



AlphaSix

# AlphaSix

## Piano di Qualifica

### Informazioni sul documento

<b>Nome Documento</b>	PianoDiQualifica v4.0.0.pdf
<b>Versione</b>	4.0.0
<b>Data di Creazione</b>	27 novembre 2018
<b>Data ultima modifica</b>	13 maggio 2019
<b>Stato</b>	Approvato
<b>Redazione</b>	Laura Cameran Matteo Marchiori
<b>Verifica</b>	Samuele Gardin
<b>Approvazione</b>	Timoty Granziero
<b>Uso</b>	Esterno
<b>Distribuzione</b>	AlphaSix
<b>Destinato a</b>	Prof. Tullio Vardanega, Prof. Riccardo Cardin, Imola Informatica
<b>Email di riferimento</b>	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
4.0.0	Approvazione per il rilascio	Responsabile	Nicola Carlesso	2019-05-13
3.3.0	Verifica finale	Verificatore	Samuele Gardin	2019-05-12
3.2.1	Aggiunte conclusioni e valutazioni sul periodo RA in §B.4	Responsabile	Nicola Carlesso	2019-05-08
3.2.0	Verifica	Verificatore	Samuele Gardin	2019-05-05
3.1.2	Aggiunto §A.5.2.2.8	Amministratore	Nicola Carlesso	2019-05-04
3.1.1	Aggiunto §A.5 per inserire il resoconto dell'ultimo periodo	Responsabile	Samuele Gardin	2019-05-03
3.1.0	Verifica	Verificatore	Matteo Marchiori	2019-04-29
3.0.2	Scorporata mitigazione dei costi	Amministratore	Samuele Gardin	2019-04-27
3.0.1	Aggiunto processo di validazione in §2.3.7	Amministratore	Nicola Carlesso	2019-04-25
3.0.0	Approvazione per il rilascio	Responsabile	Nicola Carlesso	2019-04-11
2.3.0	Verifica finale	Verificatore	Samuele Gardin	2019-04-10
2.2.2	Aggiunta degli ultimi test di unità mancanti in §3.5.6	Verificatore	Laura Cameran	2019-04-04
2.2.1	Ampliamento di §3.5.6 per aggiunta di test di unità	Verificatore	Laura Cameran	2019-04-01
2.2.0	Verifica	Verificatore	Samuele Gardin	2019-03-29
2.1.4	Aggiunte considerazioni §A per periodo RR ed RP	Verificatore	Ciprian Voinea	2019-03-28
2.1.3	Aggiunto §A.4 per inserire il resoconto dell'ultimo periodo	Progettista	Samuele Gardin	2019-03-26
2.1.2	Aggiunti titoli dei paragrafi §3.5.6 e §3.5.6.1 per iniziare ad inserire i test d'unità	Progettista	Laura Cameran	2019-03-26
2.1.1	Aggiunte valutazioni per il miglioramento in §B.3 per l'ultimo periodo	Progettista	Matteo Marchiori	2019-03-25
2.1.0	Verifica	Verificatore	Samuele Gardin	2019-03-25
2.0.1	Aggiunte metriche in §3.3.2 e i nuovi processi istanziati presenti in §2.3.5 e §2.3.6	Verificatore	Nicola Carlesso	2019-03-24
2.0.1	Cancellati alcuni test in §3.5 perché obsoleti, rimosso il tracciamento perché da aggiornare con gli ultimi test	Verificatore	Matteo Marchiori	2019-03-23
2.0.0	Approvazione per il rilascio	Responsabile	Samuele Gardin	2019-03-05

Versione	Descrizione	Ruolo	Nominativo	Data
1.9.8	Verifica finale	Verificatore	Timoty Granziero	2019-03-03
1.9.2	Aggiunta §B.2	Verificatore	Nicola Carlesso	2019-03-02
1.9.0	Aggiunta parte di §A.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 §3.5.5 e §3.5.5.1	Progettista	Laura Cameran	2019-02-23
1.1.3	Sistematte sezioni §3.5.3.1 e §3.5.5.1	Progettista	Laura Cameran	2019-02-22
1.1.2	Aggiunte sezioni §3.5.3.1 e §3.5.5.1	Verificatore	Laura Cameran	2019-02-19
1.1.1	Ampliamento §A	Progettista	Ciprian Voinea	2019-02-13
1.1.0	Verifica	Verificatore	Nicola Carlesso	2019-02-10
1.0.3	Aggiunte sezioni §3.5.3, §3.5.5 e rivisitazione §3.5.2	Verificatore	Laura Cameran	2019-02-06
1.0.2	Sistemazione §A	Progettista	Nicola Carlesso	2019-02-01
1.0.1	Aggiunte sezioni §3.5, §3.5.1 §3.5.4 e §3.5.2	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 §B	Verificatore	Matteo Marchiori	2019-01-11
0.1.1	Aggiunta §A	Verificatore	Nicola Carlesso	2019-01-08
0.1.0	Verifica documento	Verificatore	Ciprian Voinea	2018-12-10
0.0.4	Aggiunto §2	Verificatore	Nicola Carlesso	2018-12-02
0.0.3	Aggiunto §3	Verificatore	Timoty Granziero	2018-12-01
0.0.2	Aggiunto §1	Verificatore	Nicola Carlesso	2018-11-29
0.0.1	Creazione template	Redattore	Timoty Granziero	2018-11-27

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Glossario e documenti esterni . . . . .	1
1.2	Premessa . . . . .	1
1.3	Scopo del documento . . . . .	1
1.4	Scopo del prodotto . . . . .	1
1.5	Riferimenti . . . . .	1
1.5.1	Normativi . . . . .	1
1.5.2	Informativi . . . . .	1
<b>2</b>	<b>Qualità di processo</b>	<b>3</b>
2.1	Scopo . . . . .	3
2.2	Nomenclatura metriche e obiettivi di qualità . . . . .	3
2.3	Processi . . . . .	3
2.3.1	PROC001 Pianificazione del progetto, organizzazione e struttura . . . . .	3
2.3.1.1	Funzioni . . . . .	4
2.3.1.2	Metriche . . . . .	4
2.3.1.3	Obiettivi . . . . .	4
2.3.2	PROC002 Analisi . . . . .	4
2.3.2.1	Funzioni . . . . .	5
2.3.2.2	Metriche . . . . .	5
2.3.2.3	Obiettivi . . . . .	5
2.3.3	PROC003 Produzione documenti . . . . .	5
2.3.3.1	Funzioni . . . . .	5
2.3.3.2	Obiettivi . . . . .	6
2.3.4	PROC004 Verifica . . . . .	6
2.3.4.1	Funzioni . . . . .	6
2.3.4.2	Metriche . . . . .	6
2.3.4.3	Obiettivi . . . . .	6
2.3.5	PROC005 Progettazione . . . . .	6
2.3.5.1	Funzioni . . . . .	6
2.3.5.2	Metriche . . . . .	7
2.3.5.3	Obiettivi . . . . .	7
2.3.6	PROC006 Codifica . . . . .	7
2.3.6.1	Funzioni . . . . .	7
2.3.6.2	Metriche . . . . .	7
2.3.6.3	Obiettivi . . . . .	8
2.3.7	PROC007 Validazione . . . . .	8
2.3.7.1	Funzioni . . . . .	8
2.3.7.2	Metriche . . . . .	8
2.3.7.3	Obiettivi . . . . .	8
2.4	Tabella qualità di processo . . . . .	9
<b>3</b>	<b>Qualità di prodotto</b>	<b>13</b>
3.1	Scopo . . . . .	13
3.2	Nomenclatura metriche ed obiettivi di qualità . . . . .	13
3.3	Prodotti . . . . .	13
3.3.1	Documenti . . . . .	13
3.3.1.1	Metriche . . . . .	13
3.3.1.2	Obiettivi . . . . .	14

3.3.2	Software . . . . .	14
3.3.2.1	Metriche . . . . .	14
3.3.2.2	Obiettivi . . . . .	15
3.4	Tabelle qualità di prodotto . . . . .	15
3.5	Test . . . . .	20
3.5.1	Modello a V . . . . .	20
3.5.2	Classificazione e struttura dei test . . . . .	20
3.5.3	Test di validazione . . . . .	21
3.5.3.1	Tracciamento . . . . .	34
3.5.4	Test di sistema . . . . .	36
3.5.4.1	Tracciamento . . . . .	39
3.5.5	Test d'integrazione . . . . .	41
3.5.5.1	Tracciamento . . . . .	43
3.5.6	Test di unità . . . . .	45
3.5.6.1	Tracciamento . . . . .	48
<b>A</b>	<b>Resoconto delle attività di verifica</b>	<b>50</b>
A.1	Classificazione dei risultati . . . . .	50
A.2	Primo periodo (RR) . . . . .	50
A.2.1	Riassunto delle attività di verifica . . . . .	50
A.2.2	Risultati delle verifiche tramite analisi . . . . .	50
A.2.2.1	Documenti . . . . .	51
A.2.2.2	Processi . . . . .	52
A.2.3	Considerazioni sui risultati e sull'esito della revisione . . . . .	53
A.3	Secondo periodo (RP) . . . . .	53
A.3.1	Riassunto delle attività di verifica . . . . .	53
A.3.2	Risultati delle verifiche tramite analisi . . . . .	53
A.3.2.1	Documenti . . . . .	54
A.3.2.1.1	MPD001 Indice di Gulpease . . . . .	54
A.3.2.1.2	MPD002 Correttezza ortografica . . . . .	55
A.3.2.2	Processi . . . . .	56
A.3.2.2.1	MPR001 Varianza della pianificazione . . . . .	56
A.3.2.2.2	MPR002 Varianza dei costi . . . . .	57
A.3.2.2.3	MPR003 Aderenza agli standard . . . . .	58
A.3.2.3	Software . . . . .	60
A.3.2.3.1	MPS001 Presenza di bug . . . . .	60
A.3.2.3.2	MPS002 Presenza di vulnerabilità . . . . .	61
A.3.2.3.3	MPS003 Presenza di code smell . . . . .	62
A.3.2.3.4	MPS004 Duplicazione del codice . . . . .	63
A.3.3	Considerazioni sui risultati e sull'esito della revisione . . . . .	63
A.3.3.1	Considerazioni Proof of Concept (PoC) . . . . .	63
A.4	Terzo periodo (RQ) . . . . .	64
A.4.1	Riassunto delle attività di verifica . . . . .	64
A.4.2	Risultati delle verifiche tramite analisi . . . . .	64
A.4.2.1	Documenti . . . . .	64
A.4.2.1.1	MPD001 Indice di Gulpease . . . . .	64
A.4.2.1.2	MPD002 Correttezza ortografica . . . . .	65
A.4.2.2	Processi . . . . .	66
A.4.2.2.1	MPR001 Varianza della pianificazione . . . . .	66
A.4.2.2.2	MPR002 Varianza dei costi . . . . .	67
A.4.2.2.3	MPR003 Aderenza agli standard . . . . .	68
A.4.2.2.4	MPR004 Frequenza di commit nella repository . . . . .	70

A.4.2.2.5	MPR010 Moduli codificati prima della progettazione . .	71
A.4.2.2.6	MPR011 Numero di design pattern applicati . . . . .	72
A.4.2.2.7	MPR012 Moduli non testati . . . . .	73
A.4.2.3	Software . . . . .	73
A.4.2.3.1	MPS001 Presenza di bug . . . . .	74
A.4.2.3.2	MPS002 Presenza di vulnerabilità . . . . .	75
A.4.2.3.3	MPS003 Presenza di code smell . . . . .	76
A.4.2.3.4	MPS004 Duplicazione del codice . . . . .	77
A.4.2.3.5	MPS017 Code coverage . . . . .	78
A.4.2.3.6	MPS019 Test di integrazione . . . . .	79
A.4.2.3.7	MPS020 Test di unità . . . . .	80
A.5	Quarto periodo (RA) . . . . .	80
A.5.1	Riassunto delle attività di verifica . . . . .	80
A.5.2	Risultati delle verifiche tramite analisi . . . . .	80
A.5.2.1	Documenti . . . . .	81
A.5.2.1.1	MPD001 Indice di Gulpease . . . . .	81
A.5.2.1.2	MPD002 Correttezza ortografica . . . . .	82
A.5.2.2	Processi . . . . .	83
A.5.2.2.1	MPR001 Varianza della pianificazione . . . . .	83
A.5.2.2.2	MPR002 Varianza dei costi . . . . .	84
A.5.2.2.3	MPR003 Aderenza agli standard . . . . .	85
A.5.2.2.4	MPR004 Frequenza di commit nella repository . . . . .	87
A.5.2.2.5	MPR010 Moduli codificati prima della progettazione . .	88
A.5.2.2.6	MPR011 Numero di design pattern applicati . . . . .	89
A.5.2.2.7	MPR012 Moduli non testati . . . . .	90
A.5.2.2.8	MPR013 Percentuale metriche rispettate . . . . .	91
A.5.2.3	Software . . . . .	91
A.5.2.3.1	MPS001 Presenza di bug . . . . .	92
A.5.2.3.2	MPS002 Presenza di vulnerabilità . . . . .	93
A.5.2.3.3	MPS003 Presenza di code smell . . . . .	94
A.5.2.3.4	MPS004 Duplicazione del codice . . . . .	95
A.5.2.3.5	MPS017 Code coverage . . . . .	96
A.5.2.3.6	MPS018 Test di sistema . . . . .	97
A.5.2.3.7	MPS019 Test di integrazione . . . . .	98
A.5.2.3.8	MPS020 Test di unità . . . . .	99
<b>B</b>	<b>Valutazioni per il miglioramento</b>	<b>100</b>
B.1	Primo periodo (RR) . . . . .	100
B.1.1	Valutazioni sull'organizzazione . . . . .	100
B.1.2	Valutazione dei ruoli . . . . .	100
B.1.2.1	Responsabile . . . . .	100
B.1.2.2	Amministratore . . . . .	100
B.1.2.3	Analista . . . . .	100
B.1.2.4	Verificatore . . . . .	101
B.1.3	Valutazione sugli strumenti . . . . .	101
B.1.3.1	L <sup>A</sup> T <sub>E</sub> X . . . . .	101
B.1.3.2	Git . . . . .	101
B.1.4	Integrità di prodotti e strumenti . . . . .	101
B.1.5	Applicazione dei miglioramenti dopo la RR . . . . .	101
B.2	Secondo periodo (RP) . . . . .	102
B.2.1	Valutazioni sull'organizzazione . . . . .	102
B.2.2	Valutazione dei ruoli . . . . .	102



	B.2.2.1	Analista . . . . .	102
	B.2.2.2	Progettista . . . . .	103
	B.2.2.3	Programmatore . . . . .	103
	B.2.2.4	Verificatore . . . . .	103
B.2.3		Valutazione sugli strumenti . . . . .	103
	B.2.3.1	Docker . . . . .	103
B.3		Terzo periodo (RQ) . . . . .	103
	B.3.1	Valutazioni sull'organizzazione . . . . .	103
	B.3.2	Valutazione dei ruoli . . . . .	103
	B.3.2.1	Analista . . . . .	103
	B.3.2.2	Progettista . . . . .	104
	B.3.2.3	Programmatore . . . . .	104
	B.3.2.4	Verificatore . . . . .	104
	B.3.3	Valutazione sugli strumenti . . . . .	104
	B.3.3.1	Rancher . . . . .	104
B.4		Quarto periodo (RA) . . . . .	104
	B.4.1	Valutazioni conclusive sull'intero progetto . . . . .	104

## Elenco delle tabelle

1	Obiettivi di qualità per il PROC001 . . . . .	9
2	Obiettivi di qualità per il PROC002 . . . . .	10
3	Obiettivi di qualità per il PROC003 . . . . .	10
4	Obiettivi di qualità per il PROC004 . . . . .	11
5	Obiettivi di qualità per il PROC005 . . . . .	11
6	Obiettivi di qualità per il PROC006 . . . . .	11
7	Obiettivi di qualità per il PROC007 . . . . .	12
8	Obiettivi di qualità per i documenti . . . . .	16
9	Obiettivi di qualità per il software (1) . . . . .	17
10	Obiettivi di qualità per il software (2) . . . . .	18
11	Obiettivi di qualità per il software (3) . . . . .	19
12	Elenco dei test di validazione (1) . . . . .	21
13	Elenco dei test di validazione (2) . . . . .	22
14	Elenco dei test di validazione (3) . . . . .	23
15	Elenco dei test di validazione (4) . . . . .	24
16	Elenco dei test di validazione (5) . . . . .	25
17	Elenco dei test di validazione (6) . . . . .	26
18	Elenco dei test di validazione (7) . . . . .	27
19	Elenco dei test di validazione (8) . . . . .	28
20	Elenco dei test di validazione (9) . . . . .	29
21	Elenco dei test di validazione (10) . . . . .	30
22	Elenco dei test di validazione (11) . . . . .	31
23	Elenco dei test di validazione (12) . . . . .	32
24	Elenco dei test di validazione (13) . . . . .	33
25	Elenco dei test in correlazioni con i requisiti (1) . . . . .	34
26	Elenco dei test in correlazioni con i requisiti (2) . . . . .	35
27	Elenco dei test di sistema (1) . . . . .	36
28	Elenco dei test di sistema (2) . . . . .	37
29	Elenco dei test di sistema (3) . . . . .	38
30	Elenco dei test in correlazioni con i requisiti (1) . . . . .	39
31	Elenco dei test in correlazioni con i requisiti (2) . . . . .	40
32	Elenco dei test d'integrazione (1) . . . . .	41
33	Elenco dei test d'integrazione (2) . . . . .	42
34	Tracciamento dei test in correlazione con le componenti (1) . . . . .	43
35	Tracciamento dei test in correlazione con le componenti (2) . . . . .	44
36	Elenco dei test di unità (1) . . . . .	45
37	Elenco dei test di unità (2) . . . . .	46
38	Elenco dei test di unità (3) . . . . .	47
39	Elenco dei test in correlazione con le componenti (1) . . . . .	48
40	Elenco dei test in correlazione con le componenti (2) . . . . .	49
41	Risultati di MPD001 Indice Gulpease . . . . .	51
42	Risultati di MPD002 Correttezza ortografica . . . . .	51
43	Risultati di MPR001 Varianza della pianificazione . . . . .	52
44	Risultati di MPR002 Varianza dei costi . . . . .	52
45	Risultati di MPR003 Aderenza agli standard . . . . .	52
46	Risultati di MPR004 Frequenza commit nella repository . . . . .	53
47	Risultati di MPR009 Frequenza controllo prodotti . . . . .	53



