfacimento determina la buona riuscita del progetto.

Metrica: numero dei requisiti obbligatori non soddisfatti.

4.3.3.2.6 MPR006 Requisiti desiderabili non soddisfatti

I requisiti desiderabili non sono necessari, ma offrono un valore aggiunto al progetto.

Metrica: numero dei requisiti desiderabili non soddisfatti.

4.3.3.2.7 MPR007 Requisiti opzionali non soddisfatti

Tali requisiti dovranno essere adempiuti solo nel momento in cui tutti i requisiti obbligatori saranno soddisfatti. Possono essere concordati col cliente in corso d'opera.

Metrica: numero dei requisiti opzionali non soddisfatti.

4.3.3.2.8 MPR008 Rischi non previsti avvenuti

Nell'Analisi dei rischi presente nel *PianoDiProgetto v1.0.0_D*, sono presenti i rischi ritenuti possibili per i quali è proposta una soluzione. Possono presentarsi anche rischi non previsti in tale analisi. Questi devono essere il meno possibili (nulli) perché la loro soluzione sarà decisa al momento causando ritardi all'interno del progetto.

Metrica: numero di rischi non previsti avvenuti nel corso dell'intero progetto.

4.3.3.3 MPR009 Frequenza controllo prodotti

I documenti e i prodotti software hanno bisogno di una verifica frequente, commisurata in base al numero di modifiche che vengono apportate.

Chi esegue la modifica deve controllare ciò che ha fatto prima di poterla ufficializzare, mentre la verifica fatta dal Verificatore deve essere fatta solo nel momento in cui è stato raggiunto un numero significativo di modifiche, per evitare di spendere troppe risorse in questa fase.

$$\frac{\sum_{i=1}^n x_i}{n} \quad x_i = \text{numero di modifiche effettuate prima della } i\text{-esima verifica.}$$

Metrica: media del numero massimo di modifiche apportate ai prodotti prima che ricevano una verifica dal parte del Verificatore.

4.3.3.4 Analisi statica

L'analisi statica verifica solo i prodotti che non sono o che non possono essere eseguiti, come i documenti.

4.3.3.4.1 Analisi dei documenti

L'analisi statica per i documenti si limita a valutare come e con che contenuti questi vengono scritti. Per l'analisi dei documenti vengono utilizzate:

- MPD001
- MPD002

4.3.3.4.2 Analisi dei processi

Per analizzare i processi vengono usati gli standard sopra elencati. Ad ogni fase del processo verranno valutati gli attributi richiesti secondo l'ISO 15504, in che misura questi sono stati rispettati e in che fase del Ciclo di Deming il processo si trova. Per l'analisi dei processi vengono utilizzate:

- MPR001 Varianza della pianificazione
- MPR002 Varianza dei costi
- MPR004 Frequenza commit nella repository
- MPR009 Frequenza controllo prodotti