



AlphaSix

AlphaSix

Piano di Qualifica

Informazioni sul documento

Nome Documento	PianoDiQualifica v2.0.0.pdf
Versione	2.0.0
Data di Creazione	27 novembre 2018
Data ultima modifica	05 marzo 2019
Stato	Approvato
Redazione	Laura Cameran
Verifica	Timoty Granziero
Approvazione	Samuele Gardin
Uso	Esterno
Distribuzione	AlphaSix
Destinato a	Prof. Tullio Vardanega, Prof. Riccardo Cardin, Imola Informatica
Email di riferimento	alpha.six.unipd@gmail.com

Descrizione

Questo documento descrive gli obiettivi che AlphaSix si prefigge di raggiungere per ottenere processi e prodotti di qualità.

Registro delle modifiche

Versione	Descrizione	Ruolo	Nominativo	Data
2.0.0	Approvazione per il rilascio	Responsabile	Samuele Gardin	2019-03-05
1.9.8	Verifica finale	Verificatore	Timoty Granziero	2019-03-03
1.9.2	Aggiunta D.2	Verificatore	Nicola Carlesso	2019-03-02
1.9.0	Aggiunta parte di B.2	Verificatore	Nicola Carlesso	2019-03-02
1.2.0	Verifica	Verificatore	Samuele Gardin	2019-02-28
1.1.5	Aggiunta sezione §3.3.2 e aggiunta tabella in §3.4	Progettista	Laura Cameran	2019-02-26
1.1.4	Completate tabelle in §4.5 e §4.5.1	Progettista	Laura Cameran	2019-02-23
1.1.3	Sistematte sezioni §4.3.1 e §4.5.1	Progettista	Laura Cameran	2019-02-22
1.1.2	Aggiunte sezioni §4.3.1 e §4.5.1	Verificatore	Laura Cameran	2019-02-19
1.1.1	Ampliamento §B	Progettista	Ciprian Voinea	2019-02-13
1.1.0	Verifica	Verificatore	Nicola Carlesso	2019-02-10
1.0.4	Aggiunte sezioni §4.3, §4.5 e rivisitazione §4.2	Verificatore	Laura Cameran	2019-02-06
1.0.3	Aggiunto §C	Verificatore	Matteo Marchiori	2019-02-03
1.0.2	Sistemazione §B	Progettista	Nicola Carlesso	2019-02-01
1.0.1	Aggiunte sezioni §4, §4.1 §4.4 e §4.2	Progettista	Nicola Carlesso	2019-01-27
1.0.0	Approvazione per il rilascio	Responsabile	Nicola Carlesso	2019-01-13
0.2.0	Verifica finale	Verificatore	Matteo Marchiori	2019-01-12
0.1.2	Aggiunto §D	Verificatore	Matteo Marchiori	2019-01-11
0.1.1	Aggiunta §B	Verificatore	Nicola Carlesso	2019-01-08
0.1.0	Verifica documento	Verificatore	Ciprian Voinea	2018-12-10
0.0.5	Aggiunta §A	Verificatore	Nicola Carlesso	03-12-2018
0.0.4	Aggiunto §2	Verificatore	Nicola Carlesso	02-12-2018
0.0.3	Aggiunto §3	Verificatore	Timoty Granziero	01-12-2018
0.0.2	Aggiunto §1	Verificatore	Nicola Carlesso	29-11-2018
0.0.1	Creazione template	Redattore	Timoty Granziero	27-11-2018

Indice

1	Introduzione	2
1.1	Glossario e documenti esterni	2
1.2	Premessa	2
1.3	Scopo del documento	2
1.4	Scopo del prodotto	2
1.5	Riferimenti	2
1.5.1	Normativi	2
1.5.2	Informativi	3
2	Qualità di processo	4
2.1	Scopo	4
2.2	Nomenclatura metriche e obiettivi di qualità	4
2.3	Processi	4
2.3.1	PROC001 Pianificazione del progetto, organizzazione e struttura	4
2.3.2	PROC002 Analisi	5
2.3.3	PROC003 Produzione documenti	6
2.3.4	PROC004 Verifica	7
2.4	Tabella qualità di processo	7
3	Qualità di prodotto	11
3.1	Scopo	11
3.2	Nomenclatura metriche ed obiettivi di qualità	11
3.3	Prodotti	11
3.3.1	Documenti	11
3.3.2	Software	12
3.4	Tabelle qualità di prodotto	13
4	Test	18
4.1	Modello a V	18
4.2	Classificazione e struttura dei test	18
4.3	Test di validazione	19
4.3.1	Tracciamento	34
4.4	Test di sistema	36
4.4.1	Tracciamento	39
4.5	Test d'integrazione	41
4.5.1	Tracciamento	41
A	Standard di qualità	42
A.1	ISO/IEC 15504 (SPICE)	42
A.2	ISO/IEC 9126:2001	43
A.2.1	Descrizione degli attributi della qualità interna e della qualità esterna	44
A.2.2	Descrizione degli attributi della qualità in uso	45
A.3	Ciclo di Deming	46
A.4	ISO/IEC 90003:2004	47
A.4.1	Capitolo 4. dell'ISO 90003: requisiti di sistema e linee guida	47

B	Resoconto delle attività di verifica	49
B.1	Classificazione dei risultati	49
B.2	Primo periodo (RR)	49
B.2.1	Riassunto delle attività di verifica	49
B.2.2	Risultati delle verifiche tramite analisi	49
B.2.2.1	Documenti	50
B.2.2.2	Processi	51
B.3	Secondo periodo (RP)	52
B.3.1	Riassunto delle attività di verifica	52
B.3.2	Risultati delle verifiche tramite analisi	52
B.3.2.1	Documenti	53
B.3.2.1.1	MPD001 Indice di Gulpease	53
B.3.2.1.2	MPD002 Correttezza ortografica	54
B.3.2.2	Processi	55
B.3.2.2.1	MPR001 Varianza della pianificazione	55
B.3.2.2.2	MPR002 Varianza dei costi	56
B.3.2.2.3	MPR003 Aderenza agli standard	57
B.3.2.3	Software	59
B.3.2.3.1	MPS001 Presenza di bug	59
B.3.2.3.2	MPS002 Presenza di vulnerabilità	60
B.3.2.3.3	MPS003 Presenza di code smell	61
B.3.2.3.4	MPS004 Duplicazione del codice	62
C	Mitigazione delle variazioni orarie	63
C.1	Strumenti e mitigazione	63
C.2	Mitigazione dei costi	63
C.3	Conclusioni	63
D	Valutazioni per il miglioramento	64
D.1	Primo periodo (RR)	64
D.1.1	Valutazioni sull'organizzazione	64
D.1.2	Valutazione dei ruoli	64
D.1.2.1	Responsabile	64
D.1.2.2	Amministratore	64
D.1.2.3	Analista	64
D.1.2.4	Verificatore	65
D.1.3	Valutazione sugli strumenti	65
D.1.3.1	L ^A T _E X	65
D.1.3.2	Git	65
D.1.4	Integrità di prodotti e strumenti	65
D.1.5	Applicazione dei miglioramenti dopo la RR	65
D.2	Secondo periodo (RP)	66
D.2.1	Valutazioni sull'organizzazione	66
D.2.2	Valutazione dei ruoli	66
D.2.2.1	Analista	66
D.2.2.2	Progettista	67
D.2.2.3	Programmatore	67
D.2.2.4	Verificatore	67
D.2.3	Valutazione sugli strumenti	67
D.2.3.1	Docker	67

Elenco delle tabelle

1	Obiettivi di qualità per il PROC001	8
2	Obiettivi di qualità per il PROC002	9
3	Obiettivi di qualità per il PROC003	9
4	Obiettivi di qualità per il PROC004	10
5	Obiettivi di qualità per i documenti	14
6	Obiettivi di qualità per il software (1)	15
7	Obiettivi di qualità per il software (2)	16
8	Obiettivi di qualità per il software (3)	17
9	Elenco dei test di validazione (1)	19
10	Elenco dei test di validazione (2)	20
11	Elenco dei test di validazione (3)	21
12	Elenco dei test di validazione (4)	22
13	Elenco dei test di validazione (5)	23
14	Elenco dei test di validazione (6)	24
15	Elenco dei test di validazione (7)	25
16	Elenco dei test di validazione (8)	26
17	Elenco dei test di validazione (9)	27
18	Elenco dei test di validazione (10)	28
19	Elenco dei test di validazione (11)	29
20	Elenco dei test di validazione (12)	30
21	Elenco dei test di validazione (13)	31
22	Elenco dei test di validazione (14)	32
23	Elenco dei test di validazione (15)	33
24	Elenco dei test in correlazioni con i requisiti (1)	34
25	Elenco dei test in correlazioni con i requisiti (2)	35
26	Elenco dei test di sistema (1)	36
27	Elenco dei test di sistema (2)	37
28	Elenco dei test di sistema (3)	38
29	Elenco dei test in correlazioni con i requisiti (1)	39
30	Elenco dei test in correlazioni con i requisiti (2)	40
31	Elenco dei test d'integrazione	41
32	Elenco dei test in correlazioni con le componenti	41
33	Schema degli attributi di ISO/IEC 15504	43
34	Risultati di MPD001 Indice Gulpease	50
35	Risultati di MPD002 Correttezza ortografica	50
36	Risultati di MPR001 Varianza della pianificazione	51
37	Risultati di MPR002 Varianza dei costi	51
38	Risultati di MPR003 Aderenza agli standard	51
39	Risultati di MPR004 Frequenza commit nella repository	52
40	Risultati di MPR009 Frequenza controllo prodotti	52

Elenco delle figure

1	Modello a V	18
2	Schema ISO 9126 con il suo ciclo di vita	46
3	Ciclo di Deming	47
4	Diagramma con valori misurati tramite MPD001 Indice di Gulpease	53
5	Diagramma con valori misurati tramite MPD002 Correttezza ortografica	54
6	Diagramma con valori misurati tramite MPR001 Varianza della pianificazione	55



7	Diagramma con valori misurati tramite MPR002 Varianza dei costi	56
8	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (1) . . .	57
9	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (2) . . .	57
10	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (3) . . .	58
11	Diagramma con valori misurati tramite MPS001 Presenza di bug	59
12	Diagramma con valori misurati tramite MPS002 Presenza di vulnerabilità	60
13	Diagramma con valori misurati tramite MPS003 Presenza di code smell	61
14	Diagramma con valori misurati tramite MPS004 Duplicazione del codice	62

1 Introduzione

1.1 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 v2.0.0_D*).
- **G**: indica che il termine si riferisce ad una voce riportata nel Glossario (ad esempio *REDMINE_G*).

1.2 Premessa

Il DOCUMENTO_G che segue verrà prodotto incrementalmente al presentarsi di esigenze di perseguimento della QUALITÀ_G. Per questo motivo, non è da considerare al pari di un documento completo (e.g. la parte relativa ai risultati dei test non ci sarà fino al presentarsi della necessità di testare).

1.3 Scopo del documento

Lo scopo di questo documento è riportare formalmente tutte le NORME_G che rispettiamo per perseguire la qualità di prodotto, sia per quanto riguarda il prodotto in sé, sia per quanto riguarda i PROCESSI_G utilizzati per costruirlo. Per fare ciò, effettuiamo continuamente verifiche atte a impedire a qualsiasi tipo di malformità o errore di rimanere presente per troppo tempo, rendendo più facile la manutenzione.

1.4 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 a determinati canali a cui i destinatari dovranno iscriversi.

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

1.5 Riferimenti

1.5.1 Normativi

- *NormeDiProgetto v2.0.0_D*
- CAPITOLATO_G d'appalto C1:
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1.pdf>
- Documentazione PEP 8_G:
<https://www.python.org/dev/peps/pep-0008/>

1.5.2 Informativi

- Presentazione capitolato C1:
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1p.pdf>
- Slide del corso di Ingegneria del Software:
 - Qualità di prodotto:
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L13.pdf>
 - Qualità di processo:
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L14.pdf>
 - Verifica e validazione:
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L16.pdf>
- THE TWELVE-FACTOR APP_G: https://12factor.net/it/#i_dodici_fattori
- Standard ISO/IEC 9126: https://it.wikipedia.org/wiki/ISO/IEC_9126
- CICLO DI DEMING_G: https://it.wikipedia.org/wiki/Ciclo_di_Deming
- Modello a V: <https://en.wikipedia.org/wiki/V-Model>
- Slide del corso Tecnologie Open Source del professor Bertazzo Nicola sul software testing:
https://elearning.unipd.it/math/pluginfile.php/43469/mod_resource/content/4/7-Testing.pdf

2 Qualità di processo

2.1 Scopo

La qualità di un prodotto è fortemente influenzata dai processi utilizzati nell'arco di creazione del prodotto stesso: da un tubo sporco non può uscire acqua pulita.

Per questo è necessario operare con un buon CICLO DI VITA_G che determina quali processi attivare e che devono essere verificati e valutati adeguatamente. A questo scopo viene seguito lo schema del Ciclo di Deming e dell'ISO 15504_G descritti all'Appendice §A.

2.2 Nomenclatura metriche e obiettivi di qualità

La nomenclatura degli obiettivi di qualità, delle METRICHE_G ed il loro funzionamento è spiegato in dettaglio nelle Norme di Progetto. In questa sezione gli obiettivi e le metriche vengono sinteticamente descritte:

- **Obiettivi:**

QPR[ID] [Nome]

- **QPR:** indica “Qualità di processo”.
- **ID:** numero progressivo di tre cifre.
- **Nome:** descrive brevemente il processo.

- **Metriche:**

MPR[ID] [Nome]

- **MPR:** indica “Metrica di processo”.
- **ID:** numero progressivo di tre cifre.
- **Nome:** descrive brevemente la metrica.

2.3 Processi

I processi saranno elencati nel seguente modo:

PROC[ID] [Nome]

- **PROC:** sta a indicare “Processo”.
- **ID:** un numero incrementale di tre cifre per indicare in modo univoco il processo.
- **Nome:** una breve frase per indicare la funzione del processo.

Per ogni processo sono elencate le sue funzioni principali, gli obiettivi che ci prefiggiamo per ottenere la qualità desiderata e le metriche adottate per raggiungere quell'obiettivo (quando previste).

2.3.1 PROC001 Pianificazione del progetto, organizzazione e struttura

Tale MACRO-PROCESSO_G ha lo scopo di pianificare il lavoro da svolgere per soddisfare i REQUISITI_G richiesti dal PROGETTO_G. È in questo processo che viene messo in atto il nostro WAY OF WORKING_G che ha un'importanza particolarmente alta perché i suoi risultati vanno a condizionare l'esito della qualità dell'intero progetto.

Funzioni

- **Sviluppare SOTTO-PROCESSI_G**: i vari obiettivi devono poter essere associati ad azioni ben precise, ognuna delle quali appartiene ad un sotto-processo.
- **Suddividere i compiti**: assegnazione dei compiti che ognuno di noi deve realizzare.
- **Calendarizzare i documenti**: stabilire delle BASELINE_G durante il progetto.
- **FORMAZIONE_G personale**: data la nostra inesperienza è richiesto un periodo di formazione personale più lungo del normale. Questo periodo deve essere contato all'interno della calendarizzazione.
- **Standard**: vengono scelti gli standard più convenienti da seguire.
- **BUDGET_G**: è necessario conoscere le proprie RISORSE_G in termini di tempo $\frac{\text{costo}}{\text{persona}}$ in modo tale da restare il più fedeli possibile al PREVENTIVO_G stilato.