## Elenco delle figure

1	Modello a V . . . . .	20
2	Diagramma con valori misurati tramite MPD001 Indice di Gulpease . . . . .	54
3	Diagramma con valori misurati tramite MPD002 Correttezza ortografica . . . . .	55
4	Diagramma con valori misurati tramite MPR001 Varianza della pianificazione . .	56
5	Diagramma con valori misurati tramite MPR002 Varianza dei costi . . . . .	57
6	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (1) . . .	58
7	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (2) . . .	58
8	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (3) . . .	59
9	Diagramma con valori misurati tramite MPS001 Presenza di bug . . . . .	60
10	Diagramma con valori misurati tramite MPS002 Presenza di vulnerabilità . . . .	61
11	Diagramma con valori misurati tramite MPS003 Presenza di code smell . . . . .	62
12	Diagramma con valori misurati tramite MPS004 Duplicazione del codice . . . . .	63
13	Diagramma con valori misurati tramite MPD001 Indice di Gulpease . . . . .	64
14	Diagramma con valori misurati tramite MPD002 Correttezza ortografica . . . . .	65
15	Diagramma con valori misurati tramite MPR001 Varianza della pianificazione . .	66
16	Diagramma con valori misurati tramite MPR002 Varianza dei costi . . . . .	67
17	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (1) . . .	68
18	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (2) . . .	68
19	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (3) . . .	69
20	Diagramma con valori misurati tramite MPR004 Frequenza di commit nella repository . . . . .	70
21	Diagramma con valori misurati tramite MPR010 Moduli codificati prima della progettazione . . . . .	71
22	Diagramma con valori misurati tramite MPR011 Numero di design pattern applicati	72
23	Diagramma con valori misurati tramite MPR012 Moduli non testati . . . . .	73
24	Diagramma con valori misurati tramite MPS001 Presenza di bug . . . . .	74
25	Diagramma con valori misurati tramite MPS002 Presenza di vulnerabilità . . . .	75
26	Diagramma con valori misurati tramite MPS003 Presenza di code smell . . . . .	76
27	Diagramma con valori misurati tramite MPS004 Duplicazione del codice . . . . .	77
28	Diagramma con valori misurati tramite MPS017 Code coverage . . . . .	78
29	Diagramma con valori misurati tramite MPS019 Test di integrazione . . . . .	79
30	Diagramma con valori misurati tramite MPS020 Test di unità . . . . .	80
31	Diagramma con valori misurati tramite MPD001 Indice di Gulpease . . . . .	81
32	Diagramma con valori misurati tramite MPD002 Correttezza ortografica . . . . .	82
33	Diagramma con valori misurati tramite MPR001 Varianza della pianificazione . .	83
34	Diagramma con valori misurati tramite MPR002 Varianza dei costi . . . . .	84
35	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (1) . . .	85
36	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (2) . . .	85
37	Diagramma con valori misurati tramite MPR003 Aderenza agli standard (3) . . .	86
38	Diagramma con valori misurati tramite MPR004 Frequenza di commit nella repository . . . . .	87
39	Diagramma con valori misurati tramite MPR010 Moduli codificati prima della progettazione . . . . .	88
40	Diagramma con valori misurati tramite MPR011 Numero di design pattern applicati	89
41	Diagramma con valori misurati tramite MPR012 Moduli non testati . . . . .	90
42	Diagramma con valori misurati tramite MPR013 Percentuale metriche rispettate	91
43	Diagramma con valori misurati tramite MPS001 Presenza di bug . . . . .	92
44	Diagramma con valori misurati tramite MPS002 Presenza di vulnerabilità . . . .	93
45	Diagramma con valori misurati tramite MPS003 Presenza di code smell . . . . .	94



46	Diagramma con valori misurati tramite MPS004 Duplicazione del codice . . . . .	95
47	Diagramma con valori misurati tramite MPS017 Code coverage . . . . .	96
48	Diagramma con valori misurati tramite MPS018 Test di sistema . . . . .	97
49	Diagramma con valori misurati tramite MPS019 Test di integrazione . . . . .	98
50	Diagramma con valori misurati tramite MPS020 Test di unità . . . . .	99

# 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 v4.0.0<sub>D</sub>*).
- **G**: indica che il termine si riferisce ad una voce riportata nel *Glossario v3.0.0<sub>D</sub>* (ad esempio *REDMINE<sub>G</sub>*).

## 1.2 Premessa

Il DOCUMENTO<sub>G</sub> che segue verrà prodotto incrementalmente al presentarsi di esigenze di perseguimento della QUALITÀ<sub>G</sub>. 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<sub>G</sub> che rispettiamo per perseguire la qualità di prodotto, sia per quanto riguarda il prodotto in sé, sia per quanto riguarda i PROCESSI<sub>G</sub> 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<sub>G</sub> è creare un APPLICATIVO<sub>G</sub> per poter gestire i messaggi o le segnalazioni provenienti da diversi prodotti per la realizzazione di software. Queste segnalazioni passano attraverso un BROKER<sub>G</sub> 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<sub>G</sub> 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<sub>G</sub>, di reindirizzare i messaggi ricevuti al destinatario più appropriato.

## 1.5 Riferimenti

### 1.5.1 Normativi

1. *NormeDiProgetto v4.0.0<sub>D</sub>*
2. CAPITOLATO<sub>G</sub> d'appalto C1:  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1.pdf>

### 1.5.2 Informativi

1. Presentazione capitolato C1:  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1p.pdf>
2. Slide del corso di Ingegneria del Software:

- (a) Qualità di prodotto:  
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L13.pdf>
  - (b) Qualità di processo:  
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L14.pdf>
  - (c) Verifica e validazione:  
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L16.pdf>
- 3. THE TWELVE-FACTOR APP<sub>G</sub>: [https://12factor.net/it/#i\\_dodici\\_fattori](https://12factor.net/it/#i_dodici_fattori)
  - 4. Standard ISO/IEC 9126: [https://it.wikipedia.org/wiki/ISO/IEC\\_9126](https://it.wikipedia.org/wiki/ISO/IEC_9126)
  - 5. CICLO DI DEMING<sub>G</sub>: [https://it.wikipedia.org/wiki/Ciclo\\_di\\_Deming](https://it.wikipedia.org/wiki/Ciclo_di_Deming)
  - 6. Modello a V: <https://en.wikipedia.org/wiki/V-Model>
  - 7. 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](https://elearning.unipd.it/math/pluginfile.php/43469/mod_resource/content/4/7-Testing.pdf)
  - 8. Documentazione PEP 8<sub>G</sub>:  
<https://www.python.org/dev/peps/pep-0008/>

## 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<sub>G</sub> 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<sub>G</sub> descritti nelle *NormeDiProgetto v4.0.0<sub>D</sub>*.

### 2.2 Nomenclatura metriche e obiettivi di qualità

La nomenclatura degli obiettivi di qualità, delle METRICHE<sub>G</sub> ed il loro funzionamento è spiegato in dettaglio nelle *NormeDiProgetto v4.0.0<sub>D</sub>*. 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<sub>G</sub> ha lo scopo di pianificare il lavoro da svolgere per soddisfare i REQUISITI<sub>G</sub> richiesti dal PROGETTO<sub>G</sub>. È in questo processo che viene messo in atto il nostro WAY OF WORKING<sub>G</sub> che ha un'importanza particolarmente alta perché i suoi risultati vanno a condizionare l'esito della qualità dell'intero progetto.

### 2.3.1.1 Funzioni

- **Sviluppare SOTTO-PROCESSI<sub>G</sub>**: 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<sub>G</sub> durante il progetto.
- **FORMAZIONE<sub>G</sub> 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<sub>G</sub>**: è necessario conoscere le proprie RISORSE<sub>G</sub> in termini di tempo  $\frac{\text{costo}}{\text{persona}}$  in modo tale da restare il più fedeli possibile al PREVENTIVO<sub>G</sub> stilato.

### 2.3.1.2 Metriche

- MPR001 Varianza della PIANIFICAZIONE<sub>G</sub>
- MPR002 Varianza dei costi
- MPR003 Aderenza agli standard
- MPR004 Frequenza COMMIT<sub>G</sub> nella REPOSITORY<sub>G</sub>

### 2.3.1.3 Obiettivi

- **QPR001 Rispetto dei periodi della pianificazione**: all'interno della presente nel *PianoDiProgetto v4.0.0<sub>D</sub>*, 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<sub>G</sub>**: 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 v3.0.0<sub>D</sub>*), ma anche all'analisi dei rischi (presente nel *PianoDiProgetto v4.0.0<sub>D</sub>*) e altre attività di identificazione e analisi riportate qui di seguito tra le funzioni. La prima parte di questo processo consiste nel poter dire se il progetto risulta fattibile per le sue possibilità in termini di competenze e risorse. Tale processo non deve essere rendicontato per il preventivo a finire.

### 2.3.2.1 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<sub>G</sub> 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.

### 2.3.2.2 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.

### 2.3.2.3 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.

### 2.3.3.1 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

### 2.3.3.2 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.

#### 2.3.4.1 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 *NormeDiProgetto v4.0.0<sub>D</sub>*:** 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.

#### 2.3.4.2 Metriche

- MPR009 Frequenza controllo prodotti

#### 2.3.4.3 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<sub>G</sub>.
- **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.

### 2.3.5 PROC005 Progettazione

Tale processo si attiva successivamente al processo di analisi e si occupa di restituire la progettazione completa dei prodotti che verranno infine eseguiti.

La progettazione è fondamentale per tracciare il percorso di esecuzione del processo di codifica, onde per cui deve essere eseguito con criterio e cautela.

#### 2.3.5.1 Funzioni

- **Progettare i moduli prima di codificarli:** specialmente alle prime esperienze di progetti come quello che stiamo svolgendo ora, è facile cadere nella tentazione di codificare un MODULO<sub>G</sub> prima di averlo progettato. In questo modo però, spesso non si realizza un modulo nell'ottica di renderlo estensibile, mantenibile e disaccoppiato dagli altri moduli. La progettazione serve invece ad evitare questi difetti in un prodotto, per questo è meglio svolgerla prima della codifica.



- **Applicazione ponderata dei design pattern:** i DESIGN PATTERN<sub>G</sub> sono molto utili ai fini della progettazione per ottenere più facilmente le caratteristiche descritte al punto precedente. La loro applicazione però rischia di aggiungere eccessiva complessità al prodotto, per questo un design patter deve essere applicato solo se aiuta nella realizzazione del progetto, non perché un alto numero di design pattern equivale ad una buona progettazione. Dunque il loro utilizzo deve essere ragionato e non eccessivo.
- **Documentazione e tracciamento della progettazione:** un prodotto, per essere mantenibile, deve essere documentato anche nella sua struttura. Dunque anche la progettazione deve essere documentata e tracciata attraverso i *Manuale Utente v2.0.0<sub>D</sub>* e *Manuale Sviluppatore v2.0.0<sub>D</sub>*.

### 2.3.5.2 Metriche

- MPR010 Moduli codificati prima della progettazione
- MPR011 Numero di design pattern applicati

### 2.3.5.3 Obiettivi

- **QPR011 Progettare prima di codificare:** ogni modulo dovrebbe essere progettato in relazione agli altri del progetto prima di essere codificato.
- **QPR012 Usare un numero limitato di design pattern:** usare design pattern solo nel caso in cui questi semplifichino la struttura del progetto.

## 2.3.6 PROC006 Codifica

Successivamente alla progettazione avviene la realizzazione del prodotto finale che verrà rilasciato. Il processo di codifica, insieme a quello di verifica, restituisce il prodotto che verrà usato e mantenuto. La codifica di ogni modulo deve trattarsi della “traduzione” in codice della progettazione.

### 2.3.6.1 Funzioni

- **Produrre codice comprensibile:** il prodotto rilasciato in futuro potrà essere usato da altre persone, per questo il codice deve poter essere facilmente comprensibile.
- **Produrre codice mantenibile:** le componenti del prodotto nel tempo potranno risultare obsolete, per questo le nuove tecnologie devono poter essere applicate all’interno del prodotto.
- **Produrre codice estensibile:** il prodotto deve poter concedere la possibilità di ricevere nuove funzionalità in modo semplice e veloce.
- **Produrre codice corretto:** il prodotto, al momento del rilascio, non deve riscontrare anomalie ed essere eseguito correttamente.

### 2.3.6.2 Metriche

- MPR012 Moduli non testati

### 2.3.6.3 Obiettivi

- **QPR013 I moduli possiedono suite di test:** i moduli codificati devono possedere una suite di test per essere verificati in tutte le loro parti e in correlazione tra di loro.
- **QPR014 Rispetto delle norme di codifica:** il codice, per essere meglio comprensibile e corretto deve essere prodotto seguendo delle norme. Il rispetto di tali norme viene verificato nelle *NormeDiProgetto v4.0.0<sub>D</sub>*.

### 2.3.7 PROC007 Validazione

La validazione è l'ultimo processo che precede la consegna ultima del prodotto e il processo di manutenzione. Tale processo è molto simile a quello di verifica, infatti si prefigge scopi simili, solo che consiste nell'essere "l'ultima verifica" prima della consegna del prodotto.

A differenza di un'usuale verifica, la validazione viene effettuata insieme al committente per constatare se questo riceverà il prodotto che si aspettava al momento dell'acquisto. Per questo motivo il processo di validazione è strettamente legato ai requisiti.

#### 2.3.7.1 Funzioni

- **Soddisfacimento di tutti i requisiti:** al momento della validazione il prodotto deve poter soddisfare le richieste del committente. Per fare questo tutti i requisiti estraibili direttamente e indirettamente dal capitolato devono essere soddisfatti.
- **Soddisfacimento dei bisogni del committente:** durante la realizzazione del prodotto finale è probabile che, attraverso vari confronti, siano stati cambiamenti alcuni requisiti del progetto per delle esigenze del committente. Anche tali requisiti devono essere soddisfatti.
- **Svolgimento corretto dell'intero progetto:** trattandosi della verifica finale, nella validazione si effettua un'analisi retrospettiva di tutto il percorso di sviluppo del progetto, se dunque questo è stato sviluppato secondo le norme e i tempi stabiliti.

#### 2.3.7.2 Metriche

- MPR013 Percentuale soddisfacimento metriche

Il processo osserva inoltre altre importanti metriche già presenti in altri processi:

- MPR001
- MPR002
- MPR005
- MPR006
- MPR007

#### 2.3.7.3 Obiettivi

- **QPR015 Soddisfare tutti i requisiti:** al momento della consegna del prodotto tutti i requisiti devono essere soddisfatti.
- **QPR016 Possedere un prodotto sviluppato correttamente:** il prodotto, oltre che soddisfare i requisiti, deve essere stato sviluppato anche nel modo corretto. In questo modo è possibile valutare anche la professionalità del team di sviluppo.

## 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
<b>Descrizione:</b> le scadenze orarie per ogni ruolo riportate nel <i>PianoDiProgetto v4.0.0<sub>D</sub></i> 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 ad 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 nè restrigente.		
QPR002 Variazione del budget	MPR002 Varianza dei costi	0 - 200 €
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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 interezza.		
QPR004 Versionamento	MPR004 Frequenza commit nella repository	25+
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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)
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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	-	-
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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	-	-
<b>Descrizione:</b> 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

## PROC005 Progettazione

Obiettivo	Metrica	Valore desiderato
QPR011 Progettare prima di codificare	MPR010 Moduli codificati prima della progettazione	0% - 25%
<b>Descrizione:</b> i moduli devono essere progettati prima della loro codifica. Purtroppo questo è difficilmente possibile nel periodo che antecede la fase di progettazione e che richiede il rilascio di un PoC <sub>G</sub> , per questo la soglia di tolleranza per moduli codificati prima di essere progettati è al 0% - 25%.		
QPR012 Usare un numero limitato di design pattern	MPR011 Numero di design pattern applicati	10 - 15
<b>Descrizione:</b> data la dimensione del progetto, dai 10 ai 15 design pattern applicati ci sembrano un numero sufficiente per ottenere un codice di qualità non troppo complesso.		

Tabella 5: Obiettivi di qualità per il PROC005

## PROC006 Codifica

Obiettivo	Metrica	Valore desiderato
QPR013 I moduli possiedono suite di test	MPR012 Moduli non testati	0
<b>Descrizione:</b> ogni modulo deve essere testato per verificarne l'integrità e la correttezza.		
QPR014 Rispetto delle norme di codifica	-	-
<b>Descrizione:</b> all'interno del processo di codifica bisogna seguire delle norme per produrre del codice di qualità. Tali norme sono meglio indicate nelle <i>NormeDiProgetto v4.0.0<sub>D</sub></i> .		

Tabella 6: Obiettivi di qualità per il PROC006

## PROC007 Validazione

Obiettivo	Metrica	Valore desiderato
QPR015 Soddisfare tutti i requisiti	-	-
<b>Descrizione:</b> il prodotto al momento della consegna deve poter soddisfare tutti i requisiti richiesti. A tale obiettivo di qualità non sono associate metriche, perché quelle interessate sono già presenti nel processo di analisi. Queste sono MPR005, MPR006 e MPR007.		
QPR016 Possedere un prodotto sviluppato correttamente	MPR013	60%
<b>Descrizione:</b> un prodotto deve anche essere sviluppato correttamente seguendo determinate norme tenendosi ad un certo livello di qualità. Per misurarla si utilizzano opportune metriche che insieme attribuiscono un valore di qualità al prodotto.		

Tabella 7: Obiettivi di qualità per il PROC007

## 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 *NormeDiProgetto v4.0.0<sub>D</sub>*. 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.

##### 3.3.1.1 Metriche

- MPD001 INDICE DI GULPEASE<sub>G</sub>
- MPD002 Correttezza ortografica

### 3.3.1.2 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<sub>G</sub> e le norme dello standard PEP 8<sup>1</sup> per l'analisi statica del codice. Essi ci consentono di definire delle semplici metriche con dei grandi obiettivi da raggiungere nel corso del tempo.

#### 3.3.2.1 Metriche

- MPS001 Presenza di bug
- MPS002 Presenza di vulnerabilità
- MPS003 Presenza di CODE SMELL<sub>G</sub>
- 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
- MPS017 Code coverage
- MPS018 Test di sistema
- MPS019 Test di integrazione
- MPS020 Test di unità

---

<sup>1</sup>Riferirsi alla voce “8. Documentazione PEP8” in §1.5.2



### 3.3.2.2 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.
- **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.
- **QPS006 Massima code coverage:** un ulteriore indice per sapere se un modulo è stato testato correttamente, è vedere quanto codice viene effettivamente eseguito durante la durata dei test.
- **QPS007 Superamento test di sistema:** la qualità di un prodotto di verifica anche in base alla quantità di test di sistema superati.
- **QPS008 Superamento test di integrazione:** la qualità di un prodotto di verifica anche in base alla quantità di test di integrazione superati.
- **QPS009 Superamento test di unità:** la qualità di un prodotto di verifica anche in base alla quantità di test di unità superati.

### 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
<b>Descrizione:</b> 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
<b>Descrizione:</b> il testo non deve presentare alcun errore ortografico.		
QPD003 Organizzazione del documento	-	-
<b>Descrizione:</b> l'organizzazione dei documenti e la loro struttura logica vengono controllate dal Verificatore e dall'Amministratore.		

Tabella 8: Obiettivi di qualità per i documenti

Software		
Obiettivo	Metrica	Valore desiderato
QPS001 Assenza di bug	MPS001 Presenza di bug	0
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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 - 5%
<b>Descrizione:</b> una percentuale tollerabile di linee di codice duplicato non deve eccedere del 20% rispetto al numero totale.		
QPS005 Rispetto delle norme di PEP 8	MPS005 File senza intestazione	0
<b>Descrizione:</b> 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.		
QPS005 Rispetto delle norme di PEP 8	MPS006 Righe non formattate	0
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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 9: Obiettivi di qualità per il software (1)

Software		
Obiettivo	Metrica	Valore desiderato
QPS005 Rispetto delle norme di PEP 8	MPS009 Stringhe non normate	0
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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 10: Obiettivi di qualità per il software (2)

Software		
Obiettivo	Metrica	Valore desiderato
QPS005 Rispetto delle norme di PEP 8	MPS015 Commenti non normati	0
<b>Descrizione:</b> 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
<b>Descrizione:</b> 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.		
QPS006 Massima code coverage	MPS017 Code coverage	80% - 100%
<b>Descrizione:</b> l'esecuzione dei test deve eseguire tutto il codice testato per fare in modo che questo sia analizzato in tutti i suoi cammini di esecuzione.		
QPS007 Superamento test di sistema	MPS018 Test di sistema	100%
<b>Descrizione:</b> i test di sistema eseguiti ad ogni fase di verifica devono essere tutti superati.		
QPS008 Superamento test di integrazione	MPS019 Test di integrazione	100%
<b>Descrizione:</b> i test di integrazione eseguiti ad ogni fase di verifica devono essere tutti superati.		
QPS009 Superamento test di unità	MPS020 Test di unità	100%
<b>Descrizione:</b> i test di unità eseguiti ad ogni fase di verifica devono essere tutti superati.		

Tabella 11: Obiettivi di qualità per il software (3)

### 3.5 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.

#### 3.5.1 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.

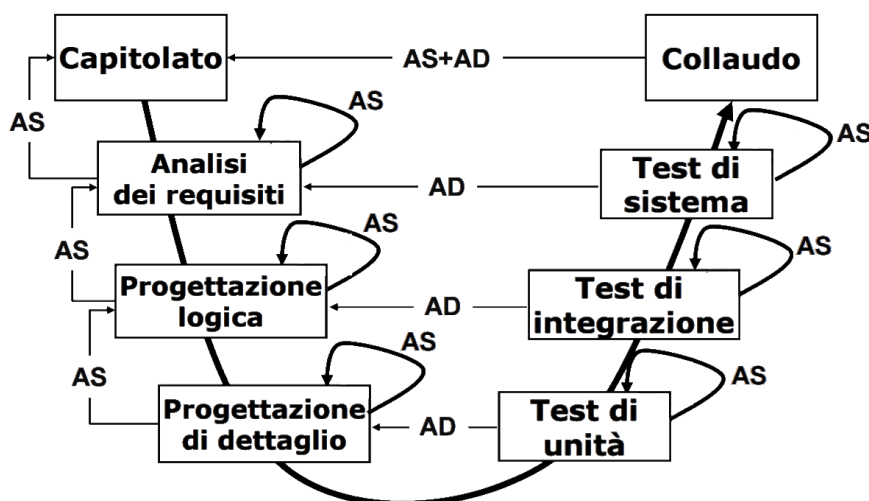


Figura 1: Modello a V<sup>2</sup>

#### 3.5.2 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<sup>3</sup>, 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.

<sup>2</sup>Riferirsi alla voce “6. Modello a V” in §1.5.2

<sup>3</sup>[https://en.wikipedia.org/wiki/V-Model\\_\(software\\_development\)](https://en.wikipedia.org/wiki/V-Model_(software_development))

- **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.

### 3.5.3 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 (figura 1), 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"><li>Viene aperta una issue in Redmine.</li><li>Redmine invia la segnalazione.</li><li>Producer Redmine riceve la segnalazione da Redmine.</li></ol>	I
TV2	Verifica la segnalazione della modifica di una issue in Redmine: <ol style="list-style-type: none"><li>Viene modificata una issue in Redmine.</li><li>Redmine invia la segnalazione.</li><li>Producer Redmine riceve la segnalazione da Redmine.</li></ol>	I
TV3	Verificata la segnalazione del commento di una issue in Redmine: <ol style="list-style-type: none"><li>Viene commentata una issue in Redmine.</li><li>Redmine invia la segnalazione.</li><li>Il Producer Redmine riceve la segnalazione da Redmine.</li></ol>	I
TV4	Verifica la segnalazione dell'apertura di una issue in GitLab: <ol style="list-style-type: none"><li>Viene aperta una issue in GitLab.</li><li>GitLab invia la segnalazione.</li><li>Producer GitLab riceve la segnalazione da GitLab.</li></ol>	I

Tabella 12: Elenco dei test di validazione (1)

Codice	Test	Stato
TV5	Verifica la segnalazione della modifica di una issue in GitLab: <ol style="list-style-type: none"><li>Viene modificata una issue in GitLab.</li><li>GitLab invia la segnalazione.</li><li>Producer GitLab riceve la segnalazione da GitLab.</li></ol>	I
TV6	Verifica la segnalazione del commento di una issue in GitLab: <ol style="list-style-type: none"><li>Viene commentata una issue in GitLab.</li><li>GitLab invia la segnalazione.</li><li>Producer GitLab riceve la segnalazione da GitLab.</li></ol>	I
TV7	Verifica di un evento di push in GitLab: <ol style="list-style-type: none"><li>Viene effettuato un push su GitLab.</li><li>GitLab invia la segnalazione.</li><li>Producer GitLab riceve la segnalazione da GitLab.</li></ol>	I
TV8	Verifica la segnalazione del commento di un commit: <ol style="list-style-type: none"><li>Viene effettuato un commento di un commit su GitLab.</li><li>Viene inviata la segnalazione.</li><li>Producer GitLab riceve la segnalazione da GitLab.</li></ol>	I
TV9	Verifica di invio del messaggio di apertura issue dal Producer Redmine al Gestore Personale: <ol style="list-style-type: none"><li>Producer Redmine elabora il messaggio di apertura issue precedentemente ricevuto.</li><li>Producer Redmine invia il messaggio.</li><li>Gestore personale riceve il messaggio.</li></ol>	I

Tabella 13: Elenco dei test di validazione (2)



Codice	Test	Stato
TV10	Verifica di invio del messaggio di modifica issue dal Producer Redmine al Gestore Personale:  1. Producer Redmine elabora il messaggio di modifica issue precedentemente ricevuto.  2. Producer Redmine invia il messaggio.  3. Gestore personale riceve il messaggio.	I
TV11	Verifica di invio del messaggio di commento issue dal Producer Redmine al Gestore Personale:  1. Producer Redmine elabora il messaggio del commento di issue precedentemente ricevuto.  2. Producer Redmine invia il messaggio.  3. Gestore personale riceve il messaggio.	I
TV12	Verifica scarto di messaggio non valido nel Producer Redmine:  1. Producer Redmine riceve un messaggio che non è in grado di interpretare.  2. Producer Redmine scarta il messaggio.	I
TV13	Verifica di invio del messaggio di commit dal Producer GitLab al Gestore Personale:  1. Producer GitLab elabora il messaggio di commit.  2. Producer GitLab invia il messaggio.  3. Gestore personale riceve il messaggio.	I
TV14	Verifica di invio del messaggio di apertura issue dal Producer GitLab al Gestore Personale:  1. Producer GitLab elabora il messaggio di apertura issue.  2. Producer GitLab invia il messaggio.  3. Gestore Personale riceve il messaggio.	I

Tabella 14: Elenco dei test di validazione (3)

Codice	Test	Stato
TV15	Verifica di invio del messaggio di modifica issue dal Producer GitLab al Gestore Personale:  1. Producer GitLab elabora il messaggio di modifica issue. 2. Producer GitLab invia il messaggio. 3. Gestore personale riceve il messaggio.	I
TV16	Verifica di invio del commento di una issue dal Producer GitLab al Gestore Personale:  1. Producer GitLab elabora il messaggio di commento di issue. 2. Producer GitLab invia il messaggio. 3. Gestore personale riceve il messaggio.	I
TV17	Verifica di invio del messaggio del commento di un commit dal Producer GitLab al Gestore Personale:  1. Producer GitLab elabora il messaggio del commento di commit. 2. Producer GitLab invia il messaggio. 3. Gestore personale riceve il messaggio.	I
TV18	Verifica scarto di messaggio non valido nel Producer GitLab:  1. Producer GitLab riceve un messaggio che non è in grado di interpretare. 2. Producer GitLab scarta il messaggio.	I
TV19	Verifica di invio del messaggio dal Gestore Personale al Consumer Telegram:  1. Gestore Personale riceve un messaggio da un Producer (Redmine o GitLab). 2. Gestore Personale elabora il messaggio ricevuto. 3. Gestore Personale invia il messaggio elaborato. 4. Consumer Telegram riceve il messaggio.	I

Tabella 15: Elenco dei test di validazione (4)

Codice	Test	Stato
TV20	Verifica di invio del messaggio dal Gestore Personale al Consumer Email: <ol style="list-style-type: none"><li>Gestore Personale riceve un messaggio da un Producer (Redmine o GitLab).</li><li>Gestore Personale elabora il messaggio ricevuto.</li><li>Gestore Personale invia il messaggio elaborato.</li><li>Consumer Email riceve il messaggio.</li></ol>	I
TV21	Verifica di invio del messaggio dal Consumer Telegram al bot Telegram: <ol style="list-style-type: none"><li>Consumer Telegram riceve un messaggio dal Gestore Personale.</li><li>Consumer Telegram invia il messaggio.</li><li>Bot Telegram riceve il messaggio.</li></ol>	I
TV22	Verifica di invio del messaggio dal consumer Email al server Email: <ol style="list-style-type: none"><li>Consumer Email riceve un messaggio dal Gestore Personale.</li><li>Consumer Email invia il messaggio.</li><li>Server Email riceve il messaggio.</li></ol>	I
TV23	Verifica di accesso di un utente al sistema tramite Email: <ol style="list-style-type: none"><li>Inserire nel Gestore Personale l'Email dell'utente</li><li>Confermare l'invio dei dati.</li><li>L'accesso al sistema è avvenuto correttamente.</li></ol>	I
TV24	Verifica di accesso di un utente al sistema tramite ID Telegram: <ol style="list-style-type: none"><li>Inserire nel Gestore Personale l'ID Telegram dell'utente</li><li>Confermare l'invio dei dati.</li><li>L'accesso al sistema è avvenuto correttamente.</li></ol>	I

Tabella 16: Elenco dei test di validazione (5)

Codice	Test	Stato
TV25	Verifica di errore per accesso con Email inesistente: <ol style="list-style-type: none"><li>1. Inserire nel Gestore Personale l'Email dell'utente</li><li>2. Confermare l'invio dei dati.</li><li>3. Viene segnalato un errore.</li></ol>	I
TV26	Verifica di errore per accesso con ID Telegram inesistente: <ol style="list-style-type: none"><li>1. Inserire nel Gestore Personale l'ID Telegram dell'utente</li><li>2. Confermare l'invio dei dati.</li><li>3. Viene segnalato un errore.</li></ol>	I
TV27	Verifica di uscita dell'utente dal sistema: <ol style="list-style-type: none"><li>1. L'utente accede al sistema.</li><li>2. L'utente conferma l'uscita dal sistema.</li><li>3. L'utente viene riportato all'accesso al sistema.</li></ol>	I
TV28	Verifica di registrazione di un nuovo utente: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Selezionare l'aggiunta di un nuovo utente.</li><li>3. Inserire i seguenti campi:<ul style="list-style-type: none"><li>• Nome</li><li>• Cognome</li><li>• Contatto Email</li><li>• Contatto Telegram</li></ul></li><li>4. Confermare l'invio dei dati.</li><li>5. L'utente viene registrato correttamente.</li></ol>	I

Tabella 17: Elenco dei test di validazione (6)

Codice	Test	Stato
TV29	Verifica di errore per ID Telegram già esistente: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Selezionare l'aggiunta di un nuovo utente.</li><li>3. Inserire i seguenti campi:<ul style="list-style-type: none"><li>• Nome</li><li>• Cognome</li><li>• Contatto Email</li><li>• Contatto Telegram</li></ul></li><li>4. Confermare l'invio dei dati.</li><li>5. Viene segnalato un errore per aver inserito un ID Telegram già presente nel sistema.</li></ol>	I
TV30	Verifica di errore per Email già esistente: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Selezionare l'aggiunta di un nuovo utente.</li><li>3. Inserire i seguenti campi:<ul style="list-style-type: none"><li>• Nome</li><li>• Cognome</li><li>• Contatto Email</li><li>• Contatto Telegram</li></ul></li><li>4. Confermare l'invio dei dati.</li><li>5. Viene segnalato un errore per aver inserito una Email già esistente.</li></ol>	I
TV31	Verifica di rimozione di un utente dal sistema tramite Email: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Inserire l'Email dell'utente da rimuovere.</li><li>3. Confermare l'invio dei dati.</li><li>4. L'utente è stato correttamente rimosso dal sistema.</li></ol>	I

Tabella 18: Elenco dei test di validazione (7)

Codice	Test	Stato
TV32	Verifica di rimozione di un utente dal sistema tramite ID Telegram: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Inserire l'ID Telegram dell'utente da rimuovere.</li><li>3. Confermare l'invio dei dati.</li><li>4. L'utente è stato correttamente rimosso dal sistema.</li></ol>	I
TV33	Verifica di errore nella rimozione di un utente con ID Telegram inesistente dal sistema: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Inserire l'ID Telegram dell'utente da rimuovere.</li><li>3. Confermare l'invio dei dati.</li><li>4. Viene segnalato un errore per l'ID Telegram non presente nel sistema.</li></ol>	I
TV34	Verifica di errore nella rimozione di un utente con Email inesistente dal sistema: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Inserire l'Email dell'utente da rimuovere.</li><li>3. Confermare l'invio dei dati.</li><li>4. Viene segnalato un errore per la Email non presente nel sistema.</li></ol>	I
TV35	Verifica di rimozione dell'utente acceduto dal sistema tramite Email: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Inserire la propria Email.</li><li>3. Confermare l'invio dei dati.</li><li>4. Si esce dal sistema e si viene riportati alla schermata di accesso.</li></ol>	I