Metriche

- MPR001 Varianza della PIANIFICAZIONE_G
- MPR002 Varianza dei costi
- MPR003 Aderenza agli standard
- MPR004 Frequenza COMMIT_G nella REPOSITORY_G

Obiettivi

- **QPR001 Rispetto dei periodi della pianificazione**: all'interno della presente nel *PianoDiProgetto v2.0.0_D*, sono presenti le date di scadenza delle varie attività. L'obiettivo è rispettarle il più possibile per proseguire al meglio la realizzazione del progetto.
- **QPR002 Variazione del budget**: le risorse messe a disposizione all'inizio del progetto devono potersi mantenere tali in tutta la sua durata.
- **QPR003 Rispetto delle fasi del ciclo di vita**: ogni processo deve rispettare le fasi del Ciclo di Deming.
- **QPR004 VERSIONAMENTO_G**: durante ogni processo, i prodotti che vengono realizzati devono possedere un numero di versione in modo da tener traccia delle modifiche. In questo modo è possibile sapere le cause di certi comportamenti dovuti ad un mutamento del prodotto in modo molto rapido. L'obiettivo è quindi tenere sotto controllo l'andamento dei cambiamenti.

2.3.2 PROC002 Analisi

Per processo di analisi si intende in questo caso un'attività di analisi generica, di più contenuti, non riferendosi solo all'analisi dei requisiti (consultabile al documento *AnalisiDeiRequisiti v2.0.0_D*), ma anche all'analisi dei rischi (presente nel *PianoDiProgetto v2.0.0_D*) e altre attività di identificazione e analisi riportate qui di seguito tra le funzioni.

Funzioni

- **Individuare i requisiti:** il capitolato che viene presentato dal cliente può possedere un lessico più discorsivo che tecnico. È necessario dunque effettuare un lavoro più approfondito per individuare i requisiti, espliciti ed impliciti, e classificarli.
- **Comprendere la quantità di risorse richieste:** sviluppare un preventivo è fondamentale, sia per il cliente che vuole sapere il costo del progetto, che per il FORNITORE_G che deve sapere se il progetto per lui è fattibile.
- **Individuare i rischi:** lungo il corso del progetto possono accadere degli imprevisti, perciò è necessario poterli prevedere ed organizzare fin da subito una reazione ad essi.

Metriche

- MPR005 Requisiti obbligatori non soddisfatti
- MPR006 Requisiti desiderabili non soddisfatti
- MPR007 Requisiti opzionali non soddisfatti
- MPR008 Rischi non previsti avvenuti

I risultati di tutte queste metriche possono essere pubblicate ed analizzate solo a progetto terminato, prima avrebbero poca utilità.

Obiettivi

- **QPR005 Soddisfacimento dei requisiti obbligatori:** tutti i requisiti obbligatori devono essere soddisfatti a fine progetto.
- **QPR006 Soddisfacimento dei requisiti opzionali e desiderabili:** i requisiti opzionali e desiderabili possono essere soddisfatti solo se si avanzano ancora risorse in termini di $\frac{\text{tempo}}{\text{persona}}$ prima del definitivo termine del progetto. I requisiti inoltre devono essere stabiliti in modo corretto: hanno l'obbligo di rispecchiare quanto chiesto dal cliente e non sovrapporsi tra loro.
- **QPR007 Verificarsi dei rischi previsti:** lo sviluppo del progetto dovrebbe procedere senza incertezze e non presentare imprevisti non precedentemente studiati. Questo perché potrebbe aumentare di molto il ritardo nello sviluppo del progetto rispetto al previsto.

2.3.3 PROC003 Produzione documenti

Questo processo rimane attivo in tutta la durata dello sviluppo del prodotto perché ha il compito di produrre dei documenti che riportino le scelte effettuate, gli strumenti utilizzati e le modifiche attuate nell'intera durata del progetto.

Funzioni

Il processo comprende il ciclo di vita di ogni documento, che è spiegato in dettaglio nelle Norme di Progetto, e prevede:

- Redazione
- Verifica
- Approvazione

Obiettivi

- **QPR008 Rispetto delle fasi del ciclo di vita:** il ciclo di vita di ogni documento deve essere rispettato in ogni sua fase e questo deve venir pubblicato secondo le scadenze prestabilite.

2.3.4 PROC004 Verifica

Il processo, attivo in tutta la durata del progetto, ha il compito di valutare la correttezza dei prodotti dati in input, stabilire se presentano errori e se sono sufficientemente di qualità.

Funzioni

- **Verificare le funzionalità dei prodotti:** i prodotti devono saper soddisfare i requisiti richiesti. Nella fase di verifica viene accertato che questo effettivamente avvenga.
- **Verificare la corretta esecuzione dei processi:** i processi possiedono un ciclo di vita che è suddiviso in fasi, per la buona esecuzione del processo queste devono essere eseguite nell'ordine corretto restituendo gli output attesi.
- **Verificare che siano rispettate le Norme di Progetto:** viene stabilito un nostro Way of Working che dobbiamo rispettare per tutta la durata del progetto. Anche la verifica di queste norme deve essere effettuata.

Metriche

- MPR009 Frequenza controllo prodotti

Obiettivi

- **QPR009 Effettuare una verifica costante:** la fase di verifica è sempre attiva perché ogni singolo prodotto deve essere testato e controllato ogni volta che viene modificato o prima dell'arrivo di una MILESTONE_G.
- **QPR010 Rispetto delle fasi di verifica:** la fase di verifica deve essere eseguita in iterativamente senza cambiare in modo sostanziale nel corso del progetto. Un mutamento della fase di verifica renderebbe incoerenti i risultati delle precedenti iterazioni, impedendo di vedere se è presente un miglioramento della qualità o meno.

2.4 Tabella qualità di processo

Le tabelle indicano gli obiettivi di qualità discussi nelle precedenti sezioni che ogni processo deve possedere.

Ogni obiettivo di qualità è indicato con:

- **Obiettivo:** indica il proprio codice identificativo come spiegato nella sezione §2.2.
- **Metrica:** la metrica utilizzata per valutare l'obiettivo di qualità assegnatole denominato sempre secondo quanto dichiarato in §2.2. Nel caso non fosse possibile associare una metrica ad un obiettivo di qualità, questa non verrà indicata.
- **Valore desiderato:** il valore che vogliamo ottenere per soddisfare appieno l'obiettivo di qualità. È possibile dare un valore quantificabile solo tramite una metrica, quindi questo campo è nullo nel caso in cui quest'ultima sia assente.
- **Descrizione:** descrizione generale dell'obiettivo di qualità prefisso.

PROC001 Pianificazione del progetto, organizzazione e struttura

Obiettivo	Metrica	Valore desiderato
QPR001 Rispetto dei periodi della pianificazione	MPR001 Varianza della pianificazione	96 ore
Descrizione: le scadenze orarie per ogni ruolo riportate nel Piano di Progetto considerano anche un tempo di $SLACK_G$ per eventuali ritardi. Queste sono state stabilite all'inizio del progetto, perciò molto probabilmente possono subire variazioni in corso d'opera, l'importante è che queste variazioni non siano eccessive. Considerando che ogni membro del gruppo deve svolgere un numero fisso di ore, è naturale che togliendo ore a un ruolo se ne aggiungano ad un altro o viceversa, per questo una variazione totale di 96 ore (4 giorni), non ci ne sembra eccessiva né restrigente.		
QPR002 Variazione del budget	MPR002 Varianza dei costi	0-200 €
Descrizione: ogni ruolo che copriamo possiede una tariffa oraria. Può accadere che nel corso del progetto sia richiesta una quantità di lavoro diversa da quella preventivamente conteggiata. Si è previsto di poter tollerare fino a un massimo di 200 € di differenza.		
QPR003 Rispetto delle fasi del ciclo di vita	MPR003 Aderenza agli standard	Livello di maturità: 3 Valutazione attributi: L
Descrizione: dai sondaggi è emerso che la maggior parte delle aziende italiane che si sono adeguate allo standard ISO/IEC 15504 ha raggiunto un livello di maturità pari a 3. Ci prefiggiamo quindi di raggiungere anche noi un tale livello di maturità soddisfacendo i vari attributi in almeno il 75% della loro interessezza.		
QPR004 Versionamento	MPR004 Frequenza commit nella repository	25+
Descrizione: i commit di una repository permettono di tenere una miglior traccia delle modifiche e di accedere facilmente all'ultima versione del prodotto. In base a sondaggi presenti in Internet, abbiamo constatato che un giusto numero di commit da effettuare in media durante una settimana produttiva è almeno 25.		

Tabella 1: Obiettivi di qualità per il PROC001

PROC002 Analisi

Obiettivo	Metrica	Valore desiderato
QPR005 Soddisfacimento dei requisiti obbligatori	MPR005 Requisiti obbligatori non soddisfatti	0
Descrizione: i requisiti obbligatori devono essere tutti soddisfatti entro la consegna finale del progetto.		
QPR006 Soddisfacimento dei requisiti opzionali e desiderabili	MPR006 Requisiti desiderabili non soddisfatti MPR007 Requisiti opzionali non soddisfatti	0-(n-2)
Descrizione: i requisiti opzionali e desiderabili, non essendo obbligatori possono non essere svolti all'interno del progetto. Dato che però offrono un valore aggiunto, miriamo a soddisfarne almeno due.		
QPR007 Verificarsi dei rischi previsti	MPR008 Rischi non previsti avvenuti	0
Descrizione: il verificarsi di imprevisti può accadere nel corso del progetto, ma devono poter essere rilevabili prima che succeda. Per questo si prevede che non si presenti nessun altro rischio al di fuori di quelli previsti.		

Tabella 2: Obiettivi di qualità per il PROC002

PROC003 Produzione documenti

Obiettivo	Metrica	Valore desiderato
QPR008 Rispetto delle fasi del ciclo di vita	-	-
Descrizione: ogni documento deve attraversare determinate fasi del ciclo di vita. È compito del Verificatore accertarsi che tali fasi vengano rispettate.		

Tabella 3: Obiettivi di qualità per il PROC003

PROC004 Verifica

Obiettivo	Metrica	Valore desiderato
QPR009 Effettuare una verifica costante	MPR009 Frequenza controllo prodotti	5 modifiche
Descrizione: il Verificatore deve controllare i prodotti in modo frequente, in modo tale che siano corretti nella forma, contenuto e funzionalità. Si considera 5 un numero sufficiente di modifiche apportate a un documento tale da necessitare di una verifica.		
QPR010 Rispetto delle fasi di verifica	-	-
Descrizione: è compito del Verificatore assicurarsi che la fase di verifica venga eseguita nel modo corretto per disporre sempre di risultati di verifica attendibili ed analizzabili.		

Tabella 4: Obiettivi di qualità per il PROC004

3 Qualità di prodotto

3.1 Scopo

Abbiamo scelto di prendere come modello gli standard ISO 9126 e ISO 90003:2004 individuando alcune caratteristiche a cui i prodotti dovrebbero puntare per raggiungere la qualità desiderata.

3.2 Nomenclatura metriche ed obiettivi di qualità

La nomenclatura degli obiettivi di qualità, delle metriche ed il loro funzionamento è spiegato in dettaglio nelle Norme di Progetto. In questa sezione gli obiettivi e le metriche vengono sinteticamente descritte:

- **Metriche:**

M[Prodotto] [ID] [Nome]

- **Prodotto:** può essere:
 - * **PD:** indica “Metrica del prodotto documento”.
 - * **PS:** indica “Metrica del prodotto software”.
- **ID:** numero progressivo di tre cifre.
- **Nome:** descrive brevemente la metrica.

- **Obiettivi:**

Q[Prodotto] [ID] [Nome]

- **Prodotto:** può essere:
 - * **PD:** indica “Qualità del prodotto documento”.
 - * **PS:** indica “Qualità del prodotto software”.
- **ID:** numero progressivo di tre cifre.
- **Nome:** descrive brevemente il processo.

3.3 Prodotti

Per prodotti è inteso tutto ciò che è concretamente utilizzabile, consultabile o eseguibile. Nel contesto di questo progetto i prodotti sono documenti e software.

3.3.1 Documenti

I documenti rilasciati devono presentarsi leggibili e comprensibili già ad una prima lettura, con contenuti che rispecchino le premesse del documento.

Metriche

- MPD001 INDICE DI GULPEASE_G
- MPD002 Correttezza ortografica

Obiettivi

Le caratteristiche di un documento che vengono analizzate sono:

- **QPD001 Leggibilità del testo:** i documenti devono poter essere letti in modo fluido, evitando perciò periodi troppo lunghi o un alto numero di subordinate.
- **QPD002 Correttezza ortografica:** non devono essere presenti errori ortografici.
- **QPD003 Organizzazione del documento:** i contenuti devono essere inseriti nelle sezioni e nei documenti appropriati.

3.3.2 Software

Vogliamo che software realizzato da noi sia di qualità. Per fare ciò, utilizziamo i dati che ci vengono resi disponibili da SONARQUBE_G e le norme dello standard PEP 8¹ per l'analisi statica del codice. Essi ci consentono di definire delle semplici metriche con dei grandi obiettivi da raggiungere nel corso del tempo.

Metriche

- MPS001 Presenza di bug
- MPS002 Presenza di vulnerabilità
- MPS003 Presenza di CODE SMELL_G
- MPS004 Duplicazione del codice
- MPS005 File senza intestazione
- MPS006 Righe non formattate
- MPS007 Nomi di variabili, metodi e classi non normati
- MPS008 Righe non indentate
- MPS009 Stringhe non normate
- MPS010 Righe troppo lunghe
- MPS011 Metodi troppo lunghi
- MPS012 Metodi con troppi parametri
- MPS013 Metodi con troppa complessità ciclomatica
- MPS014 Classi che implementano classi concrete
- MPS015 Commenti non normati
- MPS016 Metodi non documentati

¹Vedere in §1.5.2

Obiettivi

L'intero nostro prodotto software mira a raggiungere:

- **QPS001 Assenza di bug:** la totale assenza di bug è essenziale per la buona riuscita al rilascio del nostro software.
- **QPS002 Assenza di vulnerabilità:** miriamo a rimuovere qualunque tipo di vulnerabilità che possa compromettere la sicurezza del nostro prodotto.
- **QPS003 Assenza di code smell:** vogliamo che il nostro codice non abbia code smell, al fine di migliorarne la struttura e abbassarne la complessità, in modo da risultare facilmente leggibile.
- **QPS004 Minima duplicazione del codice:** tendiamo a raggiungere la minor quantità di codice duplicato possibile, poiché grande quantità di codice ripetuto è segno di inefficienza, ma non sempre si riesce ad ottenere del codice completamente non ridondante.
- **QPS005 Rispetto delle norme di PEP 8:** per produrre codice di qualità è necessario seguire delle norme per uniformare tutti i file e per renderli comprensibili ed efficienti. Tali norme è meglio che provengano da uno standard creato da persone con molta più esperienza di noi, per questo abbiamo scelto di seguire lo standard PEP 8.

3.4 Tabelle qualità di prodotto

Le tabelle indicano gli obiettivi di qualità che ogni prodotto deve possedere. Ogni obiettivo di qualità è indicato con:

- **Obiettivo:** viene indicato il codice dell'obiettivo di qualità secondo quanto descritto nella sezione §3.2.
- **Metrica:** la metrica utilizzata per valutare l'obiettivo di qualità assegnatole con identificativo secondo quanto scritto in §3.2. Nel caso in cui non fosse possibile associare una metrica ad un obiettivo di qualità questa non verrà indicata.
- **Valore desiderato:** il valore che si vuole ottenere attraverso la metrica indicata per soddisfare appieno l'obiettivo di qualità, qualora una metrica sia presente.
- **Descrizione:** descrizione generale dell'obiettivo di qualità.