Tabella 19: Elenco dei test di validazione (8)

Codice	Test	Stato
TV36	Verifica di rimozione dell'utente acceduto dal sistema tramite ID Telegram: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Inserire il proprio ID Telegram.</li><li>3. Confermare l'invio dei dati.</li><li>4. Si esce dal sistema e si viene riportati alla schermata di accesso.</li></ol>	I
TV37	Verifica di modifica dei propri dati utente: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Modificare i seguenti campi:<ul style="list-style-type: none"><li>• Nome</li><li>• Cognome</li><li>• Contatto Email</li><li>• Contatto Telegram</li></ul></li><li>3. Confermare l'invio dei dati.</li><li>4. L'utente è stato modificato correttamente.</li></ol>	I
TV38	Verifica di errore nella modifica di un utente con Email già esistente: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Inserire i seguenti campi:<ul style="list-style-type: none"><li>• Nome</li><li>• Cognome</li><li>• Contatto Email già esistente</li><li>• Contatto Telegram</li></ul></li><li>3. Confermare l'invio dei dati.</li><li>4. Viene segnalato un errore per aver inserito una Email già esistente.</li></ol>	I

Tabella 20: Elenco dei test di validazione (9)

Codice	Test	Stato
TV39	Verifica di errore nella modifica di un utente con ID Telegram già esistente: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Inserire i seguenti campi:<ul style="list-style-type: none"><li>• Nome</li><li>• Cognome</li><li>• Contatto Email</li><li>• Contatto Telegram già esistente</li></ul></li><li>3. Confermare l'invio dei dati.</li><li>4. Viene segnalato un errore per aver inserito un ID Telegram già esistente.</li></ol>	I
TV40	Verifica di iscrizione a un nuovo topic: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Selezionare un nuovo topic a cui iscriversi.</li><li>3. Confermare l'invio dei dati.</li><li>4. Il topic è stato correttamente aggiunto alle preferenze dell'utente.</li></ol>	I
TV41	Verifica di aggiunta dei giorni di indisponibilità al calendario: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Selezionare i giorni di indisponibilità da aggiungere.</li><li>3. Confermare l'invio dei dati.</li><li>4. I giorni sono stati aggiunti correttamente alle preferenze dell'utente.</li></ol>	I
TV42	Verifica di aggiunta della piattaforma di messaggistica preferita: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Selezionare la piattaforma preferita per la ricezione dei messaggi da aggiungere.</li><li>3. Confermare l'invio dei dati.</li><li>4. La piattaforma è stata correttamente aggiunta alle preferenze dell'utente.</li></ol>	I

Tabella 21: Elenco dei test di validazione (10)



Codice	Test	Stato
TV43	Verifica di aggiunta delle keyword per i push di GitLab: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Aggiungere le keyword di GitLab.</li><li>3. Confermare l'invio dei dati.</li><li>4. Le keyword di GitLab sono state aggiornate correttamente.</li></ol>	I
TV44	Verifica di errore nell'aggiunta delle keyword per i push di GitLab già esistenti: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Aggiungere delle keyword di GitLab già presenti.</li><li>3. Confermare l'invio dei dati.</li><li>4. Viene segnalato un errore.</li></ol>	I
TV45	Verifica l'aggiunta di progetti di interesse per GitLab o Redmine: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Aggiungere progetto d'interesse di GitLab o Redmine.</li><li>3. Confermare l'invio dei dati.</li><li>4. Il progetto è stato aggiunto correttamente alle preferenze dell'utente.</li></ol>	I
TV46	Verifica l'aggiunta di priorità per i progetti precedentemente selezionati: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Selezionare un progetto precedentemente aggiunto.</li><li>3. Selezionare la priorità desiderata per quel progetto.</li><li>4. Confermare l'invio dei dati.</li><li>5. La priorità relativa al progetto selezionato è stata aggiunta correttamente alle preferenze dell'utente.</li></ol>	I

Tabella 22: Elenco dei test di validazione (11)

Codice	Test	Stato
TV47	Verifica di rimozione di un topic: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Selezionare un topic da rimuovere.</li><li>3. Confermare l'invio dei dati.</li><li>4. Il topic è stato rimosso correttamente dalle preferenze.</li></ol>	I
TV48	Verifica di rimozione dei giorni di indisponibilità dal calendario: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Selezionare i giorni di indisponibilità da rimuovere.</li><li>3. Confermare l'invio dei dati.</li><li>4. I giorni sono stati rimossi dalle preferenze correttamente.</li></ol>	I
TV49	Verifica di rimozione della piattaforma di messaggistica preferita: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Selezionare la piattaforma preferita per la ricezione dei messaggi da rimuovere.</li><li>3. Confermare l'invio dei dati.</li><li>4. La piattaforma è stata correttamente rimossa dalle preferenze.</li></ol>	I
TV50	Verifica di rimozione delle keyword per i push di GitLab: <ol style="list-style-type: none"><li>1. Accedere al sistema.</li><li>2. Aggiungere le keyword di GitLab.</li><li>3. Confermare l'invio dei dati.</li><li>4. Le keyword di GitLab sono state aggiornate correttamente.</li></ol>	I

Tabella 23: Elenco dei test di validazione (12)

Codice	Test	Stato
TV51	Verifica di errore nella rimozione delle keyword inesistenti per i push di GitLab:  1. Accedere al sistema.  2. Aggiungere delle keyword di GitLab non presenti.  3. Confermare l'invio dei dati.  4. Viene segnalato un errore.	I
TV52	Verifica la rimozione di progetti di interesse precedentemente aggiunti:  1. Accedere al sistema.  2. Rimuovere un progetto d'interesse di GitLab o Redmine.  3. Confermare l'invio dei dati.  4. Il progetto è stato rimosso correttamente alle preferenze dell'utente.	I
TV53	Verifica la rimozione di priorità per i progetti precedentemente selezionati:  1. Accedere al sistema.  2. Selezionare un progetto precedentemente aggiunto.  3. Rimuovere la priorità per quel progetto.  4. Confermare l'invio dei dati.  5. La priorità relativa al progetto selezionato è stata rimossa correttamente alle preferenze dell'utente.	I

Tabella 24: Elenco dei test di validazione (13)

**3.5.3.1 Tracciamento**

Codice	Requisito
TV1	R1.1F2
TV2	R1.2F2
TV3	R1.3F2
TV4	R2.1F2
TV5	R2.2F2
TV6	R2.3F2
TV7	R2.4F2
TV8	R2.5F2
TV9	R3.1F2
TV10	R3.2F2
TV11	R3.3F2
TV12	R3F2
TV13	R4.1F2
TV14	R4.2F2
TV15	R4.3F2
TV16	R4.4F2
TV17	R4.5F2
TV18	R4F2
TV19	R5F2
TV20	R6F2
TV21	R7F2
TV22	R8F2
TV23	R9.1F0
TV24	R9.1F0
TV25	R10F0
TV26	R10F0
TV27	R11F0
TV28	R12F0
TV29	R13F0
TV30	R14F0
TV31	R15.1F0

Tabella 25: Elenco dei test in correlazioni con i requisiti (1)

Codice	Requisito
TV32	R15.2F0
TV33	R16F0
TV34	R16F0
TV35	R15F0
TV36	R15F0
TV37	R17F0
TV38	R18F0
TV39	R19F0
TV40	R20.1F0
TV41	R20.2F0
TV42	R20.3F0
TV43	R20.4F0
TV44	R20.5F0
TV45	R20.6F0
TV46	R20.7F0
TV47	R21.1F0
TV48	R21.2F0
TV49	R21.3F0
TV50	R21.4F0
TV51	R21.5F0
TV52	R21.6F0
TV53	R21.7F0

Tabella 26: Elenco dei test in correlazioni con i requisiti (2)

### 3.5.4 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 v3.0.0<sub>D</sub>*.

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

Tabella 27: Elenco dei test di sistema (1)

Codice	Test	Stato
TS16	Verifica di invio del messaggio di commento issue dal Producer GitLab al Gestore Personale	S
TS17	Verifica di invio del messaggio di commento di commit dal Producer GitLab al Gestore Personale	S
TS18	Verifica scarto di messaggio non valido nel Producer GitLab	S
TS19	Verifica di invio del messaggio al Consumer Telegram	S
TS20	Verifica di invio del messaggio al Consumer Email	S
TS21	Verifica di invio del messaggio al bot Telegram	S
TS22	Verifica di invio del messaggio al server Email	S
TS23	Verifica di accesso di un utente al sistema tramite Email	S
TS24	Verifica di accesso di un utente al sistema tramite ID Telegram	S
TS25	Verifica di errore per accesso con Email inesistente	S
TS26	Verifica di errore per accesso con ID Telegram inesistente	S
TS27	Verifica di uscita dell'utente dal sistema	S
TS28	Verifica di aggiunta di un nuovo utente	S
TS29	Verifica di errore per ID Telegram già esistente	S
TS30	Verifica di errore per Email già esistente	S
TS31	Verifica di rimozione utente dal sistema tramite Email	S
TS32	Verifica di rimozione utente dal sistema tramite ID Telegram	S
TS33	Verifica di errore nella rimozione di un utente con ID Telegram inesistente dal sistema	S
TS34	Verifica di errore nella rimozione di un utente con Email inesistente dal sistema	S
TS35	Verifica di rimozione dell'utente acceduto dal sistema tramite Email	S
TS36	Verifica di rimozione dell'utente acceduto dal sistema tramite ID Telegram	S
TS37	Verifica di modifica dei propri dati utente	S
TS38	Verifica di errore nella modifica di un utente con Email già esistente	S
TS39	Verifica di errore nella modifica di un utente con ID Telegram	S
TS40	Verifica di iscrizione a un nuovo Topic	S

Tabella 28: Elenco dei test di sistema (2)

Codice	Test	Stato
TS41	Verifica di aggiunta dei giorni di indisponibilità al calendario	S
TS42	Verifica di aggiunta della piattaforma di messaggistica preferita	S
TS43	Verifica di aggiunta delle keyword per i push di GitLab	S
TS44	Verifica di errore nell'aggiunta delle keyword per i push di GitLab già esistenti	S
TS45	Verifica di aggiunta progetti d'interesse per GitLab o Redmine	S
TS46	Verifica di aggiunta delle priorità per i progetti d'interesse	S
TS47	Verifica di rimozione di un Topic	S
TS48	Verifica di rimozione dei giorni di indisponibilità dal calendario	S
TS49	Verifica di rimozione della piattaforma di messaggistica preferita	S
TS50	Verifica di rimozione delle keyword per i push di GitLab	S
TS51	Verifica di errore nella rimozione delle keyword inesistenti per i push di GitLab	S
TS52	Verifica di rimozione di progetti d'interesse	S
TS53	Verifica di rimozione della priorità per i progetti d'interesse selezionati	S

Tabella 29: Elenco dei test di sistema (3)



**3.5.4.1 Tracciamento**

Codice	Requisito
TS1	R1.1F2
TS2	R1.2F2
TS3	R1.3F2
TS4	R2.1F2
TS5	R2.2F2
TS6	R2.3F2
TS7	R2.4F2
TS8	R2.5F2
TS9	R3.1F2
TS10	R3.2F2
TS11	R3.3F2
TS12	R3F2
TS13	R4.1F2
TS14	R4.2F2
TS15	R4.3F2
TS16	R4.4F2
TS17	R4.5F2
TS18	R4F2
TS19	R5F2
TS20	R6F2
TS21	R7F2
TS22	R8F2
TS23	R9.1F0
TS24	R9.1F0
TS25	R10F0
TS26	R10F0
TS27	R11F0
TS28	R12F0
TS29	R13F0
TS30	R14F0
TS31	R15.1F0

Tabella 30: Elenco dei test in correlazioni con i requisiti (1)

Codice	Requisito
TS32	R15.2F0
TS33	R16F0
TS34	R16F0
TS35	R15F0
TS36	R15F0
TS37	R17F0
TS38	R18F0
TS39	R19F0
TS40	R20.1F0
TS41	R20.2F0
TS42	R20.3F0
TS43	R20.4F0
TS44	R20.5F0
TS45	R20.6F0
TS46	R20.7F0
TS47	R21.1F0
TS48	R21.2F0
TS49	R21.3F0
TS50	R21.4F0
TS51	R21.5F0
TS52	R21.6F0
TS53	R21.7F0

Tabella 31: Elenco dei test in correlazioni con i requisiti (2)

### 3.5.5 Test d'integrazione

Codice	Test	Stato
TI1	Verifica l'integrazione di WebhookFactory con Webhook	S
TI1.1	Verifica l'integrazione di GitlabWebhookFactory con GitlabIssueWebhook	S
TI1.2	Verifica l'integrazione di GitlabWebhookFactory con GitlabPushWebhook	S
TI1.3	Verifica l'integrazione di GitlabWebhookFactory con GitlabCommitCommentWebhook	S
TI1.4	Verifica l'integrazione di GitlabWebhookFactory con GitlabIssueCommentWebhook	S
TI1.5	Verifica l'integrazione di RedmineWebhookFactory con RedmineIssueWebhook	S
TI2	Verifica l'integrazione tra Webhook e GitlabProducer	S
TI2.1	Verifica l'integrazione tra GitlabIssueWebhook e GitlabProducer	S
TI2.2	Verifica l'integrazione tra GitlabPushWebhook e GitlabProducer	S
TI2.3	Verifica l'integrazione tra GitlabIssueCommentWebhook e GitlabProducer	S
TI2.4	Verifica l'integrazione tra GitlabCommitCommentWebhook e GitlabProducer	S
TI3	Verifica l'integrazione tra Webhook e RedmineProducer	S
TI3.1	Verifica l'integrazione tra RedmineIssueWebhook e RedmineProducer	S
TI4	Verifica l'integrazione del EmailConsumer con SMTP	S
TI5	Verifica l'integrazione del TelegramConsumer con la libreria requests	S
TI6	Verifica l'integrazione di KafkaProducer con le componenti Producer	S
TI6.1	Verifica l'integrazione di KafkaProducer con GitlabProducer	S
TI6.2	Verifica l'integrazione di KafkaProducer con RedmineProducer	S
TI7	Verifica l'integrazione di KafkaConsumer con le componenti Consumer	S
TI7.1	Verifica l'integrazione di KafkaConsumer nel TelegramConsumer	S
TI7.2	Verifica l'integrazione di KafkaConsumer nel EmailConsumer	S
TI8	Verifica l'integrazione di KafkaProducer con ClientGestorePersonale	S
TI9	Verifica l'integrazione di KafkaConsumer con ClientGestorePersonale	S

Tabella 32: Elenco dei test d'integrazione (1)

Codice	Test	Stato
TI10	Verifica l'integrazione di Flask con FlaskServer	S
TI11	Verifica l'integrazione di FlaskServer con GitlabProducer	S
TI12	Verifica l'integrazione di FlaskServer con RedmineProducer	S
TI13	Verifica l'integrazione di KafkaProducer con KafkaProducerCreator	S
TI14	Verifica l'integrazione di KafkaConsumer con KafkaConsumerCreator	S
TI15	Verifica l'integrazione del database MongoClient con ControllerGestorePersonale	S
TI16	Verifica l'integrazione del database MongoClient con la componente di processamento dei messaggi MessageProcessor	S
TI17	Verifica l'integrazione di MongoClient con MongoSingleton	S
TI18	Verifica l'integrazione di MongoFacade con MessageProcessor	S
TI19	Verifica l'integrazione di ControllerGestorePersonale con Resource di Flask	S
TI20	Verifica l'integrazione di MongoFacade con MongoProject	S
TI21	Verifica l'integrazione di MongoFacade con MongoUser	S

Tabella 33: Elenco dei test d'integrazione (2)

### 3.5.5.1 Tracciamento

Codice	Componente
TI1	<code>webhook.WebhookFactory.create_webhook()</code>
TI1.1	<code>webhook.gitlab.GitlabWebhookFactory.create_webhook()</code>
TI1.2	<code>webhook.gitlab.GitlabWebhookFactory.create_webhook()</code>
TI1.3	<code>webhook.gitlab.GitlabWebhookFactory.create_webhook()</code>
TI1.4	<code>webhook.gitlab.GitlabWebhookFactory.create_webhook()</code>
TI1.5	<code>webhook.redmine.RedmineWebhookFactory.create_webhook()</code>
TI2	<code>producer.gitlab.GitlabProducer.produce()</code>
TI2.1	<code>producer.gitlab.GitlabProducer.produce()</code>
TI2.2	<code>producer.gitlab.GitlabProducer.produce()</code>
TI2.3	<code>producer.gitlab.GitlabProducer.produce()</code>
TI2.4	<code>producer.gitlab.GitlabProducer.produce()</code>
TI3	<code>producer.redmine.RedmineProducer.produce()</code>
TI3.1	<code>producer.redmine.GitlabProducer.produce()</code>
TI4	<code>consumer.email.EmailConsumer.send()</code>
TI5	<code>consumer.email.EmailConsumer.send()</code>
TI6	<code>producer.Producer.produce()</code>
TI6.1	<code>producer.gitlab.GitlabProducer.produce()</code>
TI6.2	<code>producer.redmine.RedmineProducer.produce()</code>

Tabella 34: Tracciamento dei test in correlazione con le componenti (1)

Codice	Componente
TI7	<code>consumer.Producer.listen()</code>
TI7.1	<code>consumer.telegram.TelegramConsumer.listen()</code>
TI7.2	<code>consumer.email.EmailConsumer.listen()</code>
TI8	<code>consumer.email.EmailConsumer.send()</code>
TI9	<code>gestore_personale.client.ClientGP.read_messages()</code>
TI10	<code>gestore_personale.client.ClientGP.send_all()</code>
TI11	<code>producer.Server.run()</code>
TI12	<code>producer.Server.run()</code>
TI13	<code>producer.creator.KafkaProducerCreator.create()</code>
TI14	<code>consumer.creator.KafkaConsumerCreator.create()</code>
TI15	<code>gestore_personale.Controller.access()</code>
TI16	<code>gestore_personale.processor.GitlabProcessor._filter_users_by_topic()</code>
TI17	<code>mongo_db.MongoSingleton.instance()</code>
TI18	<code>gestore_personale.processor.GitlabProcessor._check_labels()</code>
TI19	<code>gestore_personale.Controller.access()</code>
TI20	<code>mongo_db.MongoSingleton.instance().read()</code>
TI21	<code>mongo_db.MongoSingleton.instance().read()</code>

Tabella 35: Tracciamento dei test in correlazione con le componenti (2)

### 3.5.6 Test di unità

Codice	Test	Stato
TU1	Verifica il processamento del webhook e se è malformato	S
TU2	Verifica l'avviamento di FlaskServer con le configurazioni date	S
TU3	Verifica la creazione di un oggetto KafkaProducer	S
TU4	Verifica il tipo di segnalazione ricevuta da GitlabProducer	S
TU5	Verifica l'invio del messaggio proveniente da Gitlab a KafkaProducer	S
TU6	Verifica il tipo di segnalazione ricevuta da RedmineProducer	S
TU7	Verifica l'invio del messaggio proveniente da Redmine a KafkaProducer	S
TU8	Verifica la creazione di un oggetto KafkaConsumer	S
TU9	Verifica l'invio del messaggio finale a Telegram tramite bot	S
TU10	Verifica la formattazione del messaggio Telegram da un oggetto JSON	S
TU11	Verifica che TelegramConsumer sia in ascolto di messaggi provenienti da KafkaConsumer	S
TU12	Verifica l'invio del messaggio finale tramite server mail	S
TU13	Verifica la formattazione della mail da un oggetto JSON	S
TU14	Verifica che EmailConsumer sia in ascolto di messaggi provenienti da KafkaConsumer	S
TU15	Verifica la formattazione della mail in formato HTML	S
TU16	Verifica l'inserimento di un utente nel database	S
TU17	Verifica la presenza di un utente nel database	S
TU18	Verifica l'identificativo di un utente nel database	S
TU19	Verifica la rimozione di un utente dal database	S
TU20	Verifica l'aggiornamento del nome di un utente nel database	S
TU21	Verifica l'aggiornamento del cognome di un utente nel database	S
TU22	Verifica l'aggiornamento di un identificativo telegram di un utente nel database	S
TU23	Verifica l'aggiornamento di una mail di un utente nel database	S

Tabella 36: Elenco dei test di unità (1)

Codice	Test	Stato
TU24	Verifica l'aggiornamento di una preferenza per l'applicazione di messaggistica dell'utente nel database	S
TU25	Verifica l'aggiunta di alcune keywords alle preferenze dell'utente nel database	S
TU26	Verifica la presenza di keywords nelle preferenze dell'utente nel database	S
TU27	Verifica l'aggiunta di alcune labels alle preferenze dell'utente nel database	S
TU28	Verifica la presenza di labels nelle preferenze dell'utente nel database	S
TU29	Verifica la presenza di utenti con una determinata priorità relativa ad un progetto nel database	S
TU30	Verifica la presenza di utenti disponibili nel database	S
TU31	Verifica la presenza di utenti con priorità più alta relativa ad un progetto nel database	S
TU32	Verifica l'identificativo Telegram di un utente nel database	S
TU33	Verifica l'email di un utente nel database	S
TU34	Verifica la corrispondenza del commit dato con le keywords presenti nel database	S
TU35	Verifica la corrispondenza delle labels date con le labels presenti nel database	S
TU36	Verifica l'inserimento di un progetto nel database	S
TU37	Verifica la presenza di un progetto nel database	S
TU38	Verifica la url di un progetto nel database	S
TU39	Verifica la rimozione di un progetto dal database	S
TU40	Verifica le keyword legate al progetto dato nel database	S
TU41	Verifica le labels legate al progetto dato nel database	S
TU42	Verifica l'inserimento di una nuova keyword legata al progetto dato, nel database	S
TU43	Verifica l'inserimento di una nuova labe,legata al progetto dato, nel database	S
TU44	Verifica l'ascolto di messaggi provenienti da Kafka da parte del ClientGestorePersonale	S
TU45	Verifica l'elaborazione del messaggio da inviare a Kafka	S
TU46	Verifica l'invio del messaggio processato su Kafka per tutti i destinatari	S

Tabella 37: Elenco dei test di unità (2)



Codice	Test	Stato
TU47	Verifica la preparazione del messaggio da dare al ClientGestorePersonale	S
TU48	Verifica il progetto a cui appartiene il messaggio dato	S
TU49	Verifica gli utenti coinvolti nel progetto dato	S
TU50	Verifica gli utenti con priorità maggiore per un determinato progetto da una lista di utenti dati	S
TU51	Verifica i contatti Telegram degli utenti dati	S
TU52	Verifica i contatti email degli utenti dati	S

Tabella 38: Elenco dei test di unità (3)

### 3.5.6.1 Tracciamento

Codice	Componente
TU1	FlaskServer._webhook_handler()
TU2	FlaskServer.run()
TU3	KafkaProducerCreator.create()
TU4	GitlabProducer.webhook_kind()
TU5	GitLabProducer._produce()
TU6	RedmineProducer.webhook_kind()
TU7	RedmineProducer._produce()
TU8	KafkaConsumerCreator.create()
TU9	TelegramConsumer.send()
TU10	TelgramConsumer.format()
TU11	TelgramConsumer.listen()
TU12	EmailConsumer.send()
TU13	EmailConsumer.format()
TU14	EmailConsumer.listen()
TU15	EmailConsumer.format_html()
TU16	MongoUsers.create()
TU17	MongoUsers.exists()
TU18	MongoUsers.read()
TU19	MongoUsers.delete()
TU20	MongoUsers.update_name()
TU21	MongoUsers.update_surname()
TU22	MongoUsers.update_telegram()
TU23	MongoUsers.update_email()
TU24	MongoUsers.update_user_preference()
TU25	MongoUsers.add_keywords()
TU26	MongoUsers.user_keywords()
TU27	MongoUsers.add_labels()
TU28	MongoUsers.user_labels()
TU29	MongoUsers._get_users_by_priority()
TU30	MongoUsers._get_users_available()

Tabella 39: Elenco dei test in correlazione con le componenti (1)

TU31	MongoUsers._get_users_max_priority()
TU32	MongoUsers._get_user_telegram()
TU33	MongoUsers._get_user_email()
TU34	MongoUsers._get_match_keywords()
TU35	MongoUsers._get_match_labels()
TU36	MongoProjects.create()
TU37	MongoProjects.exists()
TU38	MongoProjects.read()
TU39	MongoProjects.delete()
TU40	MongoProjects.keywords()
TU41	MongoProjects.labels()
TU42	MongoProjects.insert_keyword_by_project()
TU43	MongoProjects.insert_label_by_project()
TU44	ClientGestorePersonale.read_message()
TU45	ClientGestorePersonale.process()
TU46	ClientGestorePersonale.send_all()
TU47	Processor.prepare_message()
TU48	Processor._check_project()
TU49	Processor.get_involved_users()
TU50	Processor.filter_users_with_max_priority()
TU51	Processor.get_telegram_contacts()
TU52	Processor.get_email_contacts()

Tabella 40: Elenco dei test in correlazione con le componenti (2)

## A 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 §??.

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.

### A.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.

### A.2 Primo periodo (RR)

Periodo individuato come *AnalisiDeiRequisiti v3.0.0<sub>D</sub>* che dura circa un mese di tempo e un quarto del tempo complessivo per realizzare il progetto.

#### A.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 v4.0.0<sub>D</sub>*, 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.

#### A.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 §A.1.

### A.2.2.1 Documenti

**QPD001 Leggibilità del testo**      **MPD001 Indice Gulpease**      **50 - 60**

Prodotto/processo testato	Risultato ottenuto	Valutazione
<i>NormeDiProgetto v1.0.0<sub>D</sub></i>	54.57	Soddisfacente
<i>StudioDiFattibilità v1.0.0<sub>D</sub></i>	53.7	Soddisfacente
<i>PianoDiProgetto v1.0.0<sub>D</sub></i>	51.75	Soddisfacente
<i>PianoDiQualifica v1.0.0<sub>D</sub></i>	53.29	Soddisfacente
<i>AnalisiDeiRequisiti v1.0.0<sub>D</sub></i>	55.62	Soddisfacente

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

Tabella 41: Risultati di MPD001 Indice Gulpease

**QPD002 Correttezza ortografica**      **MPD002 Correttezza ortografica**      **0**

Prodotto/processo testato	Risultato ottenuto	Valutazione
<i>NormeDiProgetto v1.0.0<sub>D</sub></i>	0	Soddisfacente
<i>StudioDiFattibilità v1.0.0<sub>D</sub></i>	0	Soddisfacente
<i>PianoDiProgetto v1.0.0<sub>D</sub></i>	0	Soddisfacente
<i>PianoDiQualifica v1.0.0<sub>D</sub></i>	0	Soddisfacente
<i>AnalisiDeiRequisiti v1.0.0<sub>D</sub></i>	0	Soddisfacente

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

Tabella 42: Risultati di MPD002 Correttezza ortografica

#### A.2.2.2 Processi

<b>QPR001 Rispetto dei periodi della pianificazione</b>	<b>MPR001 Varianza della pianificazione</b>	<b>96 ore</b>
Prodotto/processo testato	Risultato ottenuto	Valutazione
<i>PianoDiProgetto v4.0.0<sub>D</sub></i>	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 43: Risultati di MPR001 Varianza della pianificazione

<b>QPR002 Variazione del budget</b>	<b>MPR002 Varianza dei costi</b>	<b>0 - 200 €</b>
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 44: Risultati di MPR002 Varianza dei costi

<b>QPR003 Rispetto delle fasi del ciclo di vita</b>	<b>MPR003 Aderenza agli standard</b>	<b>Livello di maturità: 3 Valutazione attributi: L</b>
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 45: Risultati di MPR003 Aderenza agli standard

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

**Nota:** il numero di commit effettuati risulta ottimo.

Tabella 46: Risultati di MPR004 Frequenza commit nella repository

<b>QPR009 Effettuare una verifica costante</b>	<b>MPR009 Frequenza controllo prodotti</b>	<b>5 Modifiche</b>
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 47: Risultati di MPR009 Frequenza controllo prodotti

### A.2.3 Considerazioni sui risultati e sull'esito della revisione

I risultati ottenuti dall'esito della revisione (24/30) sono stati poco soddisfacenti a livello personale, mentre le correzioni attraverso colloqui e commenti si sono rivelate ma molto esplicative per le correzioni da apportare.

Quest'ultimo aspetto ci fatto rendere conto degli errori e delle imperfezioni fatte, e come poter ovviare a questi in modo da produrre un prodotto migliore con documentazione allegata adeguata.

Per arrivare a questo miglioramento abbiamo deciso di dare più importanza alla fase di verifica per i prossimi periodi, perché visti i risultati l'abbiamo ritenuta poco soddisfacente.

## A.3 Secondo periodo (RP)

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

### A.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<sub>G</sub>.

### A.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 §A.1.
- **Considerazioni:** nostro commento in relazione ai risultati ottenuti dalla misurazione.

### A.3.2.1 Documenti

#### A.3.2.1.1 MPD001 Indice di Gulpease

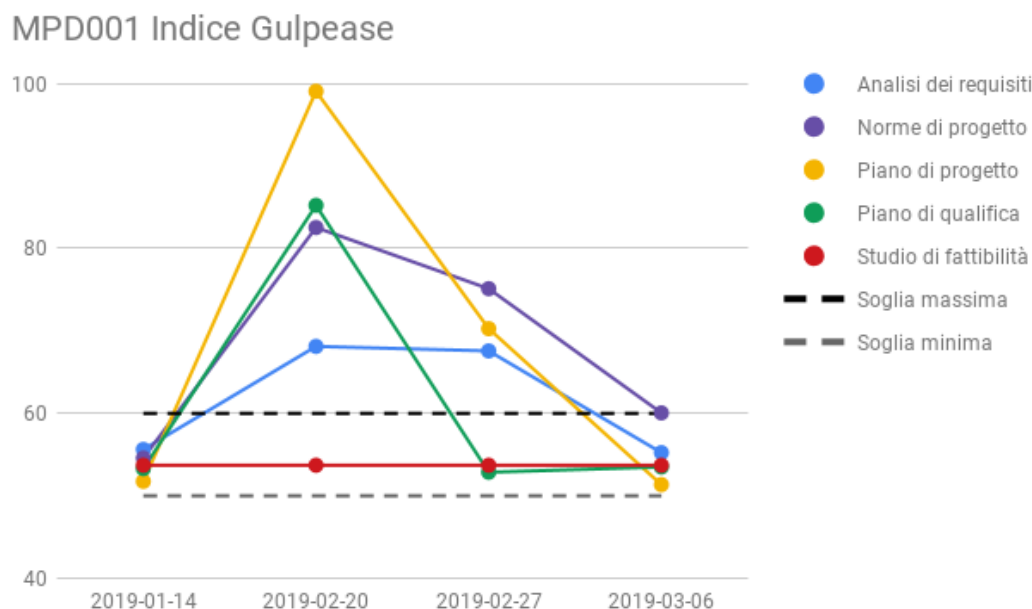


Figura 2: Diagramma con valori misurati tramite MPD001 Indice di Gulpease

- **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'*AnalisiDeiRequisiti v3.0.0<sub>D</sub>*, rispettano i valori desiderati.



## A.3.2.1.2 MPD002 Correttezza ortografica

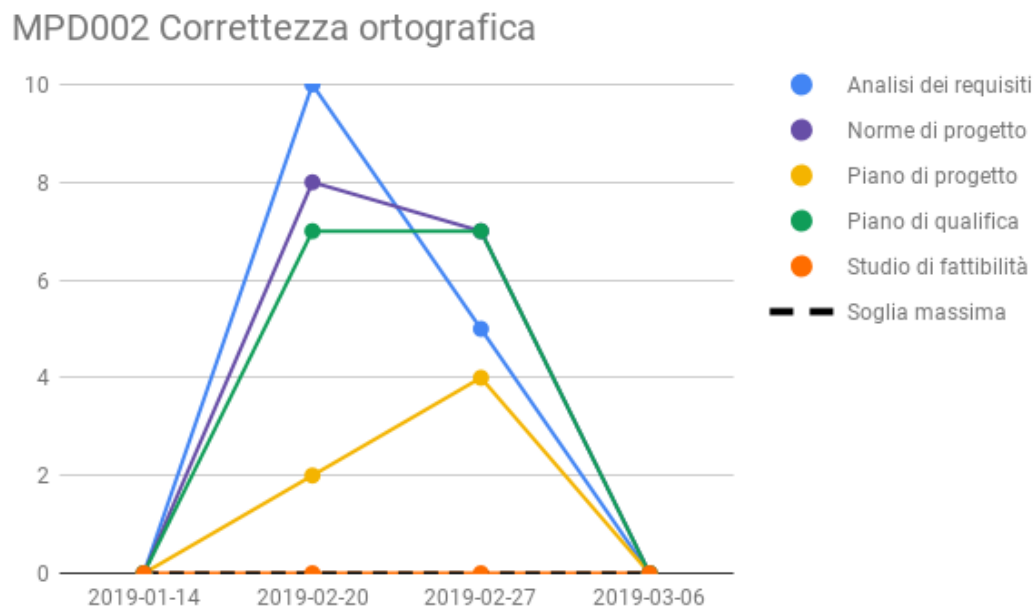


Figura 3: Diagramma con valori misurati tramite MPD002 Correttezza ortografica

- **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.