Documenti		
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 Gulpease e una verifica da parte del Verificatore. In particolare è stato scelto l'intervallo 50-60 poiché vogliamo che i nostri documenti siano facilmente leggibili, ma anche con un lessico adeguato e più ricercato del solito.		
QPD002 Correttezza ortografica	MPD002 Correttezza ortografica	0
Descrizione: il testo non deve presentare alcun errore ortografico.		
QPD003 Organizzazione del documento	-	-
Descrizione: l'organizzazione dei documenti e la loro struttura logica vengono controllate dal Verificatore e dall'Amministratore.		

Tabella 5: Obiettivi di qualità per i documenti

Software		
Obiettivo	Metrica	Valore desiderato
QPS001 Assenza di bug	MPS001 Presenza di bug	0
Descrizione: il codice non deve presentare bug al momento del rilascio per evitare problemi nella sua esecuzione, per questo il suo numero di bug deve essere nullo al termine del progetto.		
QPS002 Assenza di vulnerabilità	MPS002 Presenza di vulnerabilità	0
Descrizione: il codice non deve possedere vulnerabilità per quanto riguarda la sicurezza o la manutenibilità entro il termine del progetto.		
QPS003 Assenza di code smell	MPS003 Presenza di code smell	0
Descrizione: per mantenere il codice leggibile, il numero di code smell deve essere nullo al termine del progetto.		
QPS004 Minima duplicazione del codice	MPS004 Duplicazione del codice	0 - 20%
Descrizione: una percentuale tollerabile di linee di codice duplicato non deve eccedere del 20% rispetto al numero totale.		
QPS005 Assenza di file senza intestazione	MPS005 File senza intestazione	0
Descrizione: per comprendere ad una prima occhiata i compiti di un file, questo deve possedere un'intestazione scritta in una certa forma prestabilita da noi membri di AlphaSix. Dunque il numero di file senza la corretta intestazione deve essere nullo al termine del progetto.		
QPS006 Assenza di righe non formattate	MPS006 Righe non formattate	0
Descrizione: nel momento in cui le righe di codice sono formattate secondo le Norme di Progetto, questo risulta essere più leggibile, perciò il numero di righe non formattate deve essere nullo al termine del progetto.		
QPS005 Rispetto delle norme di PEP 8	MPS007 Nomi di variabili, metodi e classi non normati	0
Descrizione: i nomi delle variabili, classi e metodi devono risultare il più descrittivi possibile, per questo sono presenti delle norme per la loro dichiarazione. Il numero di nomi di variabili, metodi e classi non normati deve essere nullo al termine del progetto.		
QPS005 Rispetto delle norme di PEP 8	MPS008 Righe non indentate	0
Descrizione: la lunghezza delle tabulazioni nel codice deve essere sempre uguale per favorirne la leggibilità, per questo il numero di righe non correttamente indentate deve essere nullo al termine del progetto.		

Tabella 6: Obiettivi di qualità per il software (1)

Software		
Obiettivo	Metrica	Valore desiderato
QPS005 Rispetto delle norme di PEP 8	MPS009 Stringhe non normate	0
Descrizione: per dichiarare le stringhe in Python esistono molti modi, usarne uno solo aumenta la leggibilità del codice. Il numero di stringhe non normate deve essere nullo al termine del progetto.		
QPS005 Rispetto delle norme di PEP 8	MPS010 Righe troppo lunghe	0
Descrizione: scrivere righe di una lunghezza eccessiva obbliga il lettore a scorrere in orizzontale la finestra che mostra il codice, dunque è stato normato di limitare a 79 caratteri la lunghezza di una linea di codice. Il numero di righe che eccedono questo limite deve essere nullo al termine del progetto.		
QPS005 Rispetto delle norme di PEP 8	MPS011 Metodi troppo lunghi	0
Descrizione: perché un metodo sia facilmente testabile questo deve avere il minor numero di compiti, un modo per limitarne il numero è stabilire una lunghezza massima di righe (50) dei metodi. Il numero di metodi che eccedono questo limite deve essere nullo al termine del progetto.		
QPS005 Rispetto delle norme di PEP 8	MPS012 Metodi con troppi parametri	0
Descrizione: anche un numero alto di parametri in un metodi è sintomo che questo può eseguire troppe attività; per questo il numero di parametri non dovrebbe essere superiore di 6. Il numero di metodi che eccedono questo limite deve essere nullo al termine del progetto.		
QPS005 Rispetto delle norme di PEP 8	MPS013 Metodi con troppa complessità ciclomatica	0
Descrizione: se un metodo possiede troppi cammini che può compiere una variabile al suo interno, allora vuol dire che questo è difficile da testare e, in alcuni casi, da eseguire. Il numero di metodi che ha più di 3 cicli annidati deve essere nullo al termine del progetto.		
QPS005 Rispetto delle norme di PEP 8	MPS014 Classi che implementano classi concrete	0
Descrizione: l'ereditarietà è da evitare, per impedire dipendenza tra le classi, per questo il numero di classi che implementano una o più classi concrete deve essere nullo al termine del progetto.		

Tabella 7: Obiettivi di qualità per il software (2)

Software		
Obiettivo	Metrica	Valore desiderato
QPS005 Rispetto delle norme di PEP 8	MPS015 Commenti non normati	0
Descrizione: i commenti, per essere utili devono essere coerenti col codice che commentano, sintetici e ben strutturati; nel nostro caso devono possedere uno spazio prima dell'inizio e cominciare con la lettera maiuscola.		
QPS005 Rispetto delle norme di PEP 8	MPS016 Metodi non documentati	0
Descrizione: come per i file, anche per i metodi è necessaria una documentazione per capire subito il compito di tale metodo, per questo il numero di metodi non documentati deve essere nullo al termine del progetto.		

Tabella 8: Obiettivi di qualità per il software (3)

4 Test

Tale sezione ha lo scopo di spiegare il funzionamento, lo scopo e i risultati dei vari tipi di test proposti dal modello a V e che adottiamo.

4.1 Modello a V

Il modello a V descrive in modo sintetico il ciclo di vita della realizzazione del software, partendo dalla sua progettazione fino alla sua consegna al cliente escludendo la fase di manutenzione. Lo scopo del modello a V è quello di mostrare quali tipi di test accompagnano ogni fase del ciclo e di come anche questi siano propedeutici l'uno dall'altro.

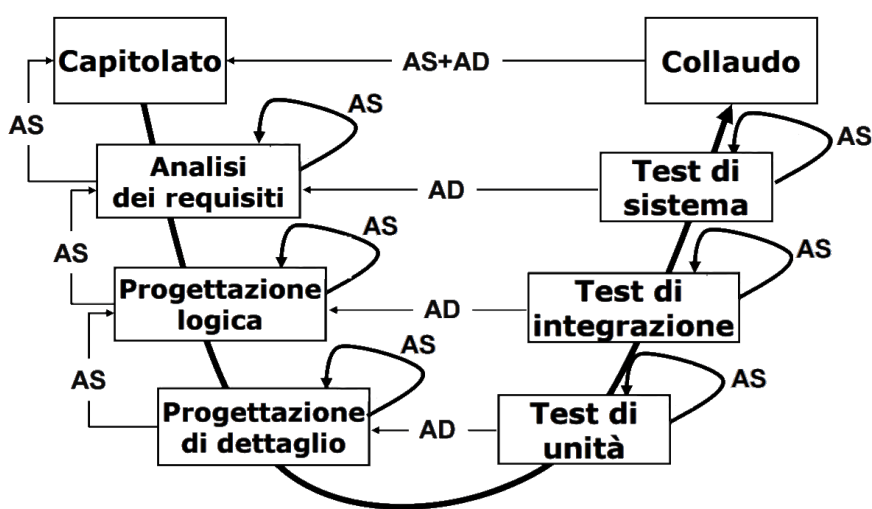


Figura 1: Modello a V²

4.2 Classificazione e struttura dei test

Ogni test viene identificato univocamente dal seguente codice:

T[Tipo] [ID]

- **T**: si riferisce a “Test”.
- **Tipo**: la tipologia a cui il test appartiene che, seguendo il modello a V, può essere:
 - **V**: validazione.
 - **S**: sistema
 - **I**: integrazione.
 - **U**: unità.
- **ID**: numero incrementale che rispetta una struttura gerarchica.

Le tabelle che raccolgono i test di una determinata tipologia presentano i campi:

- **Codice**: comprendente il codice identificativo del test.
- **Test**: descrive cosa il test deve verificare.

²Vedere modello a V in §1.5.2

- **Stato:** indica lo stato del test e può essere:
 - **NI:** non implementato.
 - **I:** implementato ma non ancora avviato.
 - **NS:** avviato e fallito.
 - **S:** avviato e superato.

4.3 Test di validazione

È un tipo di test da determinare e sviluppare in parallelo con la comprensione del capitolato. Come si può vedere dal modello a V, sono i primi tipi di test che vengono creati e saranno gli ultimi ad essere eseguiti prima della consegna del prodotto.

Codice	Test	Stato
TV1	Verifica la segnalazione dell'apertura di una issue in Redmine: <ol style="list-style-type: none">Viene aperta una issue in Redmine.Redmine invia la segnalazione.Producer Redmine riceve la segnalazione da Redmine.	NI
TV2	Verifica la segnalazione della modifica di una issue in Redmine: <ol style="list-style-type: none">Viene modificata una issue in Redmine.Redmine invia la segnalazione.Producer Redmine riceve la segnalazione da Redmine.	NI
TV3	Verifica la segnalazione dell'apertura di una issue in GitLab: <ol style="list-style-type: none">Viene aperta una issue in GitLab.GitLab invia la segnalazione.Producer GitLab riceve la segnalazione da GitLab.	NI
TV4	Verifica la segnalazione della modifica di una issue in GitLab: <ol style="list-style-type: none">Viene modificata una issue in GitLab.GitLab invia la segnalazione.Producer GitLab riceve la segnalazione da GitLab.	NI

Tabella 9: Elenco dei test di validazione (1)

Codice	Test	Stato
TV5	Verifica di un evento di push in GitLab: <ol style="list-style-type: none">Viene effettuato un push su GitLab.GitLab invia la segnalazione.Producer GitLab riceve la segnalazione da GitLab.	NI
TV6	Verifica di invio di un messaggio di apertura issue dal Producer Redmine al Gestore Personale: <ol style="list-style-type: none">Producer Redmine rielabora un messaggio di apertura issue precedentemente ricevuto.Producer Redmine invia il messaggio.Gestore personale riceve il messaggio.	NI
TV7	Verifica di invio di un messaggio di modifica issue dal Producer Redmine al Gestore Personale: <ol style="list-style-type: none">Producer Redmine rielabora un messaggio di modifica issue precedentemente ricevuto.Producer Redmine invia il messaggio.Gestore personale riceve il messaggio.	NI
TV8	Verifica di scarto di messaggi non validi nel Producer Redmine: <ol style="list-style-type: none">Producer Redmine riceve un messaggio di modifica issue non valido (non è stato modificato né il campo Subject né il campo Assignee).Producer Redmine scarta il messaggio.	NI
TV9	Verifica di invio di un messaggio di commit dal Producer GitLab al Gestore Personale: <ol style="list-style-type: none">Producer GitLab rielabora un messaggio di commit.Producer GitLab invia il messaggio.Gestore personale riceve il messaggio.	NI

Tabella 10: Elenco dei test di validazione (2)

Codice	Test	Stato
TV10	Verifica di invio di un messaggio di apertura issue dal Producer GitLab al Gestore Personale: 1. Producer GitLab rielabora un messaggio di apertura issue. 2. Producer GitLab invia il messaggio. 3. Gestore Personale riceve il messaggio.	NI
TV11	Verifica di invio di un messaggio di modifica issue dal Producer GitLab al Gestore Personale: 1. Producer GitLab rielabora un messaggio di modifica issue. 2. Producer GitLab invia il messaggio. 3. Gestore personale riceve il messaggio.	NI
TV12	Verifica di scarto di messaggi non validi nel Producer GitLab: 1. Producer GitLab riceve un messaggio di modifica issue non valido (non è stato modificato né il campo Label né il campo Title). 2. Producer GitLab scarta il messaggio.	NI
TV13	Verifica di invio di messaggi al Consumer Telegram: 1. Gestore Personale riceve un messaggio da un Producer (Redmine o GitLab). 2. Gestore Personale rielabora il messaggio ricevuto. 3. Gestore Personale invia il messaggio rielaborato. 4. Consumer Telegram riceve il messaggio.	NI
TV14	Verifica di invio di messaggi al Consumer Email: 1. Gestore Personale riceve un messaggio da un Producer (Redmine o GitLab). 2. Gestore Personale rielabora il messaggio ricevuto. 3. Gestore Personale invia il messaggio rielaborato. 4. Consumer Email riceve il messaggio.	NI

Tabella 11: Elenco dei test di validazione (3)

Codice	Test	Stato
TV15	Verifica di invio di messaggi al bot Telegram: <ol style="list-style-type: none">1. Consumer Telegram riceve un messaggio dal Gestore Personale.2. Consumer Telegram invia il messaggio.3. Bot Telegram riceve il messaggio.	NI
TV16	Verifica di invio di messaggi al server Email: <ol style="list-style-type: none">1. Consumer Email riceve un messaggio dal Gestore Personale.2. Consumer Email invia il messaggio.3. Server Email riceve il messaggio.	NI
TV17	Verifica di accesso di un utente al sistema tramite Email: <ol style="list-style-type: none">1. Inserire nel Gestore Personale l'Email dell'utente2. Confermare l'invio dei dati.3. L'accesso al sistema è avvenuto correttamente.	NI
TV18	Verifica di accesso di un utente al sistema tramite ID Telegram: <ol style="list-style-type: none">1. Inserire nel Gestore Personale l'ID Telegram dell'utente2. Confermare l'invio dei dati.3. L'accesso al sistema è avvenuto correttamente.	NI
TV19	Verifica di errore per accesso con Email inesistente: <ol style="list-style-type: none">1. Inserire nel Gestore Personale l'Email dell'utente2. Confermare l'invio dei dati.3. Viene segnalato un errore.	NI

Tabella 12: Elenco dei test di validazione (4)

Codice	Test	Stato
TV20	Verifica di errore per accesso con ID Telegram inesistente: <ol style="list-style-type: none">1. Inserire nel Gestore Personale l'ID Telegram dell'utente2. Confermare l'invio dei dati.3. Viene segnalato un errore.	NI
TV21	Verifica di uscita dell'utente dal sistema: <ol style="list-style-type: none">1. Confermare l'uscita dal sistema.2. L'utente viene riportato all'accesso al sistema.	NI
TV22	Verifica di registrazione di un nuovo utente: <ol style="list-style-type: none">1. Accedere al sistema (utente non registrato).2. Inserire i seguenti campi:<ul style="list-style-type: none">• Nome• Cognome• Contatto Email• Contatto Telegram3. Confermare l'invio dei dati.4. L'utente viene registrato correttamente.	NI
TV23	Verifica di errore per Email già esistente: <ol style="list-style-type: none">1. Accedere al sistema.2. Inserire i seguenti campi:<ul style="list-style-type: none">• Nome• Cognome• Contatto Email• Contatto Telegram3. Confermare l'invio dei dati.4. Viene segnalato un errore.	NI

Tabella 13: Elenco dei test di validazione (5)

Codice	Test	Stato
TV24	Verifica di errore per ID Telegram già esistente: <ol style="list-style-type: none">1. Accedere al sistema (utente non registrato).2. Inserire i seguenti campi:<ul style="list-style-type: none">• Nome• Cognome• Contatto Email• Contatto Telegram3. Confermare l'invio dei dati.4. Viene segnalato un errore.	NI
TV25	Verifica di rimozione di un utente dal sistema tramite Telegram: <ol style="list-style-type: none">1. Accedere al sistema.2. Inserire l'ID Telegram dell'utente da rimuovere.3. Confermare l'invio dei dati.4. L'utente è stato correttamente rimosso dal sistema.	NI
TV26	Verifica di rimozione di un utente dal sistema tramite Email: <ol style="list-style-type: none">1. Accedere al sistema.2. Inserire l'Email dell'utente da rimuovere.3. Confermare l'invio dei dati.4. L'utente è stato correttamente rimosso dal sistema.	NI
TV27	Verifica di errore nella rimozione di un utente con Email inesistente dal sistema: <ol style="list-style-type: none">1. Accedere al sistema.2. Inserire l'Email dell'utente da rimuovere.3. Confermare l'invio dei dati.4. Viene segnalato un errore.	NI

Tabella 14: Elenco dei test di validazione (6)

Codice	Test	Stato
TV28	Verifica di errore nella rimozione di un utente con ID Telegram inesistente dal sistema: 1. Accedere al sistema. 2. Inserire l'ID Telegram dell'utente da rimuovere. 3. Confermare l'invio dei dati. 4. Viene segnalato un errore.	NI
TV29	Verifica di rimozione dell'utente acceduto dal sistema tramite Email: 1. Accedere al sistema. 2. Inserire la propria Email. 3. Confermare l'invio dei dati. 4. Si esce dal sistema e si viene riportati alla schermata di accesso.	NI
TV30	Verifica di rimozione dell'utente acceduto dal sistema tramite ID Telegram: 1. Accedere al sistema. 2. Inserire la propria Email. 3. Confermare l'invio dei dati. 4. Si esce dal sistema e si viene riportati alla schermata di accesso.	NI

Tabella 15: Elenco dei test di validazione (7)

Codice	Test	Stato
TV31	Verifica di modifica di un utente tramite Email: <ol style="list-style-type: none">1. Accedere al sistema.2. Inserire i seguenti campi:<ul style="list-style-type: none">• Email dell'utente da modificare• Nome• Cognome• Contatto Email• Contatto Telegram3. Confermare l'invio dei dati.4. L'utente è stato modificato correttamente.	NI
TV32	Verifica di modifica di un utente tramite ID Telegram: <ol style="list-style-type: none">1. Accedere al sistema.2. Inserire i seguenti campi:<ul style="list-style-type: none">• Email dell'utente da modificare• Nome• Cognome• Contatto Email• Contatto Telegram3. Confermare l'invio dei dati.4. L'utente è stato modificato correttamente.	NI

Tabella 16: Elenco dei test di validazione (8)

Codice	Test	Stato
TV33	Verifica di errore nella modifica di un utente con Email già esistente tramite Email: 1. Accedere al sistema. 2. Inserire i seguenti campi: <ul style="list-style-type: none">• Email dell'utente da modificare• Nome• Cognome• Contatto Email già esistente• Contatto Telegram 3. Confermare l'invio dei dati. 4. Viene segnalato un errore.	NI
TV34	Verifica di errore nella modifica di un utente con ID Telegram già esistente tramite Email: 1. Accedere al sistema. 2. Inserire i seguenti campi: <ul style="list-style-type: none">• Email dell'utente da modificare• Nome• Cognome• Contatto Email• Contatto Telegram già esistente 3. Confermare l'invio dei dati. 4. Viene segnalato un errore.	NI

Tabella 17: Elenco dei test di validazione (9)

Codice	Test	Stato
TV35	Verifica di errore nella modifica di un utente con Email già esistente tramite ID Telegram: 1. Accedere al sistema. 2. Inserire i seguenti campi: <ul style="list-style-type: none">• ID Telegram dell'utente da modificare• Nome• Cognome• Contatto Email già esistente• Contatto Telegram 3. Confermare l'invio dei dati. 4. Viene segnalato un errore.	NI
TV36	Verifica di errore nella modifica di un utente con ID Telegram già esistente tramite ID Telegram: 1. Accedere al sistema. 2. Inserire i seguenti campi: <ul style="list-style-type: none">• ID Telegram dell'utente da modificare• Nome• Cognome• Contatto Email• Contatto Telegram già esistente 3. Confermare l'invio dei dati. 4. Viene segnalato un errore.	NI

Tabella 18: Elenco dei test di validazione (10)

Codice	Test	Stato
TV37	Verifica di errore nella modifica di un utente con Email inesistente: <ol style="list-style-type: none">1. Accedere al sistema.2. Inserire i seguenti campi:<ul style="list-style-type: none">• Email dell'utente da modificare non esistente• Nome• Cognome• Contatto Email• Contatto Telegram3. Confermare l'invio dei dati.4. Viene segnalato un errore.	NI
TV38	Verifica di errore nella modifica di un utente con ID Telegram inesistente: <ol style="list-style-type: none">1. Accedere al sistema.2. Inserire i seguenti campi:<ul style="list-style-type: none">• ID Telegram dell'utente da modificare non esistente• Nome• Cognome• Contatto Email• Contatto Telegram3. Confermare l'invio dei dati.4. Viene segnalato un errore.	NI
TV39	Verifica di iscrizione a un nuovo topic: <ol style="list-style-type: none">1. Accedere al sistema.2. Selezionare un nuovo topic a cui iscriversi.3. Confermare l'invio dei dati.4. Il topic è stato correttamente aggiunto alle preferenze dell'utente.	NI

Tabella 19: Elenco dei test di validazione (11)

Codice	Test	Stato
TV40	Verifica di aggiunta dei giorni di indisponibilità al calendario: <ol style="list-style-type: none">1. Accedere al sistema.2. Selezionare i giorni di indisponibilità da aggiungere.3. Confermare l'invio dei dati.4. I giorni sono stati aggiunti correttamente alle preferenze dell'utente.	NI
TV41	Verifica di aggiunta della piattaforma di messaggistica preferita: <ol style="list-style-type: none">1. Accedere al sistema.2. Selezionare la piattaforma preferita per la ricezione dei messaggi da aggiungere.3. Confermare l'invio dei dati.4. La piattaforma è stata correttamente aggiunta alle preferenze dell'utente.	NI
TV42	Verifica di aggiunta della persona di fiducia: <ol style="list-style-type: none">1. Accedere al sistema.2. Selezionare la persona di fiducia da aggiungere.3. Confermare l'invio dei dati.4. La persona di fiducia è stata correttamente aggiunta alle preferenze dell'utente.	NI
TV43	Verifica di errore nell'aggiunta di una persona di fiducia inesistente: <ol style="list-style-type: none">1. Accedere al sistema.2. Inserire la persona di fiducia da aggiungere (non esistente).3. Confermare l'invio dei dati.4. Viene segnalato un errore.	NI

Tabella 20: Elenco dei test di validazione (12)

Codice	Test	Stato
TV44	Verifica di aggiunta delle keyword per i push di GitLab: <ol style="list-style-type: none">1. Accedere al sistema.2. Aggiungere le keyword GitLab.3. Confermare l'invio dei dati.4. Le keyword di GitLab sono state aggiornate correttamente.	NI
TV45	Verifica di errore nell'aggiunta delle keyword per i push di GitLab già esistenti: <ol style="list-style-type: none">1. Accedere al sistema.2. Aggiungere delle keyword GitLab già presenti.3. Confermare l'invio dei dati.4. Viene segnalato un errore.	NI
TV46	Verifica di rimozione di un topic: <ol style="list-style-type: none">1. Accedere al sistema.2. Selezionare un topic da rimuovere.3. Confermare l'invio dei dati.4. Il topic è stato rimosso correttamente dalle preferenze.	NI
TV47	Verifica di rimozione dei giorni di indisponibilità dal calendario: <ol style="list-style-type: none">1. Accedere al sistema.2. Selezionare i giorni di indisponibilità da rimuovere.3. Confermare l'invio dei dati.4. I giorni sono stati rimossi dalle preferenze correttamente.	NI

Tabella 21: Elenco dei test di validazione (13)

Codice	Test	Stato
TV48	Verifica di rimozione della piattaforma di messaggistica preferita: <ol style="list-style-type: none">1. Accedere al sistema.2. Selezionare la piattaforma preferita per la ricezione dei messaggi da rimuovere.3. Confermare l'invio dei dati.4. La piattaforma è stata correttamente rimossa dalle preferenze.	NI
TV49	Verifica di rimozione della persona di fiducia: <ol style="list-style-type: none">1. Accedere al sistema.2. Selezionare la persona di fiducia da rimuovere.3. Confermare l'invio dei dati.4. La persona di fiducia è stata correttamente rimossa alle preferenze.	NI
TV50	Verifica di errore nella rimozione di una persona di fiducia inesistente: <ol style="list-style-type: none">1. Accedere al sistema.2. Selezionare la persona di fiducia da rimuovere (non esistente).3. Confermare l'invio dei dati.4. Viene segnalato un errore.	NI
TV51	Verifica di rimozione delle keyword per i push di GitLab: <ol style="list-style-type: none">1. Accedere al sistema.2. Aggiungere le keyword GitLab.3. Confermare l'invio dei dati.4. Le keyword GitLab sono state aggiornate correttamente.	NI

Tabella 22: Elenco dei test di validazione (14)

Codice	Test	Stato
TV52	Verifica di errore nella rimozione delle keyword inesistenti per i push di GitLab: 1. Accedere al sistema. 2. Aggiungere delle keyword GitLab non presenti. 3. Confermare l'invio dei dati. 4. Viene segnalato un errore.	NI

Tabella 23: Elenco dei test di validazione (15)

4.3.1 Tracciamento

Codice	Requisito
TV1	R1F2
TV2	R2F2
TV3	R3F2
TV4	R4F2
TV5	R5F2
TV6	R6.1F2
TV7	R6.2F2
TV8	R7F2
TV9	R8.1F2
TV10	R8.2.1F2
TV11	R8.2.2F2
TV12	R9F2
TV13	R10F2
TV14	R11F2
TV15	R12F2
TV16	R13F2
TV17	R14.1F2
TV18	R14.1F2
TV19	R15F2
TV20	R15F2
TV21	R16F2
TV22	R17F2
TV23	R17.3.1F2
TV24	R17.4.1F2
TV25	R18.2F2
TV26	R18.1F2
TV27	R18.3F2
TV28	R18.3F2
TV29	R18.4F2
TV30	R18.4F2
TV31	R19.1F2

Tabella 24: Elenco dei test in correlazioni con i requisiti (1)

Codice	Requisito
TV32	R19.1F2
TV33	R19.4.1F2
TV34	R19.5.1F2
TV35	R19.4.1F2
TV36	R19.5.1F2
TV37	R19.1.1F2
TV38	R19.1.1F2
TV39	R20.1F2
TV40	R20.2F2
TV41	R20.3F2
TV42	R20.4F2
TV43	R20.4.1F2, R20.4.2F2
TV44	R20.5F2
TV45	R20.5.1F2
TV46	R21.1F2
TV47	R21.2F2
TV48	R21.3F2
TV49	R21.4F2
TV50	R21.4.1F2, R21.4.2F2
TV51	R21.5F2
TV52	R21.5.1F2

Tabella 25: Elenco dei test in correlazioni con i requisiti (2)

4.4 Test di sistema

Tipo di test da scegliere durante l'attività di analisi dei requisiti. Servono per svolgere la validazione del prodotto.

I test di sistema servono per testare l'intero prodotto, in particolare le sue specifiche tecniche e funzionali. Per questo si creano dei test derivanti dai casi d'uso e dai requisiti funzionali presenti nell'*AnalisiDeiRequisiti v2.0.0_D*.

Codice	Test	Stato
TS1	Verifica che il sistema riceva la segnalazione dell'apertura di una issue da parte di Redmine	NI
TS2	Verifica che il sistema riceva la segnalazione della modifica di una issue da parte di Redmine	NI
TS3	Verifica che il sistema riceva la segnalazione dell'apertura di una issue da parte di GitLab	NI
TS4	Verifica che il sistema riceva la segnalazione della modifica di una issue da parte di GitLab	NI
TS5	Verifica che il sistema riceva eventi di push da parte di GitLab	NI
TS6	Verifica di invio di un messaggio di apertura issue dal Producer Redmine al Gestore Personale	NI
TS7	Verifica di invio di un messaggio di modifica issue dal Producer Redmine al Gestore Personale	NI
TS8	Verifica di scarto di messaggi non validi nel Producer Redmine	NI
TS9	Verifica di invio di un messaggio di commit dal Producer GitLab al Gestore Personale	NI
TS10	Verifica di invio di un messaggio di apertura issue dal Producer GitLab al Gestore Personale	NI
TS11	Verifica di invio di un messaggio di modifica issue dal Producer GitLab al Gestore Personale	NI
TS12	Verifica di scarto di messaggi non validi nel Producer GitLab	NI
TS13	Verifica di invio di messaggi al Consumer Telegram	NI
TS14	Verifica di invio di messaggi al Consumer Email	NI
TS15	Verifica di invio di messaggi al bot Telegram	NI

Tabella 26: Elenco dei test di sistema (1)

Codice	Test	Stato
TS16	Verifica di invio di messaggi al server Email	NI
TS17	Verifica di accesso di un utente al sistema tramite Email	NI
TS18	Verifica di accesso di un utente al sistema tramite ID Telegram	NI
TS19	Verifica di errore per accesso con Email inesistente	NI
TS20	Verifica di errore per accesso con ID Telegram inesistente	NI
TS21	Verifica di uscita dell'utente dal sistema	NI
TS22	Verifica di registrazione di un nuovo utente	NI
TS23	Verifica di errore per Email già esistente	NI
TS24	Verifica di errore per ID Telegram già esistente	NI
TS25	Verifica di rimozione utente dal sistema tramite Telegram	NI
TS26	Verifica di rimozione utente dal sistema tramite Email	NI
TS27	Verifica di errore nella rimozione di un utente con Email inesistente dal sistema	NI
TS28	Verifica di errore nella rimozione di un utente con ID Telegram inesistente dal sistema	NI
TS29	Verifica di rimozione dell'utente acceduto dal sistema tramite Email	NI
TS30	Verifica di rimozione dell'utente acceduto dal sistema tramite ID Telegram	NI
TS31	Verifica di modifica di un utente tramite Email	NI
TS32	Verifica di modifica di un utente tramite ID Telegram	NI
TS33	Verifica di errore nella modifica di un utente con Email già esistente tramite Email	NI
TS34	Verifica di errore nella modifica di un utente con ID Telegram già esistente tramite Email	NI
TS35	Verifica di errore nella modifica di un utente con Email già esistente tramite ID Telegram	NI
TS36	Verifica di errore nella modifica di un utente con ID Telegram già esistente tramite ID Telegram	NI
TS37	Verifica di errore nella modifica di un utente con Email inesistente	NI
TS38	Verifica di errore nella modifica di un utente con ID Telegram inesistente	NI

Tabella 27: Elenco dei test di sistema (2)

Codice	Test	Stato
TS39	Verifica di iscrizione a un nuovo topic	NI
TS40	Verifica di aggiunta dei giorni di indisponibilità al calendario	NI
TS41	Verifica di aggiunta della piattaforma di messaggistica preferita	NI
TS42	Verifica di aggiunta della persona di fiducia	NI
TS43	Verifica di errore nell'aggiunta di una persona di fiducia inesistente	NI
TS44	Verifica di aggiunta delle keyword per i push di GitLab	NI
TS45	Verifica di errore nell'aggiunta delle keyword per i push di GitLab già esistenti	NI
TS46	Verifica di rimozione di un topic	NI
TS47	Verifica di rimozione dei giorni di indisponibilità dal calendario	NI
TS48	Verifica di rimozione della piattaforma di messaggistica preferita	NI
TS49	Verifica di rimozione della persona di fiducia	NI
TS50	Verifica di errore nella rimozione di una persona di fiducia inesistente	NI
TS51	Verifica di rimozione delle keyword per i push di GitLab	NI
TS52	Verifica di errore nella rimozione delle keyword inesistenti per i push di GitLab	NI