### A.3.2.2 Processi

#### A.3.2.2.1 MPR001 Varianza della pianificazione

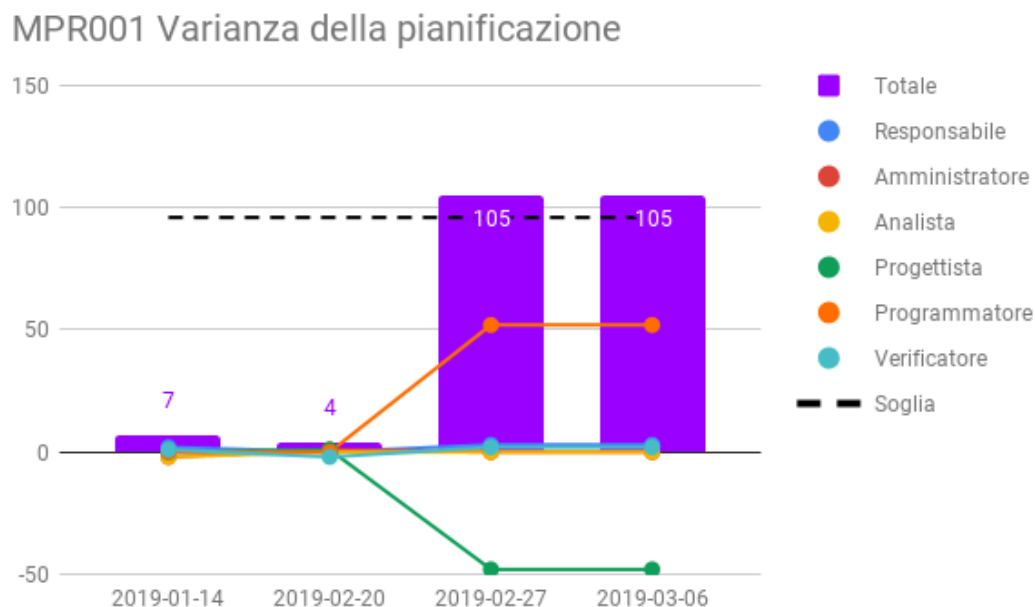


Figura 4: Diagramma con valori misurati tramite MPR001 Varianza della pianificazione

- **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.

### A.3.2.2.2 MPR002 Varianza dei costi

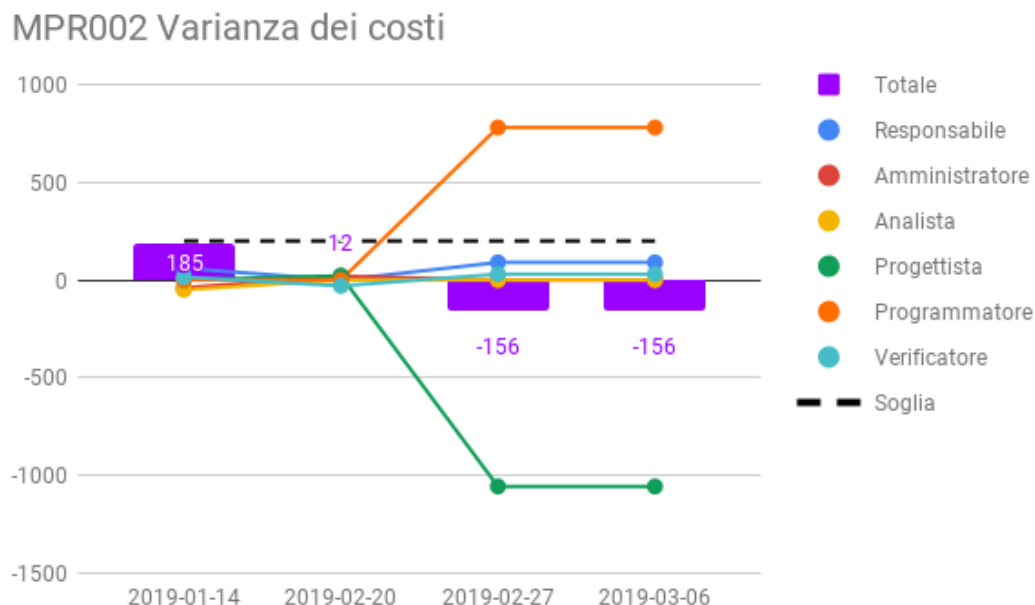


Figura 5: Diagramma con valori misurati tramite MPR002 Varianza dei costi

- **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.

### A.3.2.2.3 MPR003 Aderenza agli standard

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

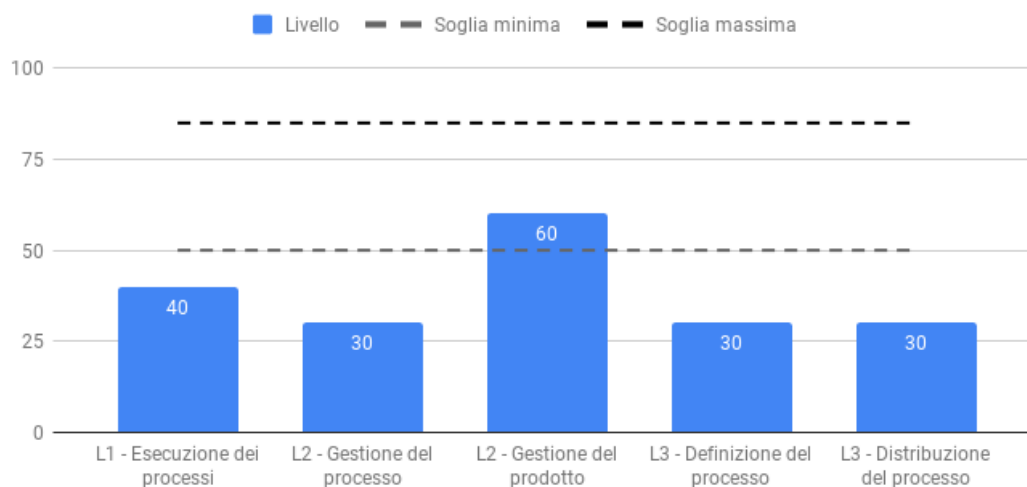


Figura 6: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (1)

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

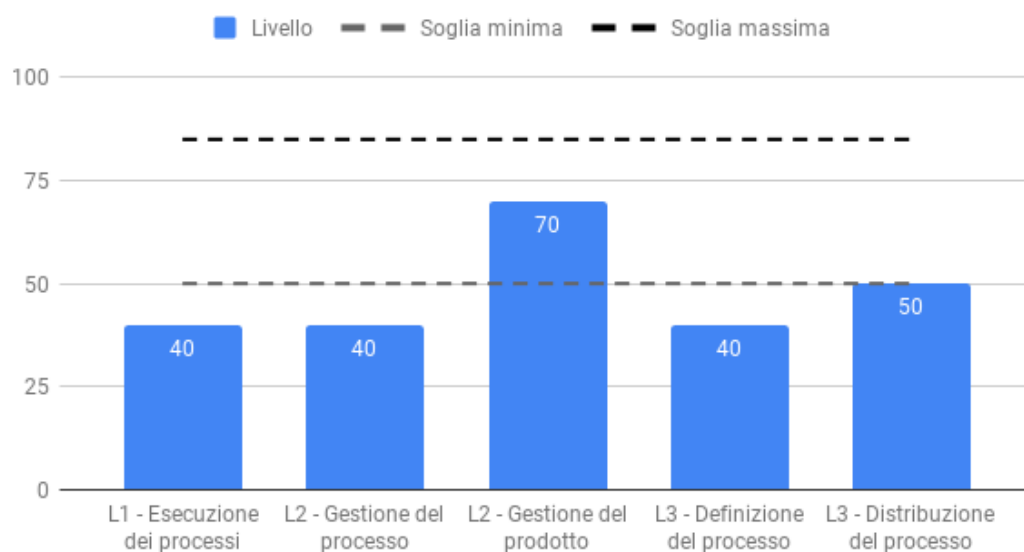


Figura 7: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (2)

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

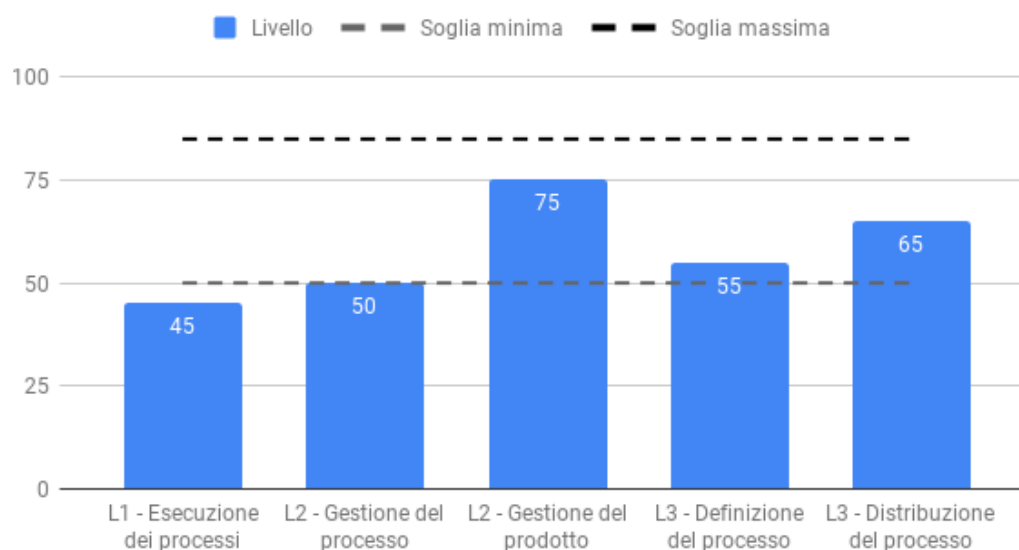


Figura 8: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (3)

- **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.

### A.3.2.3 Software

#### A.3.2.3.1 MPS001 Presenza di bug



Figura 9: Diagramma con valori misurati tramite MPS001 Presenza di bug

- **Obiettivo:** QPS001 Assenza di bug.
- **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.

### A.3.2.3.2 MPS002 Presenza di vulnerabilità



Figura 10: Diagramma con valori misurati tramite MPS002 Presenza di vulnerabilità

- **Obiettivo:** QPS002 Assenza di vulnerabilità.
- **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.

### A.3.2.3.3 MPS003 Presenza di code smell

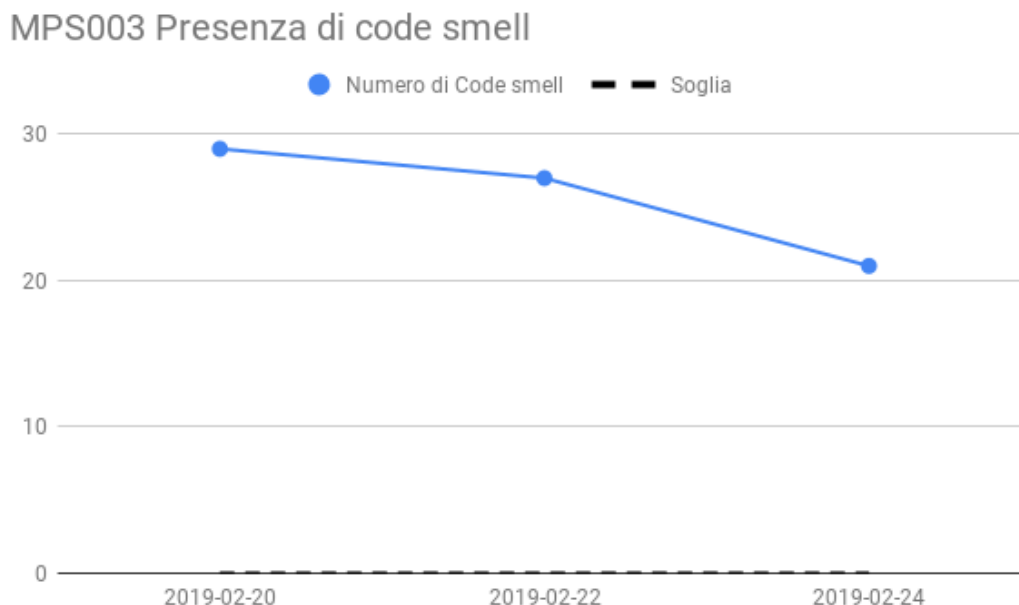


Figura 11: Diagramma con valori misurati tramite MPS003 Presenza di code smell

- **Obiettivo:** QPS003 Assenza di code smell.
- **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.



#### A.3.2.3.4 MPS004 Duplicazione del codice

##### MPS004 Duplicazione del codice

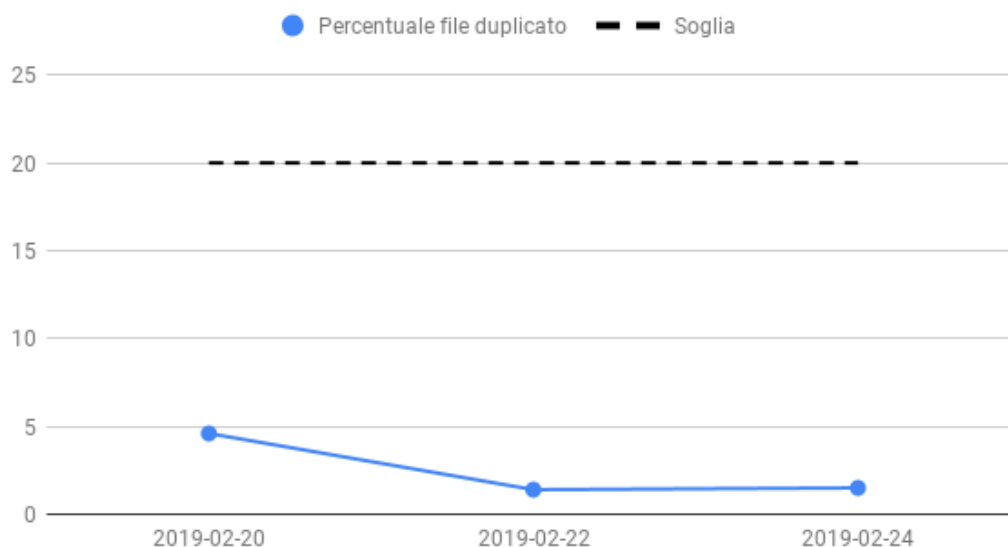


Figura 12: Diagramma con valori misurati tramite MPS004 Duplicazione del codice

- **Obiettivo:** QPS004 Minima duplicazione del codice.
- **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%.

#### A.3.3 Considerazioni sui risultati e sull'esito della revisione

L'esito della revisione (27/30) ci ha dimostrato l'effettivo miglioramento della qualità del lavoro effettuato dopo la Revisione dei Requisiti grazie alla comprensione degli errori commessi in precedenza.

Tuttavia anche questa volta abbiamo deciso di dover dare più importanza alla parte di verifica per impedire che non ci siano errori (logici e ortografici), e per fare in modo che tutti i membri siano a conoscenza del contenuto dei prodotti.

##### A.3.3.1 Considerazioni Proof of Concept (PoC)

Durante questo periodo è stato effettuato il colloquio del Proof of Concept con il Prof. Riccardo Cardin in data 26 Febbraio 2019.

Questa presentazione, effettuata con l'ausilio di slide per la parte introduttiva, è stata ritenuta poco soddisfacente in quanto, come ci è stato fatto notare dal Prof. Riccardo Cardin, non abbiamo dato la giusta importanza al ruolo del broker Apache Kafka.

Nonostante questo abbiamo raggiunto gli obiettivi prefissati e raffinato i casi d'uso, relativi al nostro progetto, che precedentemente non avevamo compreso completamente.

## A.4 Terzo periodo (RQ)

Periodo individuato come Revisione di qualifica, la quale dura all'incirca un mese dell'intera durata del progetto. L'obiettivo di questo periodo è quello di misurare il conseguimento di una solida base architettuale.

### A.4.1 Riassunto delle attività di verifica

In questo periodo di tempo abbiamo soprattutto attuato verifiche sul codice, a differenza del precedente periodo di Revisione di Progettazione, che era stata investita più nei documenti. Le attività riguardanti i documenti sono state posticipate nei giorni successivi al colloquio con il Prof. Riccardo Cardin riguardante la PRODUCT BASELINE<sub>G</sub>.

### A.4.2 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 2019-03-08) 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 §A.1.
- **Considerazioni:** nostro commento in relazione ai risultati ottenuti dalla misurazione.

#### A.4.2.1 Documenti

##### A.4.2.1.1 MPD001 Indice di Gulpease

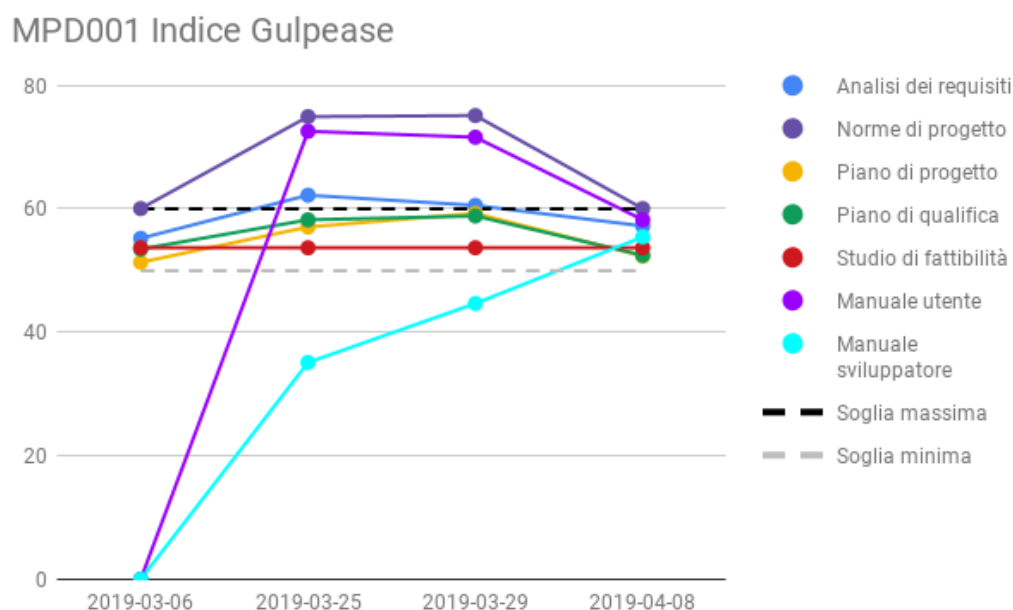


Figura 13: Diagramma con valori misurati tramite MPD001 Indice di Gulpease

- **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 rispettano i valori desiderati. Studiando il diagramma si nota che, a differenza del precedente periodo di Revisione di Progettazione, i valori ottenuti dalle verifiche non variano di molto durante la stesura. Questo è una conseguenza del fatto che i documenti non hanno subito radicali modifiche.