Tabella 28: Elenco dei test di sistema (3)

4.4.1 Tracciamento

Codice	Requisito
TS1	R1F2
TS2	R2F2
TS3	R3F2
TS4	R4F2
TS5	R5F2
TS6	R6.1F2
TS7	R6.2F2
TS8	R7F2
TS9	R8.1F2
TS10	R8.2.1F2
TS11	R8.2.2F2
TS12	R9F2
TS13	R10F2
TS14	R11F2
TS15	R12F2
TS16	R13F2
TS17	R14.1F2
TS18	R14.1F2
TS19	R15F2
TS20	R15F2
TS21	R16F2
TS22	R17F2
TS23	R17.3.1F2
TS24	R17.4.1F2
TS25	R18.2F2
TS26	R18.1F2
TS27	R18.3F2
TS28	R18.3F2
TS29	R18.4F2
TS30	R18.4F2
TS31	R19.1F2

Tabella 29: Elenco dei test in correlazioni con i requisiti (1)

Codice	Requisito
TS32	R19.1F2
TS33	R19.4.1F2
TS34	R19.5.1F2
TS35	R19.4.1F2
TS36	R19.5.1F2
TS37	R19.1.1F2
TS38	R19.1.1F2
TS39	R20.1F2
TS40	R20.2F2
TS41	R20.3F2
TS42	R20.4F2
TS43	R20.4.1F2, R20.4.2F2
TS44	R20.5F2
TS45	R20.5.1F2
TS46	R21.1F2
TS47	R21.2F2
TS48	R21.3F2
TS49	R21.4F2
TS50	R21.4.1F2, R21.4.2F2
TS51	R21.5F2
TS52	R21.5.1F2

Tabella 30: Elenco dei test in correlazioni con i requisiti (2)

4.5 Test d'integrazione

Codice	Test	Stato
TI1	Verifica l'integrazione tra Webhook di GitLab e Producer GitLab	NI
TI2	Verifica l'integrazione tra Webhook di Redmine e Producer Redmine	NI
TI3	Verifica l'integrazione del Cosumer Telegram con il bot Telegram	NI
TI4	Verifica l'integrazione del Consumer Email con il server Email	NI
TI5	Verifica l'integrazione del database MongoDB con l'interfaccia del Gestore Personale	NI

Tabella 31: Elenco dei test d'integrazione

4.5.1 Tracciamento

Codice	Componente
TI1	Producer GitLab
TI2	Producer Redmine
TI3	Consumer Telegram
TI4	Consumer Email
TI5	Interfaccia Gestore Personale

Tabella 32: Elenco dei test in correlazioni con le componenti

A Standard di qualità

Prendiamo come riferimento per gli obiettivi di qualità del prodotto determinati standard. In questa appendice descriviamo gli standard adottati nelle loro parti più rilevanti ai fini del progetto. Nelle *NormeDiProgetto v2.0.0_D* viene indicato in che misura questi standard saranno applicati e nel *PianoDiQualifica v2.0.0_D* il piano che abbiamo scelto per rispettarli.

A.1 ISO/IEC 15504 (SPICE)

Lo standard ISO/IEC 15504 è stato creato per unire in un unico standard le caratteristiche principali di CMMI_G e SPY_G; entrambi standard riguardanti la qualità di processi software. ISO/IEC 15504 è chiamato anche SPICE come acronimo di *Software Process Improvement and Capability Determination*, dando importanza al termine “Capability” inteso come la capacità di un processo di essere cognitivamente capace di raggiungere il suo scopo.

Un processo con un’alta Capability è osservato da tutti in modo disciplinato e sistematico. In caso di bassa Capability il processo viene effettuato in modo opportunistico e disorganizzato.

SPICE mette a disposizione una metrica per valutare diversi attributi per ogni processo ed assegna un valore quantificabile ad ognuno di questi in modo tale da esplicitare come poter migliorare tale processo. Ogni valutazione in questo modo può essere oggettiva, ripetibile e comparabile.

I processi vengono classificati in:

- Cliente/Fornitore
- Ingegneria
- Supporto
- Gestione
- Organizzazione

I livelli sono:

- **0 Incompleto:** il processo è caotico perché con risultati e performance incomplete.
- **1 Performato:** il processo inizia ad essere eseguito mettendo a disposizione degli input ed output.

Attributi:

- **Esecuzione dei processi:** indica il numero di obiettivi raggiunti.

- **2 Gestito:** le responsabilità e la gestione del processo sono definite.

Attributi:

- **Gestione del processo:** indica quanto sono organizzati gli obiettivi fissati.
- **Gestione del prodotto:** indica quanto sono organizzati o gestiti i prodotti rilasciati.

- **3 Stabilito:** il processo è pronto per diventare un processo standard ed essere rilasciato.

Attributi:

- **Definizione del processo:** indica quanto il processo aderisce agli standard.

- **Distribuzione del processo:** indica in che misura il processo possa essere rilasciato potendo restituire sempre lo stesso risultato.
- **4 Prevedibile:** il processo è in grado di essere sottoposto a metriche e valutazioni quantitative. Spesso i risultati sono predicibili.

Attributi:

- **Misurazioni del processo:** indica quanto le metriche possono essere applicate al processo.
- **Controllo del processo:** indica quanto i risultati delle valutazioni siano predicibili.
- **5 Ottimizzante:** il processo attua miglioramenti qualitativi e quantitativi.

Attributi:

- **Innovazione del processo:** indica quanto i cambiamenti attuati nel processo risultino innovativi e positivi grazie ad una fase di analisi.
- **Ottimizzazione del processo:** indica quanto la curva di miglioramento del processo sia lineare.

Ad ogni attributo viene data una valutazione assegnata in base alla percentuale di soddisfacimento dell'attributo:

- **N:** il processo non è implementato e non svolge niente di significativo (0% - 15%).
- **P:** il processo è parzialmente implementato (15% - 50%).
- **L:** il processo è largamente implementato (50% - 85%).
- **F:** il processo è completamente implementato (85% - 100%).

Attributi	Valutazioni			
	N	P	L	F
Esecuzione dei processi	[0 - 1]			
Gestione del processo Gestione del prodotto	[1 - 2]			
Definizione del processo Distribuzione del processo	[2 - 3]			
Misurazione del processo Controllo del processo	[3 - 4]			
Innovazione del processo Ottimizzazione del processo	[4 - 5]			

Tabella 33: Schema degli attributi di ISO/IEC 15504

A.2 ISO/IEC 9126:2001

ISO/IEC 9126 è uno standard inerente alla qualità del software. Esso è strutturato in modo tale che si possa migliorare l'insieme dei processi.

La sua struttura prevede tre tipi di qualità, ognuna delle quali possiede determinate caratteristiche:

- **Qualità interna:** misura la qualità del prodotto non in esecuzione, in questo caso parliamo del codice sorgente a cui si possono aggiungere diverse metriche attraverso l'analisi statica. Queste poi possono stabilire la manutenibilità e portabilità del prodotto.

Gli attributi ad essa assegnati sono:

- Manutenibilità
- Portabilità

- **Qualità esterna:** misura attraverso l'analisi dinamica quanto l'esecuzione del prodotto rispetti gli obiettivi prefissati.

Gli attributi ad essa assegnati sono:

- Funzionalità
- Efficienza
- Affidabilità
- Usabilità

- **Qualità in uso:** definisce le metriche del prodotto rilasciato ed usato dal cliente che ne misurerà la qualità.

Gli attributi ad essa assegnati sono:

- Efficacia
- Produttività
- Soddisfazione
- Safety

A.2.1 Descrizione degli attributi della qualità interna e della qualità esterna

- **Manutenibilità:** il software nel corso delle sue revisioni deve essere facilmente modificabile.

Nello specifico si prevede:

- **Analizzabilità:** prevedere una lettura del codice fruibile.
- **Modificabilità:** poter capire subito dove applicare la modifica.
- **Stabilità:** evitare effetti indesiderati dopo le modifiche.
- **Testabilità:** poter creare facilmente dei test su tutto il codice.

- **Portabilità:** il software dovrebbe poter essere eseguito in più ambienti.

Nello specifico si prevede:

- **Adattabilità:** potersi adattare automaticamente ai vari ambienti.
- **Installabilità:** la sua fase d'installazione dovrebbe essere semplice.
- **Conformità:** il software deve sapere coesistere con le altre applicazioni.
- **Sostituibilità:** essere capace di sostituire un software con gli stessi scopi.

- **Funzionalità:** il software deve mettere a disposizione le funzionalità richieste in rapporto all'ambiente d'esecuzione.

Nello specifico si prevede:

- **Appropriatezza:** le funzionalità del software sono appropriate ai requisiti richiesti.
 - **Accuratezza:** in che misura le funzionalità aderiscono ai requisiti richiesti.
 - **Interoperabilità:** la capacità di interagire coi sistemi specificati.
 - **Security:** proteggere le informazioni da agenti esterni.
- **Efficienza:** misura della capacità di raggiungere gli obiettivi stabiliti cercando di usare meno risorse possibili, in particolare:
 - **Nel tempo:** poter dare risposte in un tempo di scadenza appropriato.
 - **Nello spazio:** utilizzo del minor numero di risorse.

- **Affidabilità:** il software deve mantenere le specifiche richieste senza inconvenienti.

Nello specifico si prevede:

- **Maturità:** indica il livello minimo della qualità delle parti che determina poi il livello di qualità dell'intero prodotto.
 - **Robustezza:** la capacità di saper reagire agli errori.
 - **Recuperabilità:** per poter tornare alla versione precedente del software in modo semplice.
- **Usabilità:** il software deve poter essere compreso fin da subito, dunque semplice ed immediato nell'utilizzo e nella comprensione.

Nello specifico si prevede:

- **Comprensibilità:** il significato delle funzionalità deve essere compreso il prima possibile dall'utente.
- **Apprendibilità:** capacità dell'utente di imparare ad utilizzare le funzionalità disponibili.
- **Operabilità:** capacità dell'utente di usare e controllare il software.
- **Attrattiva:** il risultato deve risultare attraente all'utente.

A.2.2 Descrizione degli attributi della qualità in uso

- **Efficacia:** misura della capacità di riuscire a raggiungere i compiti fissati. Essa si calcola in base al grado di raggiungimento degli obiettivi.
- **Produttività:** intesa come $\frac{\text{unità di prodotto realizzato}}{\text{unità di risorse utilizzate}}$.
- **Soddisfazione:** il software deve soddisfare l'utente.
- **Safety:** il software deve possedere adeguati livelli di sicurezza per il tipo di utente che ne usufruisce.

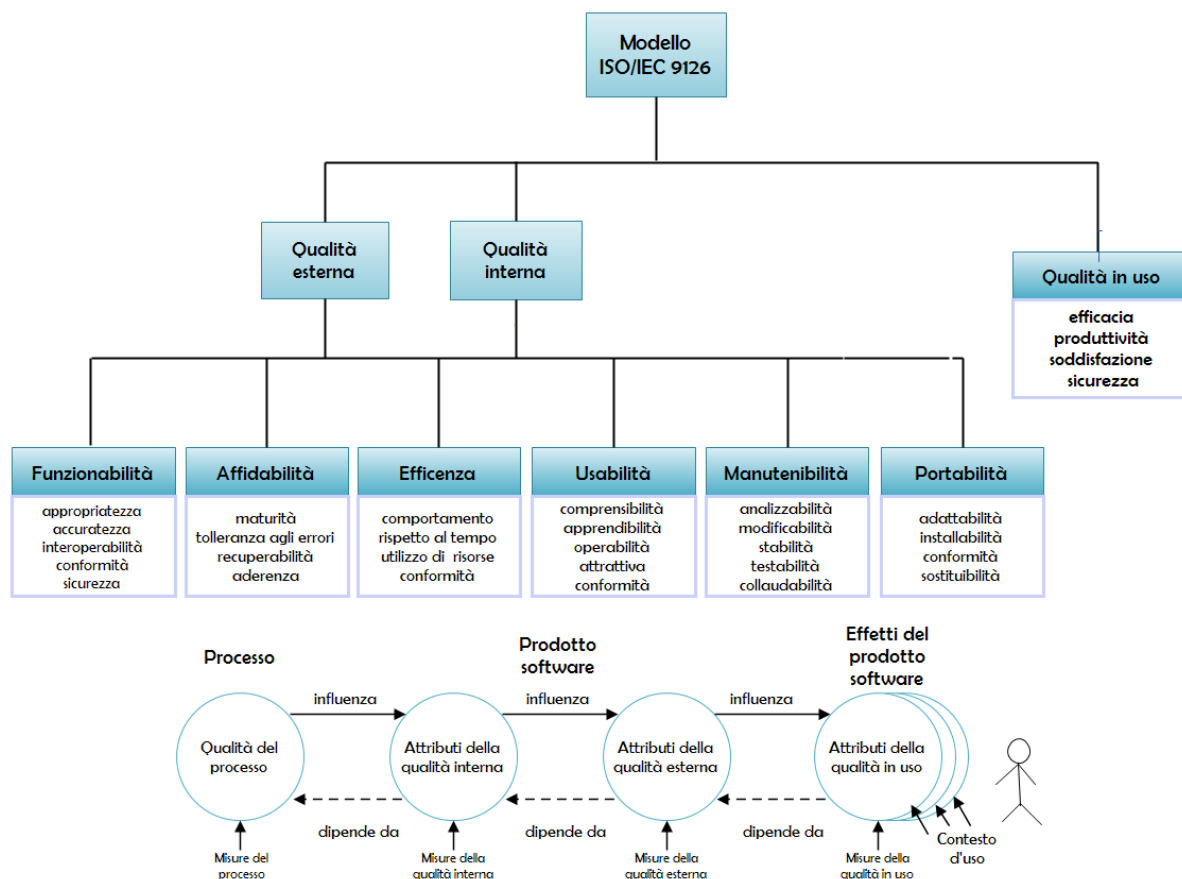


Figura 2: Schema ISO 9126 con il suo ciclo di vita³

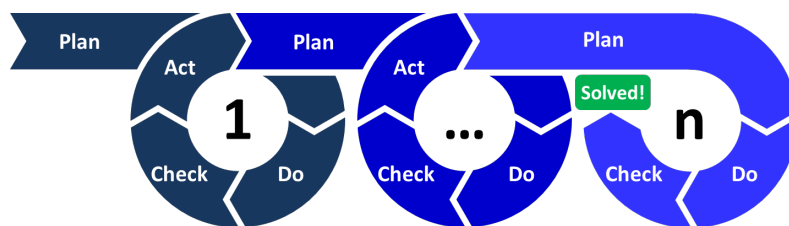
A.3 Ciclo di Deming

Il Ciclo di Deming è un metodo iterativo creato per migliorare la qualità dei processi e dei prodotti software. Questo opera nell'ottica di un miglioramento continuo in termini di risultati e risorse utilizzate e al termine di ogni ciclo, l'eventuale miglioramento effettuato diventa la nuova base da cui inizia l'iterazione successiva.

Si distingue in quattro fasi, chiamate anche PDCA, che sono:

- **Plan**: pianificare i processi per ottenere i risultati attesi ed osservare dove questi possono essere migliorati.
- **Do**: eseguire il programma e i miglioramenti inseriti nella fase precedente.
- **Check**: effettuare test e controlli dell'output della fase precedente confrontandoli con gli obiettivi della fase di Plan. Dare una valutazione dei risultati per verificare se i cambiamenti attuati portano effettivamente dei miglioramenti.
- **Act**: le modifiche, se risultano essere delle migliorie vengono applicate al processo o al prodotto.

³Vedere in §1.5.2

Figura 3: Ciclo di Deming⁴

A.4 ISO/IEC 90003:2004

Lo standard ISO/IEC 90003 consiste nello standard ISO 9001:2008 applicato al software. Quest'ultimo tratta di uno standard sulla qualità dei sistemi aziendali che segue il sistema il Ciclo di Deming descritto in §A.3.

L'ISO/IEC 90003:2004 si suddivide in otto capitoli che sono:

1. Scopo
2. Riferimenti normativi
3. Termini e definizioni
4. Sistema di gestione della qualità
5. Gestione delle responsabilità
6. Gestione delle risorse
7. Realizzazione del prodotto
8. Misurazione, analisi e miglioramento

Date le nostre capacità e la nostra esperienza, prendiamo in considerazione solo alcuni punti del quarto capitolo. In particolar modo quelli riguardanti la documentazione, non potendo ancora stabilire metriche sul software.

A.4.1 Capitolo 4. dell'ISO 90003: requisiti di sistema e linee guida

1. Requisiti e linee guida sull'organizzazione:

(a) Stabilire la gestione del sistema di qualità (QMS)

- **Identificare i requisiti di cui il QMS ha bisogno:** attraverso il Piano di Qualifica osservando i processi interni ed esterni.
- **Assicurarsi che ogni processo sia efficace:** attraverso una verifica continua su processi e prodotti.

(b) Documentare il proprio QMS: riportare le relazioni tra i vari processi e come questi vengono gestiti e verificati.

(c) Cercare di migliorare il proprio QMS: applicando il Ciclo di Deming.

(d) Mantenere la qualità del QMS

⁴Vedere Riferimenti Informativi in §1.5.2

2. Requisiti e linee guida sulla documentazione:

(a) Documentare la qualità:

- **Pianificare la documentazione del QMS:** assicurandosi che quello che verrà scritto rispecchi gli obiettivi e la gestione dei processi.
- **Stabilire la documentazione del QMS:** stabilendo la qualità delle diverse parti del QMS.
- Mantenere la documentazione nel tempo

(b) Controllare la qualità dei documenti:

- **Stabilire una procedura per controllare i documenti:** documentare la stessa procedura di controllo che dovrebbe contenere la fase di approvazione del documento, un suo versionamento e continuo aggiornamento.
- Mantenere ed aggiornare la procedura di controllo del documento

B Resoconto delle attività di verifica

In questa sezione viene riportata una sintesi conclusiva dei risultati ottenuti dalle fasi di verifica effettuate nei vari periodi del progetto attraverso le metriche indicate in tale documento. Essi possono coincidere o meno con i valori desiderati da AlphaSix e, nel secondo caso in particolar modo, saranno oggetto di valutazioni per il miglioramento descritte all'appendice §C.

I test che vengono introdotti nel corso del progetto non vanno a sostituire i test precedentemente sviluppati. Ogni test, appena è creato viene eseguito periodicamente fino alla fine del progetto, per evitare che i nuovi cambiamenti possano reintrodurre errori precedentemente risolti.

B.1 Classificazione dei risultati

I risultati ottenuti tramite una metrica, rispetto all'obiettivo che vogliamo raggiungere e ai valori da noi desiderati, sono classificati in:

- **Soddisfacente:** il risultato è quello atteso.
- **Poco soddisfacente:** il risultato non è quello atteso, ma gli è vicino.
- **Insoddisfacente:** il risultato non è per niente quello atteso.

B.2 Primo periodo (RR)

Periodo individuato come Analisi dei Requisiti che dura circa un mese di tempo e un quarto del tempo complessivo per realizzare il progetto.

B.2.1 Riassunto delle attività di verifica

L'attività di verifica si è rivelata per noi più faticosa del previsto. I motivi sono molteplici:

- Nel primo periodo perché non avevamo sufficiente esperienza per effettuare una verifica sistematica e perché la nostra attenzione si è interamente concentrata sull'organizzazione dei ruoli e la loro funzione
- Nel secondo periodo invece la verifica è stata più sistematica e meno impegnativa perché gli elaborati venivano consegnati in orario, ma comunque la correzione si è rivelata onerosa

Come indicato nelle *NormeDiProgetto v2.0.0_D*, l'attività di verifica è stata effettuata inizialmente attraverso Walkthrough e successivamente attraverso Inspection. Essendo ancora alle prime armi all'inizio del progetto, abbiamo effettuato la verifica secondo Walkthrough per una buona parte del periodo di analisi dei requisiti realizzata insieme. Dopodiché siamo passati ad adottare Inspection in quanto una ricerca dispersiva non era utile alla parallelizzazione dei compiti. Nel momento in cui sarà necessario verificare prodotti software, riteniamo opportuno effettuare il passaggio da Walkthrough a Inspection in tempi più rapidi rispetto a come è avvenuto finora.

B.2.2 Risultati delle verifiche tramite analisi

Nelle tabelle seguenti vengono riportati i risultati ottenuti applicando le metriche in correlazione all'obiettivo scelto. In ogni tabella sono indicati i prodotti o i processi sottoposti alle metriche e i risultati ottenuti sono presenti in "Valutazione", classificati secondo §B.1.

B.2.2.1 Documenti

QPD001 Leggibilità del testo	MPD001 Indice Gulpease	50 - 60
Prodotto/processo testato	Risultato ottenuto	Valutazione
<i>NormeDiProgetto v2.0.0_D</i>	54.57	Soddisfacente
<i>StudioDiFattibilità v1.0.0_D</i>	53.7	Soddisfacente
<i>PianoDiProgetto v2.0.0_D</i>	51.75	Soddisfacente
<i>PianoDiQualifica v2.0.0_D</i>	53.29	Soddisfacente
<i>AnalisiDeiRequisiti v2.0.0_D</i>	55.62	Soddisfacente

Nota: tutti i documenti soddisfano pienamente l'obiettivo di qualifica indicato.

Tabella 34: Risultati di MPD001 Indice Gulpease

QPD002 Correttezza ortografica	MPD002 Correttezza ortografica	0
Prodotto/processo testato	Risultato ottenuto	Valutazione
<i>NormeDiProgetto v2.0.0_D</i>	0	Soddisfacente
<i>StudioDiFattibilità v1.0.0_D</i>	0	Soddisfacente
<i>PianoDiProgetto v2.0.0_D</i>	0	Soddisfacente
<i>PianoDiQualifica v2.0.0_D</i>	0	Soddisfacente
<i>AnalisiDeiRequisiti v2.0.0_D</i>	0	Soddisfacente

Nota: tutti i documenti soddisfano pienamente l'obiettivo di qualifica indicato.

Tabella 35: Risultati di MPD002 Correttezza ortografica

B.2.2.2 Processi

QPR001 Rispetto dei periodi della pianificazione	MPR001 Varianza della pianificazione	96 ore
Prodotto/processo testato	Risultato ottenuto	Valutazione
Piano di Progetto	7	Soddisfacente

Nota: non tutte le scadenze sono state rispettate, ma i ritardi sono rientrati nei valori tollerati. Il valore ottenuto è la somma delle ore di variazione dei vari ruoli.

Tabella 36: Risultati di MPR001 Varianza della pianificazione

QPR002 Variazione del budget	MPR002 Varianza dei costi	0 - 200 €
Prodotto/processo testato	Risultato ottenuto	Valutazione
Differenza consuntivo rimanente e consuntivo di periodo	- 15,00 €	Soddisfacente

Nota: il valore desiderato indicato precedentemente sembra essere fin troppo permissivo. Il dettaglio del risultato ottenuto è indicato nel consuntivo di periodo.

Tabella 37: Risultati di MPR002 Varianza dei costi

QPR003 Rispetto delle fasi del ciclo di vita	MPR003 Aderenza agli standard	Livello di maturità: 3 Valutazione attributi: L
Prodotto/processo testato	Risultato ottenuto	Valutazione
PROC001 Pianificazione del progetto, organizzazione e struttura	Livello di maturità: 2 Valutazione attributi: L	Poco soddisfacente

Nota: essendo il primo periodo è prevedibile che il livello di maturità desiderato non sia ancora quello sperato. Si prevede un miglioramento per le successive revisioni.

Tabella 38: Risultati di MPR003 Aderenza agli standard

QPR004 Versionamento	MPR004 Frequenza commit nella repository	25
Prodotto/processo testato	Risultato ottenuto	Valutazione
Repository	31.25	Soddisfacente

Nota: il numero di commit effettuati risulta ottimo.

Tabella 39: Risultati di MPR004 Frequenza commit nella repository

QPR009 Effettuare una verifica costante	MPR009 Frequenza controllo prodotti	5 Modifiche
Prodotto/processo testato	Risultato ottenuto	Valutazione
Documenti	7.2	Poco soddisfacente

Nota: il valore ottenuto non soddisfa il risultato atteso perché nella prima parte del macro periodo AlphaSix ha prestato molta attenzione alla formazione personale a discapito del tempo che poteva essere dedicato alla verifica dei prodotti.

Tabella 40: Risultati di MPR009 Frequenza controllo prodotti

B.3 Secondo periodo (RP)

Periodo individuato come Revisione Progettuale la quale dura all'incirca un mese ed un quarto dell'intera durata del progetto.

B.3.1 Riassunto delle attività di verifica

In questo periodo di tempo abbiamo per lo più attuato verifiche sui documenti, come per il periodo di Revisione dei Requisiti, ma in maniera più approfondita e sistematica, e verifiche sul software, in particolare sul codice scritto per il PROOF OF CONCEPT_G.

B.3.2 Risultati delle verifiche tramite analisi

In questa sezione riportiamo i diagrammi contenenti tutti i valori di nostro interesse per dare un giudizio sul lavoro svolto fino alla fine di questo periodo. Essi contengono i valori ottenuti nel tempo partendo dall'ultima consegna dei documenti (il 2019-01-14) e mostrano con chiarezza l'andamento delle misurazioni che abbiamo effettuato. Per questo motivo, ogni diagramma è intitolato con la metrica utilizzata ed è accompagnato da:

- **Obiettivo:** codice dell'obiettivo di qualità.
- **Valore desiderato:** il valore desiderato ottenibile da una determinata metrica.
- **Descrizione:** breve descrizione del diagramma.
- **Valutazione:** la nostra valutazione classificata secondo §B.1.
- **Considerazioni:** nostro commento in relazione ai risultati ottenuti dalla misurazione.

B.3.2.1 Documenti

B.3.2.1.1 MPD001 Indice di Gulpease

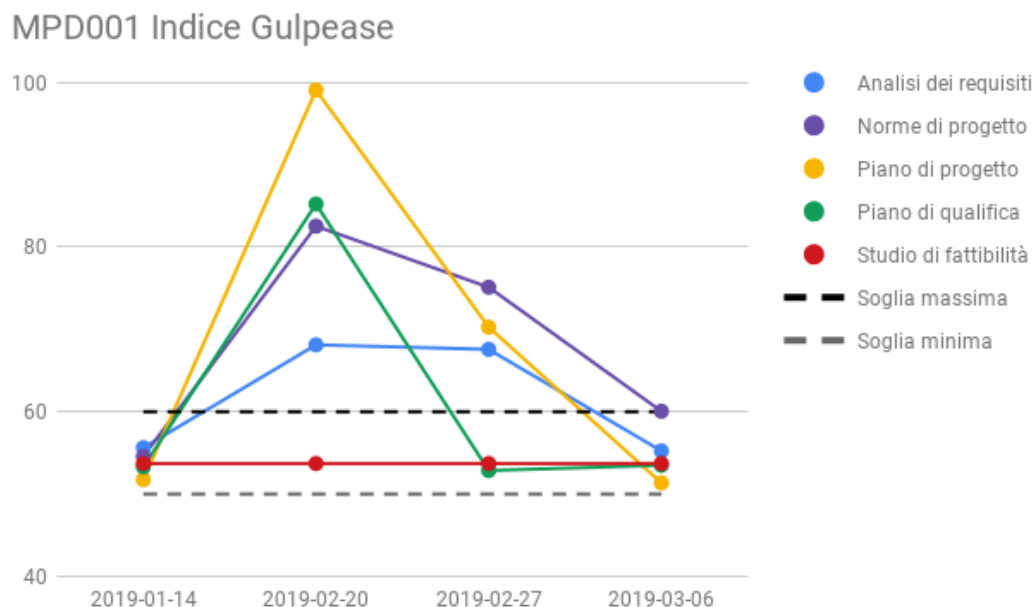


Figura 4: Diagramma con valori misurati tramite MPD001 Indice di Gulpease

- **Obiettivo:** QPD001 Leggibilità del testo.
- **Valore desiderato:** 50 - 60.
- **Descrizione:** vengono mostrati i cambiamenti dei valori dell'Indice di Gulpease nei documenti e la soglia minima e massima (in grigio e nero) in cui devono rientrare i valori dei documenti.
- **Valutazione:** soddisfacente.
- **Considerazioni:** ogni documento ha degli alti e bassi nel corso della stesura, ma infine tutti, eccetto l'Analisi dei Requisiti, rispettano i valori desiderati.

B.3.2.1.2 MPD002 Correttezza ortografica

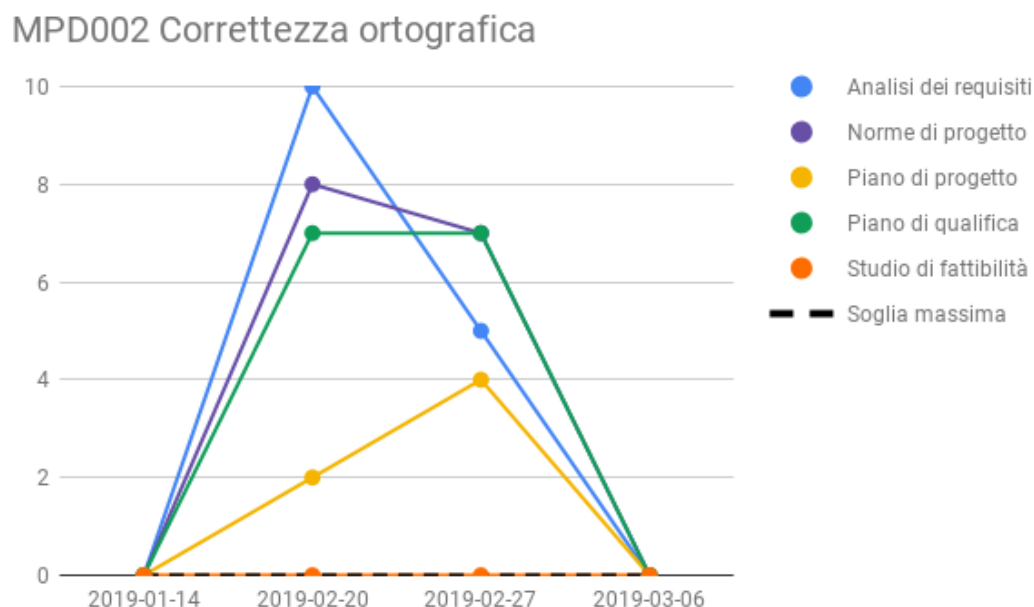


Figura 5: Diagramma con valori misurati tramite MPD002 Correttezza ortografica

- **Obiettivo:** QPD002 Correttezza ortografica.
- **Valore desiderato:** 0.
- **Descrizione:** per ogni documento è mostrato l'andamento del numero di errori ortografici e la soglia (in nero) messa a zero.
- **Valutazione:** soddisfacente.
- **Considerazioni:** dopo la modifica iniziale dei documenti il numero di errori è diminuito, tendendo ad annullarsi per la fine del periodo Revisione di Progettazione.