#### A.4.2.1.2 MPD002 Correttezza ortografica

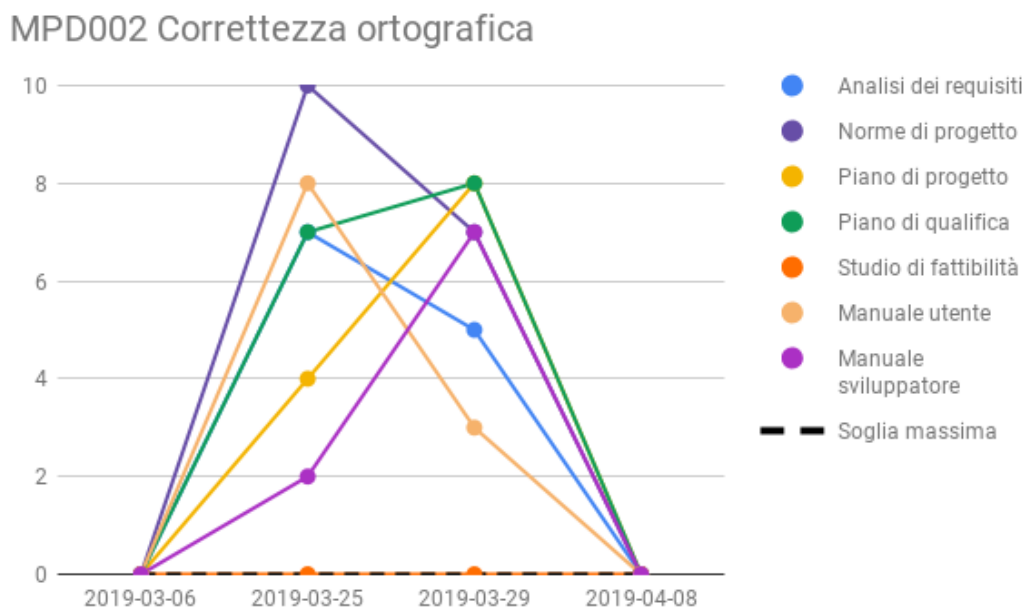


Figura 14: Diagramma con valori misurati tramite MPD002 Correttezza ortografica

- **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:** proseguendo con le varie verifiche notiamo un generale miglioramento con una costante diminuzione degli errori grammaticali.

#### A.4.2.2 Processi

##### A.4.2.2.1 MPR001 Varianza della pianificazione

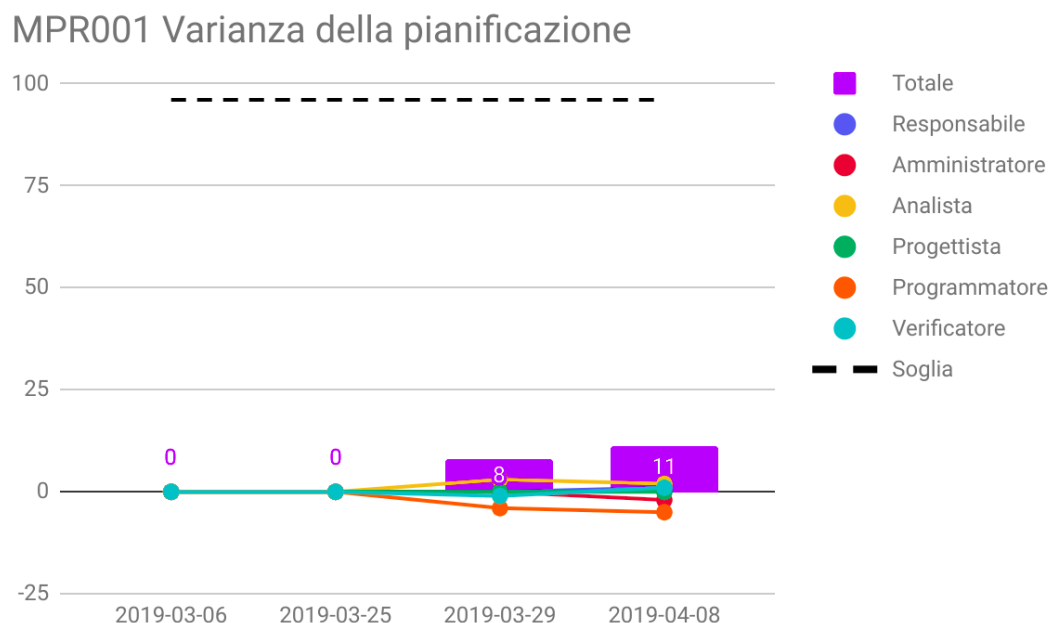


Figura 15: Diagramma con valori misurati tramite MPR001 Varianza della pianificazione

- **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:** soddisfacente.
- **Considerazioni:** in questo periodo abbiamo dovuto ridurre le ore al Programmatore e al Verificatore per darne all'Analista avendo bisogno di alcune ore per fare delle piccole correzioni ad *AnalisiDeiRequisiti v3.0.0<sub>D</sub>*.

#### A.4.2.2.2 MPR002 Varianza dei costi

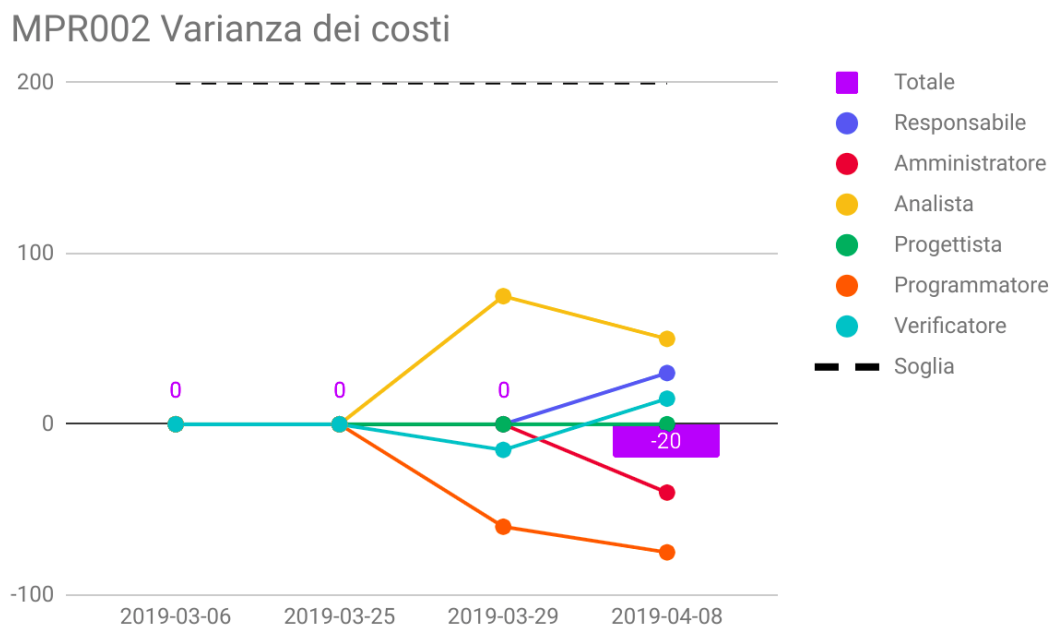


Figura 16: Diagramma con valori misurati tramite MPR002 Varianza dei costi

- **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 Programmatore e al Verificatore a favore di quelle per l'Analista non ha portato nessun cambiamento del preventivo. Un dato che risulta essere soddisfacente.

#### A.4.2.2.3 MPR003 Aderenza agli standard

MPR003 Aderenza agli standard (del 2019-03-25)

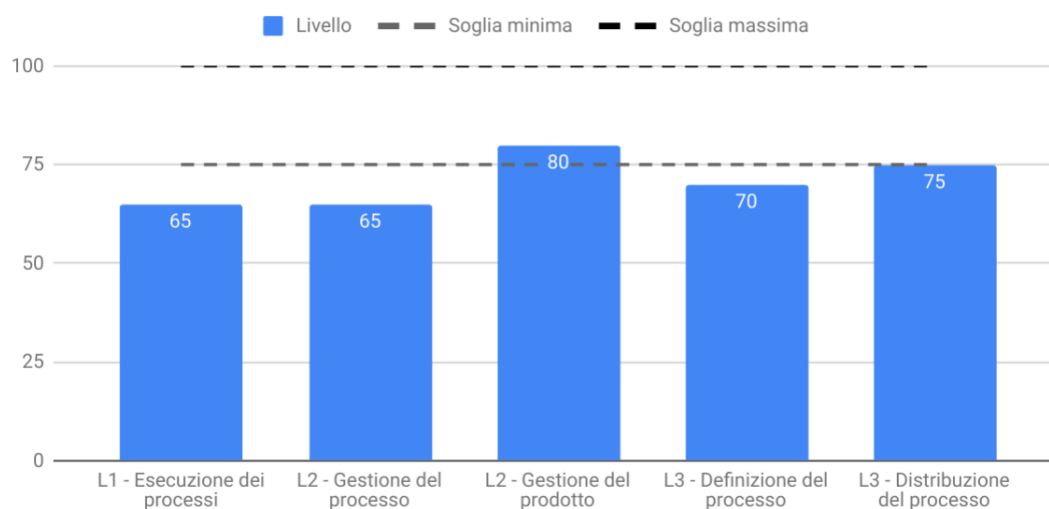


Figura 17: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (1)

MPR003 Aderenza agli standard (del 2019-02-29)

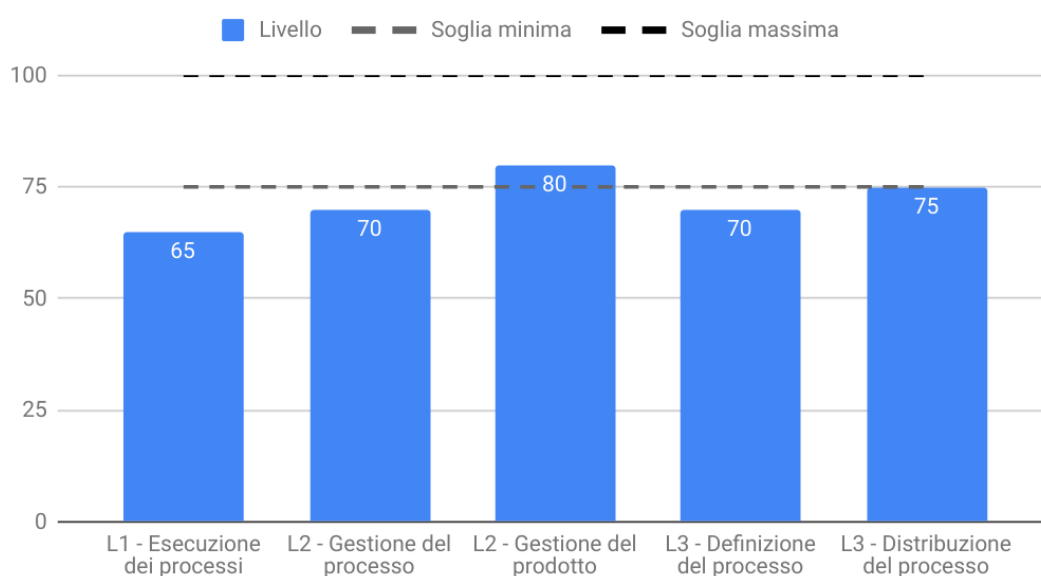


Figura 18: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (2)

## MPR003 Aderenza agli standard (del 2019-04-08)

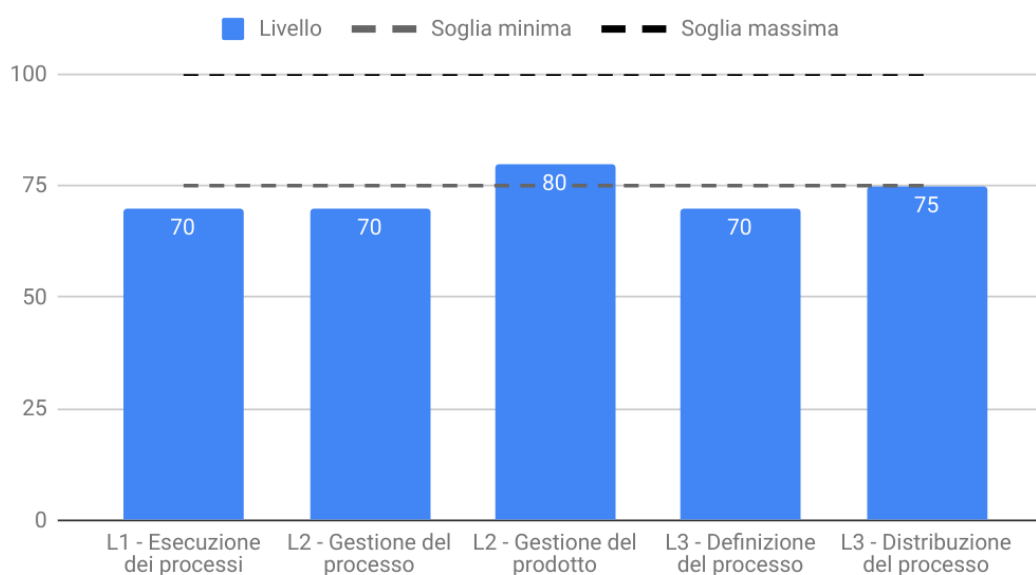


Figura 19: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (3)

- **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. Rispetto alla revisione precedente si possono notare dei notevoli miglioramenti.

## A.4.2.2.4 MPR004 Frequenza di commit nella repository

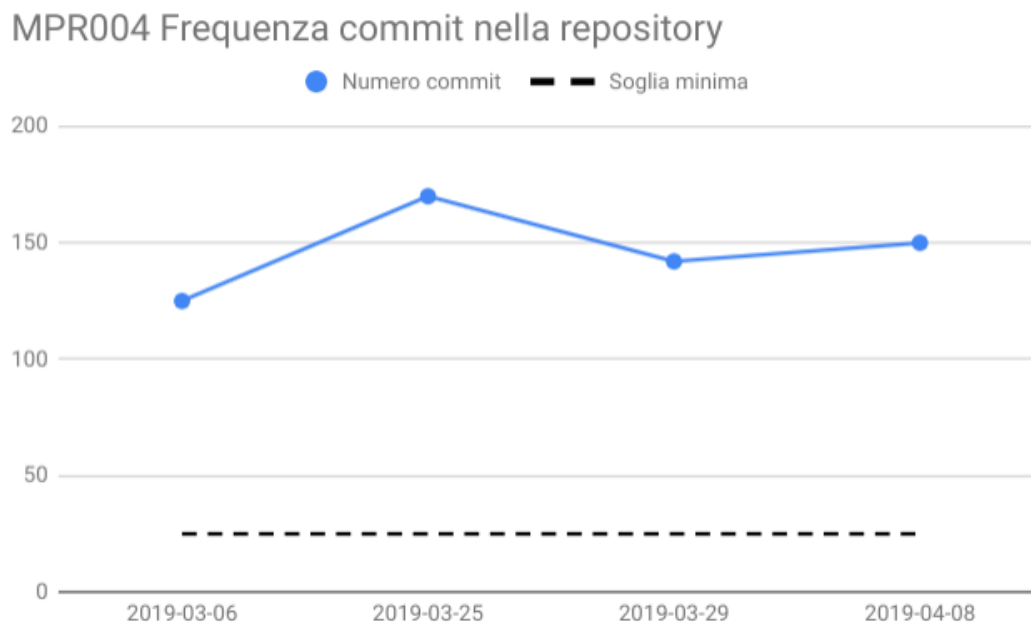


Figura 20: Diagramma con valori misurati tramite MPR004 Frequenza di commit nella repository

- **Obiettivo:** QPR004 Rispetto del numero minimo di commit nella repository.
- **Valore desiderato:** 25.
- **Descrizione:** è mostrato il numero di commit effettuati tra una verifica e la successiva.
- **Valutazione:** soddisfacente.
- **Considerazioni:** studiando il diagramma notiamo che il numero di commit è stato abbastanza lineare per tutto il periodo, sintomo di un lavoro costante e impegno.