B.3.2.2 Processi

B.3.2.2.1 MPR001 Varianza della pianificazione

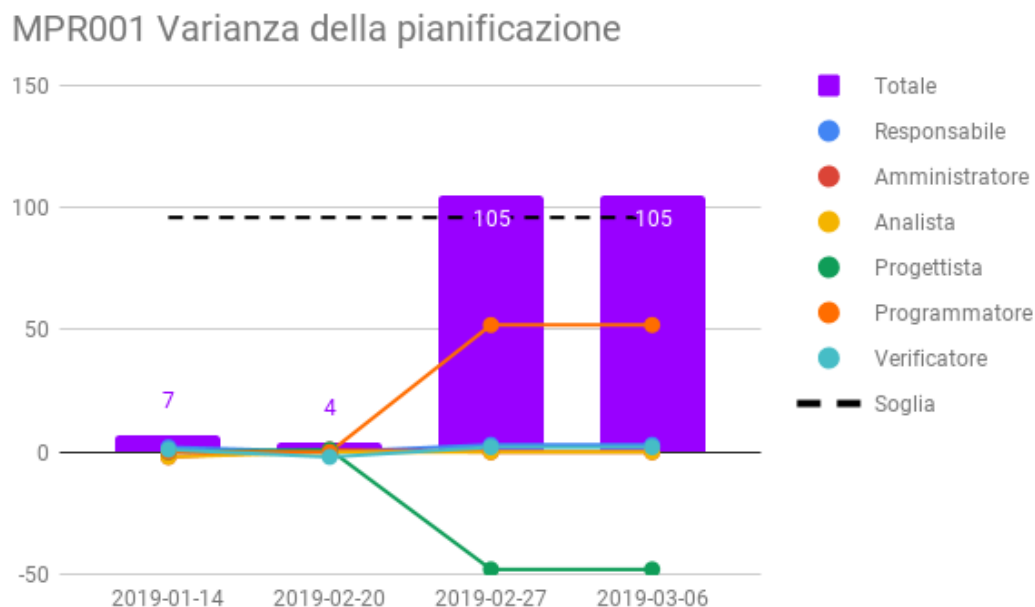


Figura 6: Diagramma con valori misurati tramite MPR001 Varianza della pianificazione

- **Obiettivo:** QPR001 Rispetto dei periodi della pianificazione.
- **Valore desiderato:** 96 ore.
- **Descrizione:** oltre a mostrare le ore di varianza di ogni ruolo, viene mostrato anche il totale delle ore di variazione (in viola) coi relativi valori.
- **Valutazione:** poco soddisfacente.
- **Considerazioni:** Nelle ultime settimane siamo stati costretti a ridurre drasticamente le ore al Progettista per darne al Programmatore essendo a ridosso della consegna del Proof of Concept. Questo purtroppo ha portato ad una variazione troppo elevata del preventivo in termini di ore.

B.3.2.2.2 MPR002 Varianza dei costi

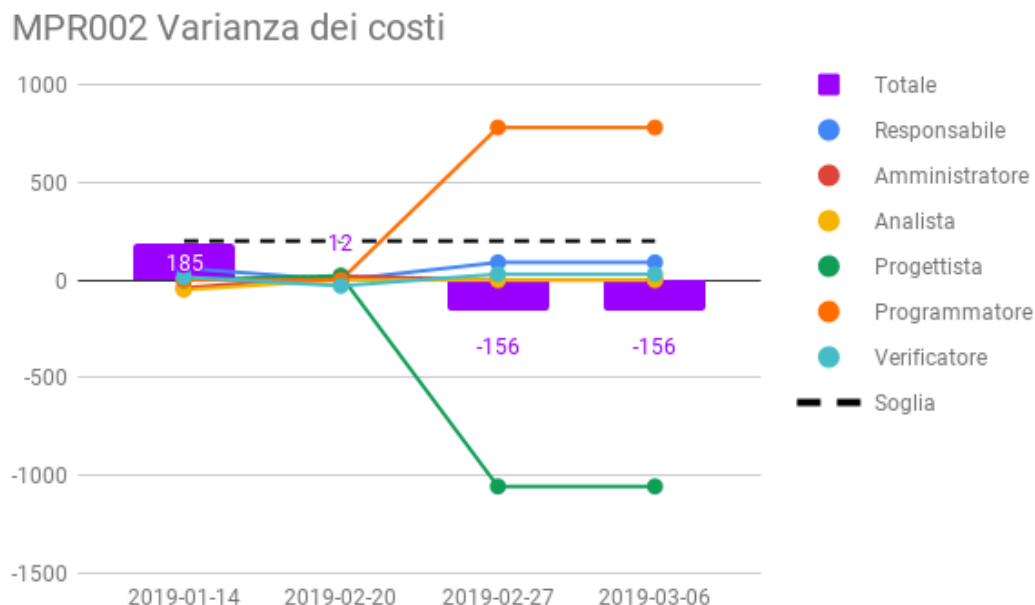


Figura 7: Diagramma con valori misurati tramite MPR002 Varianza dei costi

- **Obiettivo:** QPR002 Varianza del budget.
- **Valore desiderato:** 0 - 200€.
- **Descrizione:** vengono mostrate le variazioni della spesa attribuita ad ogni ruolo, inoltre ne viene mostrato il totale attraverso delle colonne (in viola) coi relativi valori.
- **Valutazione:** soddisfacente.
- **Considerazioni:** la diminuzione delle ore assegnate al Progettista a favore di quelle per il Programmatore, ha portato ad una diminuzione di 156€ del preventivo, un dato che comunque risulta essere soddisfacente.

B.3.2.2.3 MPR003 Aderenza agli standard

MPR003 Aderenza agli standard (del 2019-02-20)

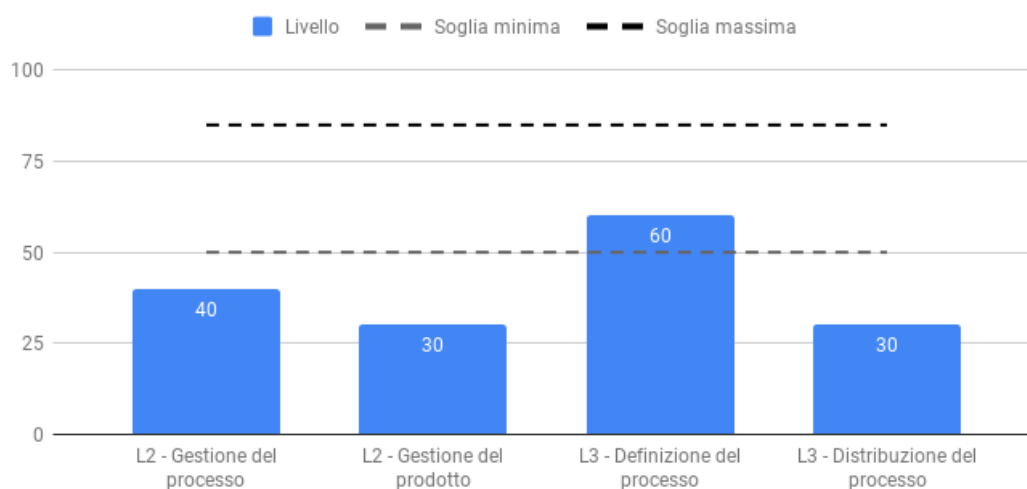


Figura 8: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (1)

MPR003 Aderenza agli standard (del 2019-02-27)

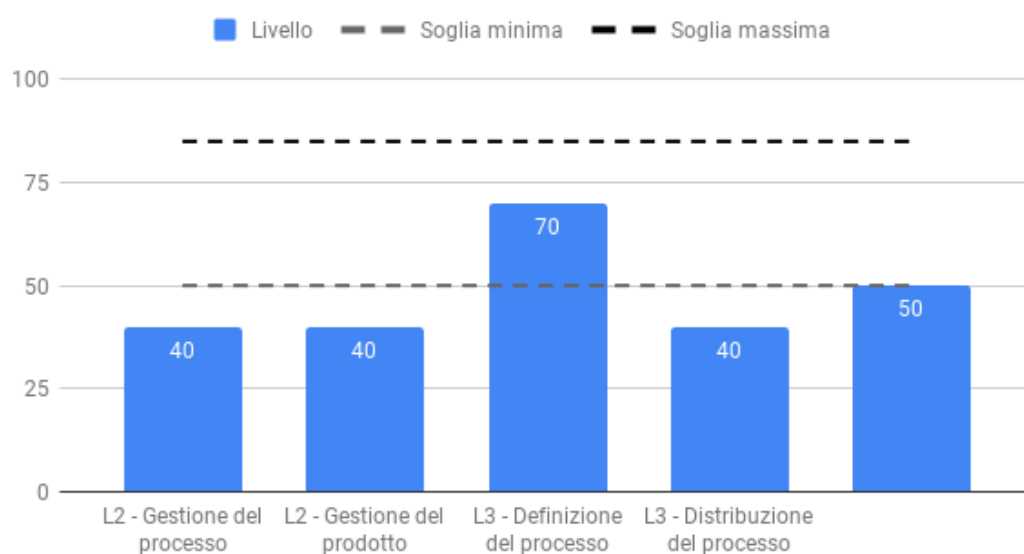


Figura 9: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (2)

MPR003 Aderenza agli standard (del 2019-03-06)

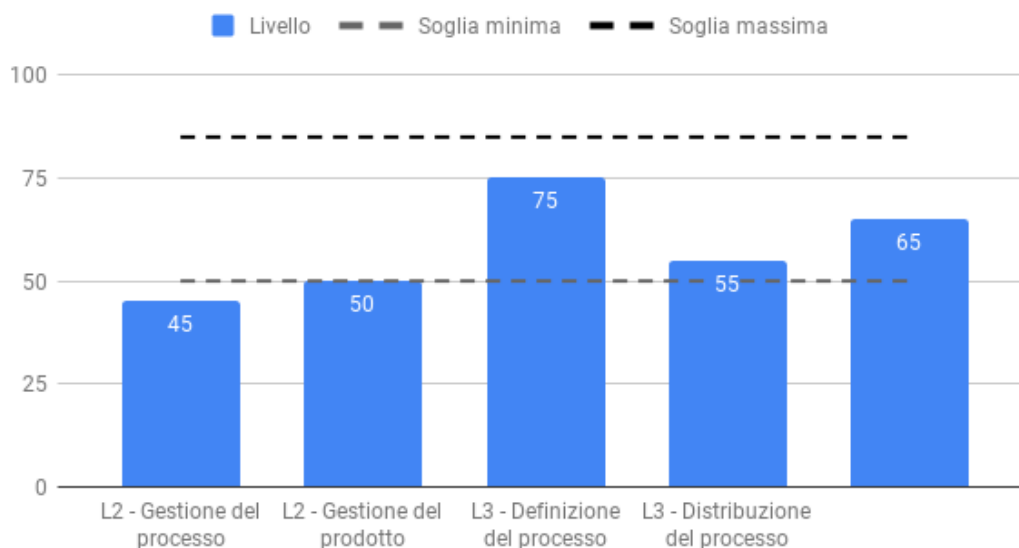


Figura 10: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (3)

- **Obiettivo:** QPR003 Rispetto delle fasi del ciclo di vita.
- **Valore desiderato:** Livello di maturità: 3, Valutazione attributi: L.
- **Descrizione:** per ogni attributo dei processi viene mostrato in che percentuale sono soddisfatti attraverso le colonne (in blu), mostrando anche la soglia minima e massima (in grigio e nero) che questi valori dovrebbero avere.
- **Valutazione:** poco soddisfacente.
- **Considerazioni:** il miglioramento per quanto riguarda l'aderenza agli standard è presente, ma non ancora sufficiente per gli obiettivi prefissati.

B.3.2.3 Software

B.3.2.3.1 MPS001 Presenza di bug



Figura 11: Diagramma con valori misurati tramite MPS001 Presenza di bug

- **Obiettivo:** QPS005 Rispetto delle norme di PEP 8.
- **Valore desiderato:** 0.
- **Descrizione:** viene mostrato il numero totale di bug rilevati da SonarQube.
- **Valutazione:** soddisfacente.
- **Considerazioni:** per la consegna del Proof of Concept abbiamo cercato di togliere tutti i bug presenti nel codice.

B.3.2.3.2 MPS002 Presenza di vulnerabilità



Figura 12: Diagramma con valori misurati tramite MPS002 Presenza di vulnerabilità

- **Obiettivo:** QPS005 Rispetto delle norme di PEP 8.
- **Valore desiderato:** 0.
- **Descrizione:** viene mostrato il numero totale di vulnerabilità rilevate da SonarQube.
- **Valutazione:** soddisfacente.
- **Considerazioni:** per la consegna del Proof of Concept abbiamo cercato di togliere tutte le vulnerabilità presenti nel codice.

B.3.2.3.3 MPS003 Presenza di code smell

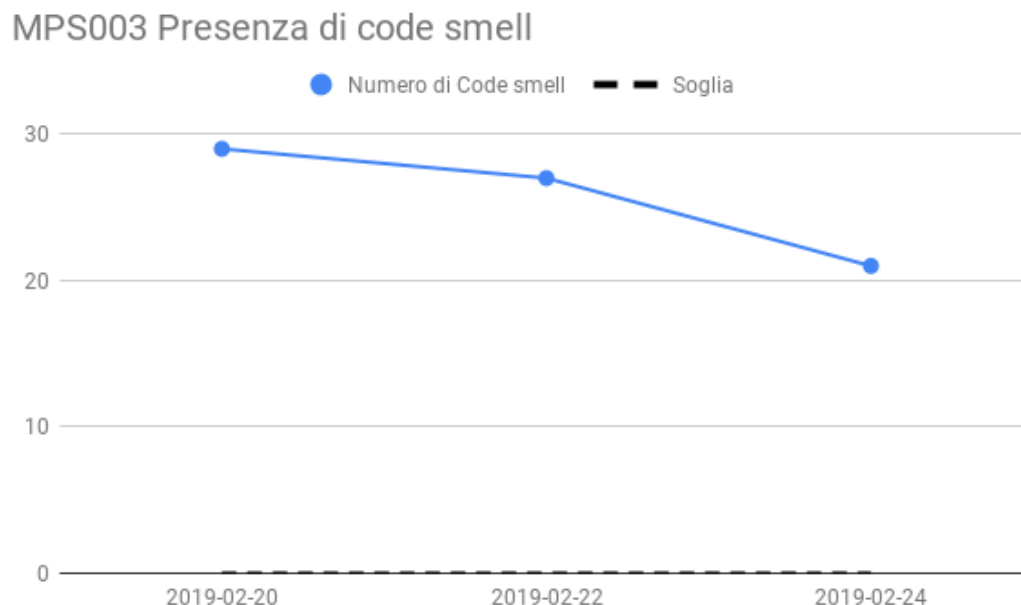


Figura 13: Diagramma con valori misurati tramite MPS003 Presenza di code smell

- **Obiettivo:** QPS005 Rispetto delle norme di PEP 8.
- **Valore desiderato:** 0.
- **Descrizione:** viene mostrato il numero totale di code smell rilevati da SonarQube.
- **Valutazione:** poco soddisfacente.
- **Considerazioni:** dato che non risultano essere errori particolarmente gravi all'interno del codice si ha dato priorità a correggere altri tipi di errori prima della consegna del Proof of Concept. Prevediamo di toglierli completamente nelle parti utilizzeremo successivamente per il prodotto finale.

B.3.2.3.4 MPS004 Duplicazione del codice

MPS004 Duplicazione del codice

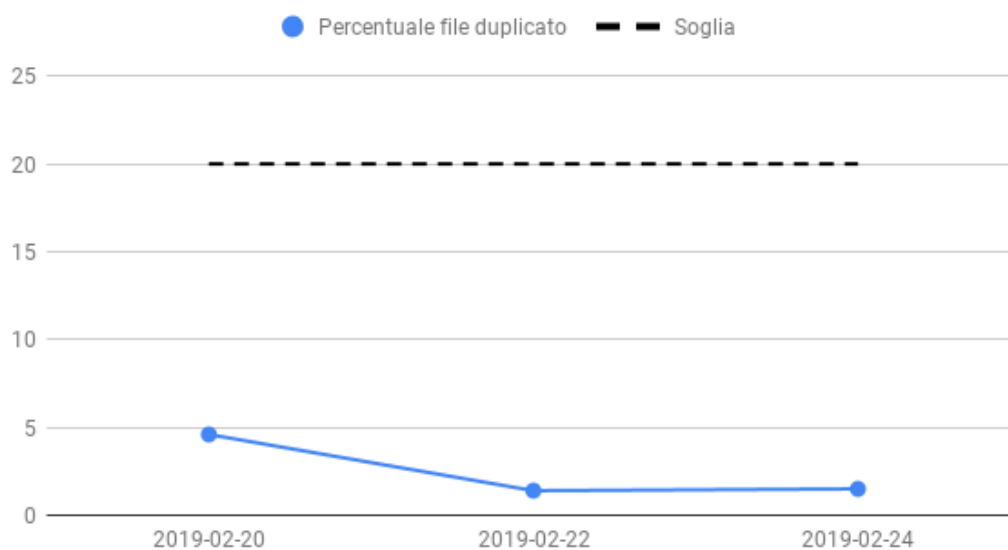


Figura 14: Diagramma con valori misurati tramite MPS004 Duplicazione del codice

- **Obiettivo:** QPS005 Rispetto delle norme di PEP 8.
- **Valore desiderato:** 0 - 20%.
- **Descrizione:** viene mostrata la percentuale di codice duplicato rilevato da SonarQube.
- **Valutazione:** soddisfacente.
- **Considerazioni:** La quantità di codice duplicato, dopo la prima verifica, è rimasto molto basso. Miriamo ad abbassarlo ulteriormente nel prossimo periodo, ma di non raggiungere ad un totale 0%.

C Mitigazione delle variazioni orarie

Alla fine di ogni fase di revisione ci siamo accorti che sono quasi sempre presenti delle variazioni orarie nel consuntivo rispetto a quanto preventivato. Può trattarsi di grandi o piccoli scostamenti, dovuti spesso a dei ritardi accumulati nei periodi immediatamente successivi alla consegna dei prodotti. In ogni caso il gruppo è formato da sei persone e le ore a preventivo sono state calcolate in modo da poter coprire tutte le attività previste dal progetto.

Per fare in modo che le variazioni orarie vengano mitigate cerchiamo di usare sempre con maggior frequenza gli strumenti già impiegati per la coordinazione del gruppo in modo da stabilire con maggior precisione a che punto delle attività siamo arrivati.

Si cerca di tenere sotto controllo le ore dedicate da ciascun membro a ciascun ruolo, in modo da evitare che si verifichino eccessi o difetti rispetto a quanto preventivato.

C.1 Strumenti e mitigazione

Gli strumenti in uso per tenere sotto controllo l'andamento delle attività e le ore impiegate sono gli stessi usati per il coordinamento del gruppo:

- **GITHUB_G**: lo usiamo per tenere traccia di quali attività sono da svolgere e da quale membro è stata completata ciascuna attività.
- **Slack**: lo usiamo per coordinarci, discutere e capire quanto lavoro è servito per il completamento delle attività.

C.2 Mitigazione dei costi

Con l'uso degli strumenti citati cerchiamo di restare, in ogni momento del progetto in cui verrà effettuata una verifica delle ore impiegate da ciascun membro, in pari o al di sotto del limite preventivato.

Nella prima fase del progetto, pur con delle variazioni orarie, siamo riusciti a rimanere al di sotto dei costi posti in preventivo.

Aumentando il numero dei controlli delle ore impiegate tra una revisione e l'altra potremo ottenere risultati migliori.

Nel secondo periodo invece ci siamo sentiti costretti variare di molto le ore assegnate al Programmatore e al Progettista, per poter consegnare i prodotti richiesti.

C.3 Conclusioni

Per mitigare le variazioni verificatesi valuteremo altri strumenti per il controllo più accurato delle ore impiegate da ciascun membro per ciascuna attività. Questo sarà utile specialmente per le fasi successive, perché sarà necessario verificare di soddisfare le aspettative del proponente e al contempo risolvere le eventuali mancanze riscontrate nella prima fase.

Per le variazioni verificatesi nel primo e secondo periodo, ogni membro con delle carenze orarie in un ruolo cercherà di apportare un maggior contributo in quel ruolo nel secondo periodo, in modo da restare in pari con le ore dedicate al progetto.

D Valutazioni per il miglioramento

Questo paragrafo vuole elencare i problemi riscontrati nel corso del progetto evidenziati dalle nostre varie considerazioni e dai risultati riportati all'appendice §B.

Per ogni problema verrà considerata una soluzione di miglioramento da applicare dalla versione attuale del documento alle successive.

D.1 Primo periodo (RR)

D.1.1 Valutazioni sull'organizzazione

- **Problema riscontrato:** la difficoltà maggiore è stata quella di entrare nell'ottica del progetto abituandoci ai cambi di ruolo e ai compiti da svolgere coordinandoci tra di noi.
- **Soluzione proposta:** per ovviare a questo i membri si sono parlati e si è organizzato il lavoro dopo aver studiato il capitolato e i documenti indicati nei riferimenti normativi. Ognuno di noi ha avuto modo di coprire ogni ruolo attivo fino ad ora, perciò sarà meno impegnativo in futuro rispettare la rotazione dei ruoli citata nel *PianoDiProgetto v2.0.0_D*.
- **Problema riscontrato:** le issue create richiedevano dei compiti che si sovrapponevano tra loro, rischiando di effettuare più volte un lavoro inutilmente, oppure, non venendo assegnate, più COMPONENTI_G risolvevano la stessa issue.
- **Soluzione proposta:** l'Amministratore si impegna a creare issue più precise e circoscritte, evitando di dimenticarsi di inserire un assegnatario.

D.1.2 Valutazione dei ruoli

D.1.2.1 Responsabile

- **Problema riscontrato:** la difficoltà più sentita da chi ha ricoperto questo ruolo è stata la suddivisione del lavoro in maniera equa ed omogenea capendo anche i punti di forza e di debolezza di ciascun membro.
- **Soluzione proposta:** per evitare che questo problema persistesse, in corso del progetto è stato usato lo strumento di ISSUE TRACKING SYSTEM_G di GitHub per assegnare i compiti in maniera incrementale e omogenea.

D.1.2.2 Amministratore

- **Problema riscontrato:** la difficoltà maggiore è stata quella di non avere una base di partenza per quanto riguarda la redazione delle Norme di Progetto e sulla granularità delle informazioni contenute al suo interno.
- **Soluzione proposta:** all'inizio e nel corso della revisione dei requisiti sono state fatte delle discussioni per normare gli aspetti più importanti e per avere una solida base di partenza per redarle in maniera incrementale nel corso del progetto e senza modificare quello che è stato scritto in precedenza.

D.1.2.3 Analista

- **Problema riscontrato:** la principale difficoltà è stata la stesura dell'Analisi dei Requisiti in quanto i contenuti di questo documento sono nuovi e per noi di difficile comprensione. In particolar modo l'individuazione dei corretti CASI D'USO_G del progetto.

- **Soluzione proposta:** anche in questo caso, dopo aver studiato autonomamente l'argomento, abbiamo fatto delle riunioni per poter redarre i paragrafi di maggiore importanza come in particolare quello riguardante le tecnologie e i casi d'uso.

D.1.2.4 Verificatore

- **Problema riscontrato:** la verifica è avvenuta in maniera non costante all'inizio e questo ha provocato una mole di documenti da verificare più ampia del previsto.
- **Soluzione proposta:** una pianificazione migliore del lavoro da svolgere ha aiutato in corso d'opera a evitare che questo succedesse nuovamente e si continuerà ad usare nello sviluppo del progetto per evitare che si possa ripresentare.

D.1.3 Valutazione sugli strumenti

D.1.3.1 L^AT_EX

- **Problema riscontrato:** la necessità iniziale di avere dei TEMPLATE_G su cui poter lavorare è stato uno dei problemi iniziali che ha necessitato grande attenzione in quanto non tutti i membri sapevano usare L^AT_EX allo stesso livello.
- **Soluzione proposta:** inizialmente la creazione dei template è andata a essere definita insieme alle norme più importanti per poi continuare la loro costruzione in maniera incrementale.

D.1.3.2 Git

- **Problema riscontrato:** una difficoltà riscontrata, anche se raramente, è stata quella dei conflitti durante i commit sulla REPOSITORY_G in quanto questa è utilizzata da più persone.
- **Soluzione proposta:** tramite coordinazione e azioni varie fornite da Git, i conflitti si sono presentati di rado. Questo anche perché ciascun membro ha sempre lavorato su file separati non sovrapponendo il proprio lavoro con quello degli altri. Un ulteriore miglioramento consiste nel tener costantemente monitorata lo stato della repository mentre si lavora al progetto, in modo tale da effettuare un aggiornamento ogni qual volta avviene un PUSH_G da un'altro di noi.

D.1.4 Integrità di prodotti e strumenti

- **Problema riscontrato:** nel corso del progetto non sono state rispettate tutte le Norme di Progetto prestabilite o nell'aggiornamento della repository sono stati inseriti problemi inattesi.
- **Soluzione proposta:** prima di effettuare una modifica nella repository è tassativo controllare che non si presentino problemi a qualcuno di noi. Dunque, oltre a dover avere ben chiaro il contenuto delle Norme di Progetto, è necessario avvisare tempestivamente chi ha introdotto l'errore nella repository oppure nel prodotto testato o utilizzato in fase di sviluppo. La maggior parte di questi errori dovrebbero essere segnalati dal Verificatore, ma è possibile anche che le segnalazioni arrivino da chi ricopre altri ruoli.

D.1.5 Applicazione dei miglioramenti dopo la RR

Durante il secondo periodo del progetto abbiamo cercato di applicare le soluzioni proposte nei precedenti paragrafi, e in questo riportiamo gli esiti di tali applicazioni.

- **Organizzazione:** dopo il primo periodo ogni membro del gruppo ha compreso meglio i compiti di ogni ruolo e in quale si trova più a suo agio. In questo modo è stato più semplice assegnare i ruoli seguendo le preferenze di ognuno, tenendo sempre in mente che ogni membro del gruppo deve ricoprire ogni ruolo. Anche l'esperienza accumulata con l'issue tracking system ha permesso una miglior assegnazione di ruoli.
- **Ruoli:**
 - **Responsabile:** conoscendo ora meglio i membri del gruppo, il Responsabile riesce ad assegnare meglio i compiti, avendo anche più esperienza con l'issue tracking system.
 - **Amministratore:** essendo già in possesso di un documento per le norme, l'Amministratore è ora più agevolato nello svolgere i suoi compiti.
 - **Analista:** essendo l'argomento dei casi d'uso particolarmente ostico l'idea di trovarsi in gruppo per redarli si è rivelata corretta.
 - **Verificatore:** indicando con anticipo le date o le condizioni per effettuare verifica, la pianificazione di questa fase si è rivelata meno difficoltosa; grazie anche al supporto di strumenti che effettuano verifiche in automatico.
- **Strumenti:**
 - **L^AT_EX:** l'utilizzo per tutto il corso del progetto di questo strumento ha portato ad assorbire tutti i problemi precedentemente riscontrati.
 - **Git:** l'utilizzo maggiore di BRANCH_G, ha dato una più solidità alla repository, causando meno conflitti.
- **Prodotti e strumenti:** l'uso più tassativo delle norme e dei controlli nei prodotti prima di inserire un cambiamento nella repository si è rilevata una scelta EFFICIENTE_G.

D.2 Secondo periodo (RP)

D.2.1 Valutazioni sull'organizzazione

- **Problema riscontrato:** avendo avuto meno tempo da dedicare al progetto a causa della sessione degli esami, ci siamo trovati a comprimere molti compiti in un periodo ristretto e a doverli svolgere in modo celere.
- **Soluzione proposta:** avere ben chiari i giorni di disponibilità fino alla fine del progetto, per effettuare una pianificazione che riscontri sempre meno variazioni.
- **Problema riscontrato:** l'accelerazione dello svolgimento dei compiti ha portato i vari membri del gruppo a non conoscere in modo molto chiaro tutte le componenti del progetto, in quanto il lavoro è stato più parallelizzato.
- **Soluzione proposta:** oltre ad una miglior organizzazione, ci si fa spiegare dal membro del gruppo, che ha effettuato una modifica nei prodotti, in cosa questa consiste.

D.2.2 Valutazione dei ruoli

D.2.2.1 Analista

- **Problema riscontrato:** la natura particolare del progetto ha richiesto più volte la modifica dei casi d'uso anche nel secondo periodo, quando l'attività di analisi dei requisiti doveva solo essere ultimata.
- **Soluzione proposta:** per avere un'idea più chiara dei casi d'uso e, più in generale, dei requisiti del progetto, serve avere uno scambio di opinioni più frequente, sia col committente, che con l'azienda proponente.

D.2.2.2 Progettista

- **Problema riscontrato:** le ore assegnategli sono risultate essere eccessive per il ruolo che ha avuto in questo periodo di lavoro, a discapito di altri ruoli che si sono rivelati essere più presenti del previsto.
- **Soluzione proposta:** purtroppo, essendo la prima esperienza di un progetto simile, è difficile stabilire con precisione quanto verrà ricoperto un ruolo fino alla fine del progetto. Ora però è più chiaro l'andamento che questo può avere. Ad esempio il Progettista nel prossimo periodo (RQ) sarà una delle figure principali per il progetto.

D.2.2.3 Programmatore

- **Problema riscontrato:** si è presentata qualche difficoltà nel modificare del codice scritto da un altro membro del gruppo, questo perché non tutte le norme di codifica sono state rispettate fin da subito.
- **Soluzione proposta:** perseguire in modo più preciso e costante le norme di codifica, dovrebbe rendere il codice più fruibile a tutti i membri del gruppo.

D.2.2.4 Verificatore

- **Problema riscontrato:** nei periodi in cui è stato più impellente rilasciare i prodotti, la verifica è avvenuta in modo meno frequente.
- **Soluzione proposta:** inserire, in modo più stringete, il numero o le date per le fasi di verifica obbligatorie che non possono essere prorogate.

D.2.3 Valutazione sugli strumenti

D.2.3.1 Docker

- **problema riscontrato:** conoscere una nuova tecnologia richiede tempo, e nel nostro caso con DOCKER_G abbiamo impiegato più del previsto a conoscerne le caratteristiche per noi di interesse.
- **Soluzione proposta:** individuare le parti di interesse di una tecnologia, che verranno utilizzate, per poi dividersele tra i vari membri del gruppo. Successivamente, in una riunione, riportare quanto imparato.