#### A.4.2.2.5 MPR010 Moduli codificati prima della progettazione

##### MPR010 Moduli codificati prima della progettazione

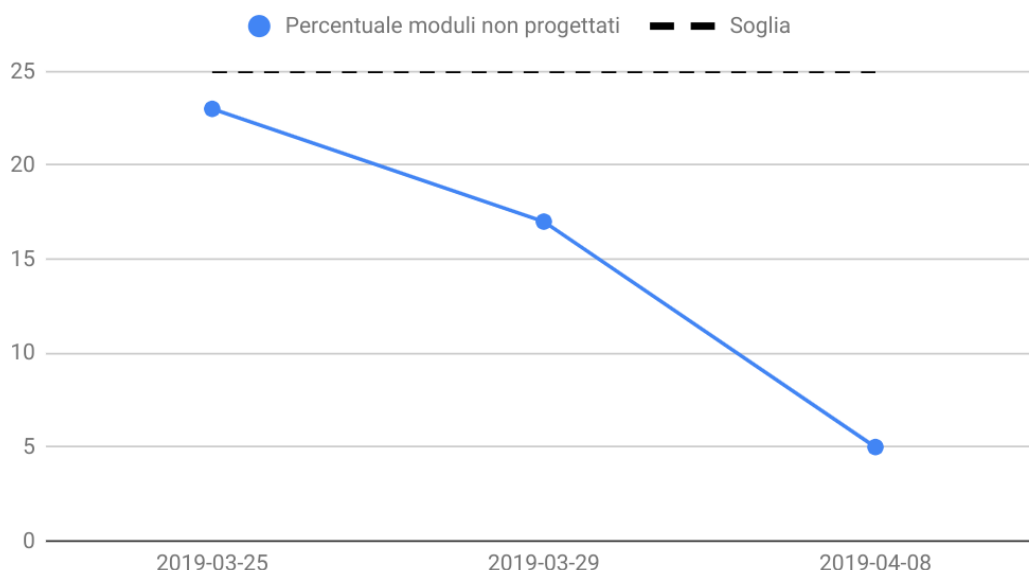


Figura 21: Diagramma con valori misurati tramite MPR010 Moduli codificati prima della progettazione

- **Obiettivo:** QPR011 Progettare prima di codificare.
- **Valore desiderato:** 25%.
- **Descrizione:** è mostrato il numero di moduli codificati prima dell'effettiva progettazione.
- **Valutazione:** soddisfacente.
- **Considerazioni:** studiando il diagramma notiamo che il numero di moduli codificati e non progettati è stato sotto la soglia per tutto il periodo di Revisione di Qualifica, sintomo di un lavoro accurato e appassionato.

#### A.4.2.2.6 MPR011 Numero di design pattern applicati

##### MPR011 Numero di design pattern applicati

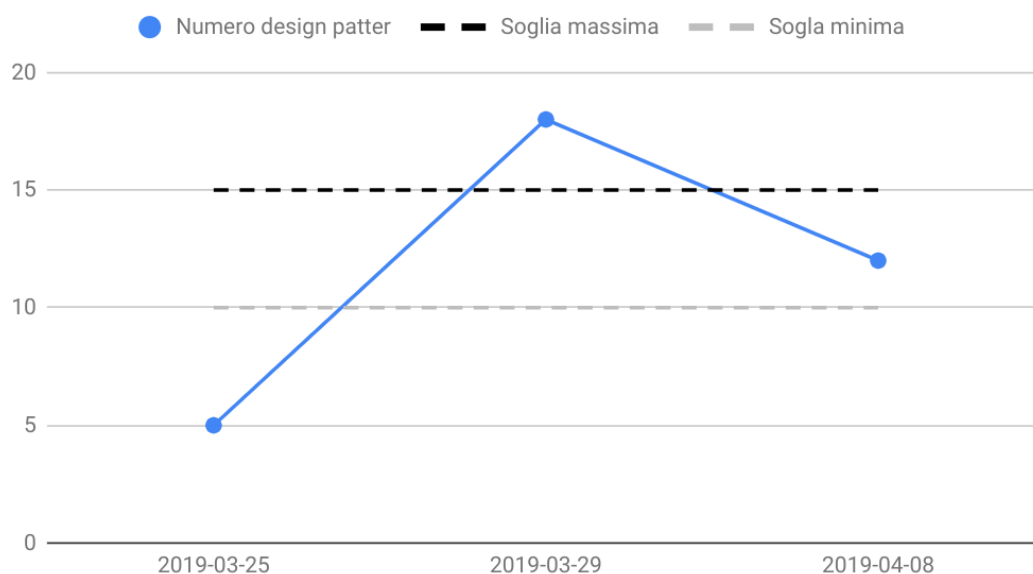


Figura 22: Diagramma con valori misurati tramite MPR011 Numero di design pattern applicati

- **Obiettivo:** QPR012 Usare un numero limitato di design pattern.
- **Valore desiderato:** 5-6.
- **Descrizione:** è mostrato il numero di design pattern applicati tra una verifica e la successiva.
- **Valutazione:** soddisfacente.
- **Considerazioni:** studiando il diagramma notiamo che nella verifica intermedia il numero di design pattern adottati supera il valore di soglia massimo. Il valore nella verifica successiva diminuisce e ritorna all'interno della soglia accettata, questo perché nel colloquio con Prof. Riccardo Cardinci è stato fatto notare che avendo esagerato con i design pattern aumentava la complessità architettuale del progetto.

#### A.4.2.2.7 MPR012 Moduli non testati

##### MPR012 Moduli non testati

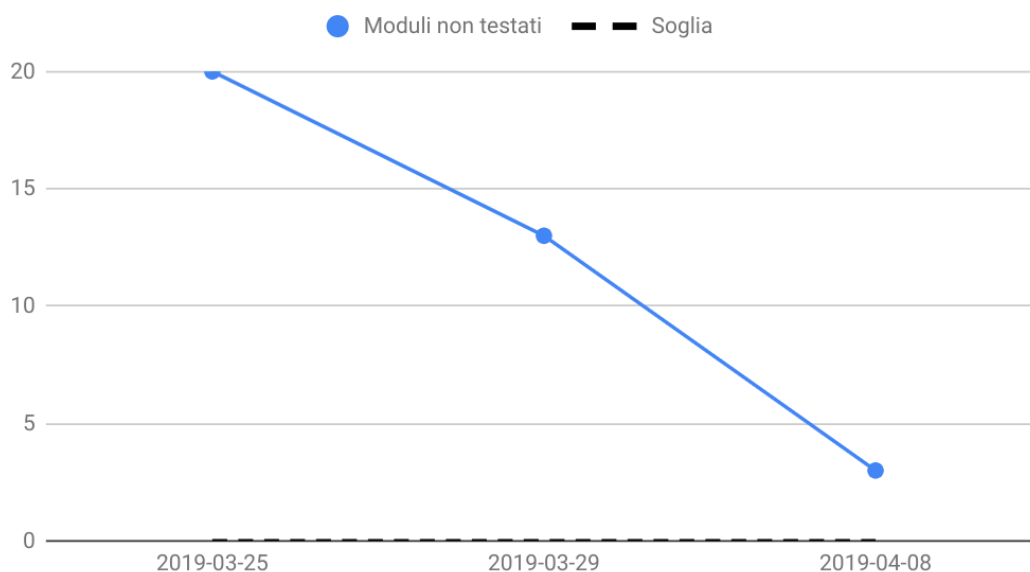


Figura 23: Diagramma con valori misurati tramite MPR012 Moduli non testati

- **Obiettivo:** QPR013 I moduli possiedono suite di test.
- **Valore desiderato:** 0.
- **Descrizione:** viene mostrato il numero di moduli non testati.
- **Valutazione:** non soddisfacente.
- **Considerazioni:** in questo periodo non siamo riusciti a raggiungere l'obiettivo che ci siamo prefissati, questo perché nell'ultima verifica avevamo ancora metodi non testati. Miriamo a raggiungere lo 0 nel prossimo periodo.

#### A.4.2.3 Software

Non tutte le metriche elencate in §3.3.2 sono riportate. Questo per non rendere il documento troppo verboso. I risultati non riportati appartengono alle metriche meno significative che vanno da MPS006 a MPS016. Tale scelta è stata fatta perché nella fase di codifica tutte le norme di codifica venivano rispettate durante la stesura del codice, perciò inserire diagrammi che riportano solo circa un 100% di rispetto della norma ci è sembrato superfluo.

## A.4.2.3.1 MPS001 Presenza di bug

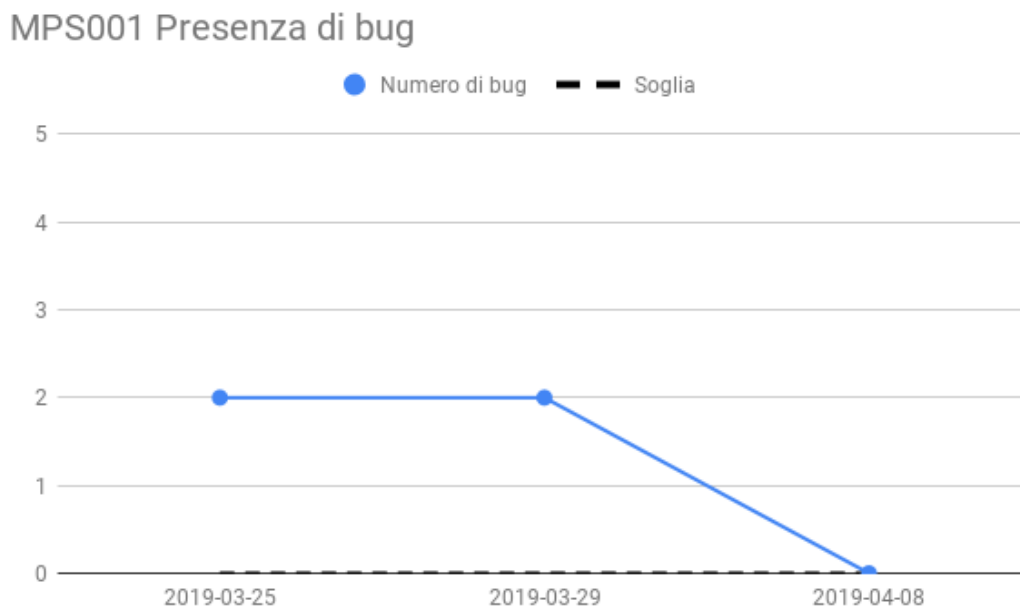


Figura 24: Diagramma con valori misurati tramite MPS001 Presenza di bug

- **Obiettivo:** QPS001 Assenza di bug.
- **Valore desiderato:** 0.
- **Descrizione:** viene mostrato il numero totale di bug rilevati da SonarQube.
- **Valutazione:** soddisfacente.
- **Considerazioni:** ci siamo impegnati a raggiungere il valore desiderato prima della Revisione di Qualifica.

#### A.4.2.3.2 MPS002 Presenza di vulnerabilità

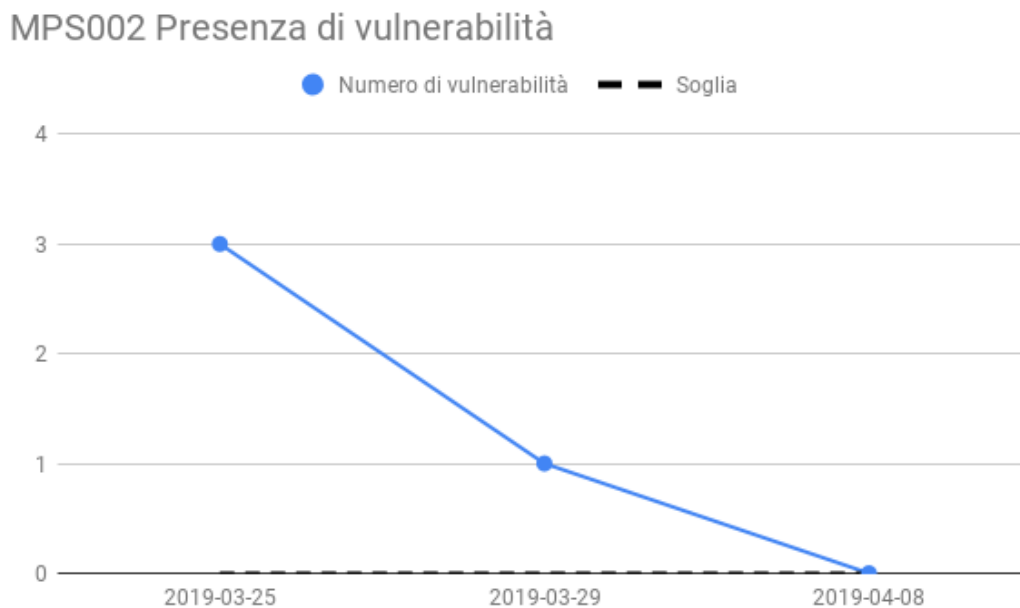


Figura 25: Diagramma con valori misurati tramite MPS002 Presenza di vulnerabilità

- **Obiettivo:** QPS002 Assenza di vulnerabilità.
- **Valore desiderato:** 0.
- **Descrizione:** viene mostrato il numero totale di vulnerabilità rilevate da SonarQube.
- **Valutazione:** soddisfacente.
- **Considerazioni:** ci siamo impegnati a raggiungere il valore desiderato prima della Revisione di Qualifica.

#### A.4.2.3.3 MPS003 Presenza di code smell

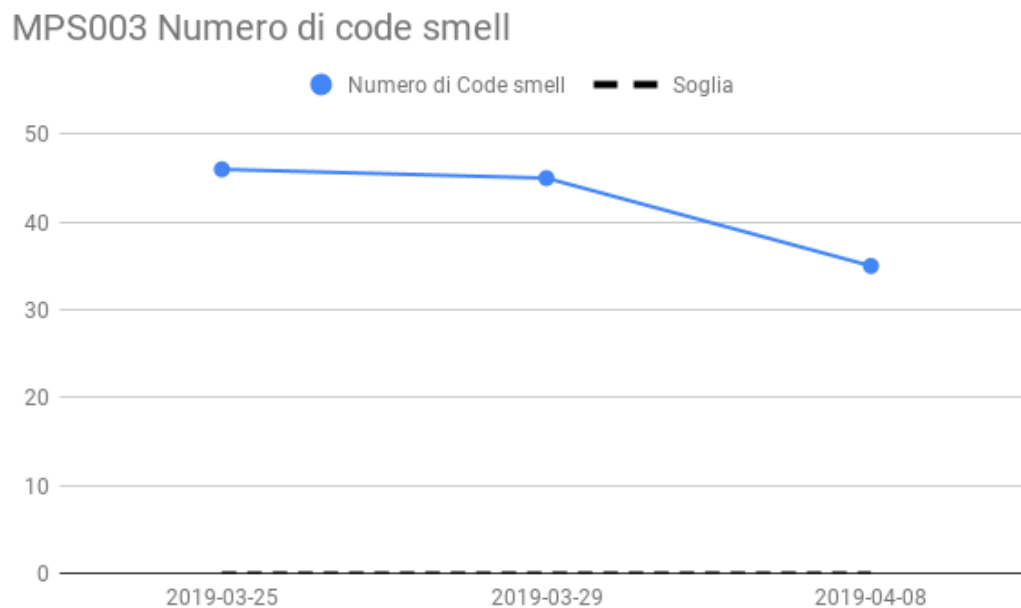


Figura 26: Diagramma con valori misurati tramite MPS003 Presenza di code smell

- **Obiettivo:** QPS003 Assenza di code smell.
- **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 Revisione di Qualifica. Prevediamo di raggiungere il valore desiderato alla prossima revisione.

## A.4.2.3.4 MPS004 Duplicazione del codice

## MPS004 Duplicazione del codice

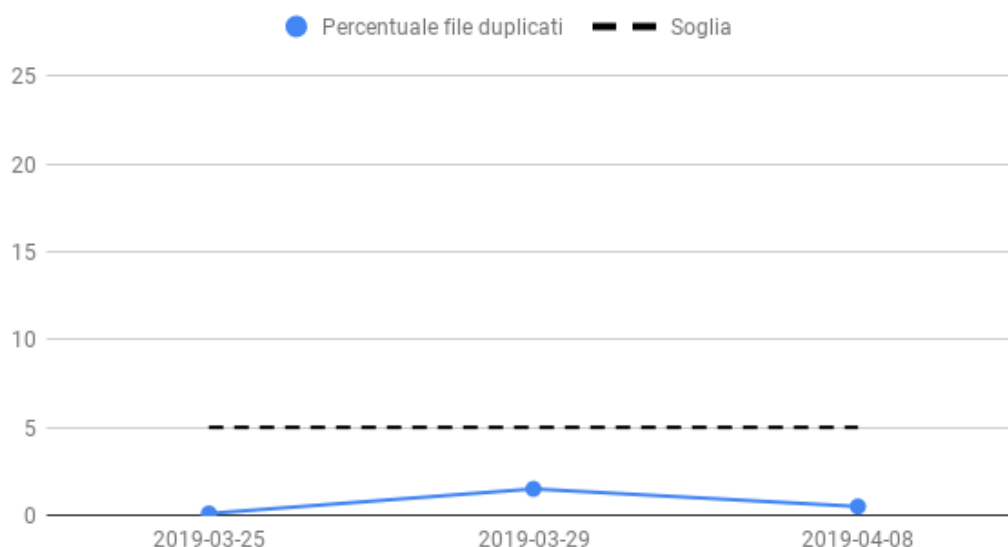


Figura 27: Diagramma con valori misurati tramite MPS004 Duplicazione del codice

- **Obiettivo:** QPS004 Minima duplicazione del codice.
- **Valore desiderato:** 0 - 5%.
- **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, per questo il valore soglia è stato abbassato da 20 al 5%. Miriamo ad abbassare la percentuale di duplicazione di codice ulteriormente nel prossimo periodo, ma difficilmente raggiungeremo uno 0%.

## A.4.2.3.5 MPS017 Code coverage

## MPS017 Code coverage

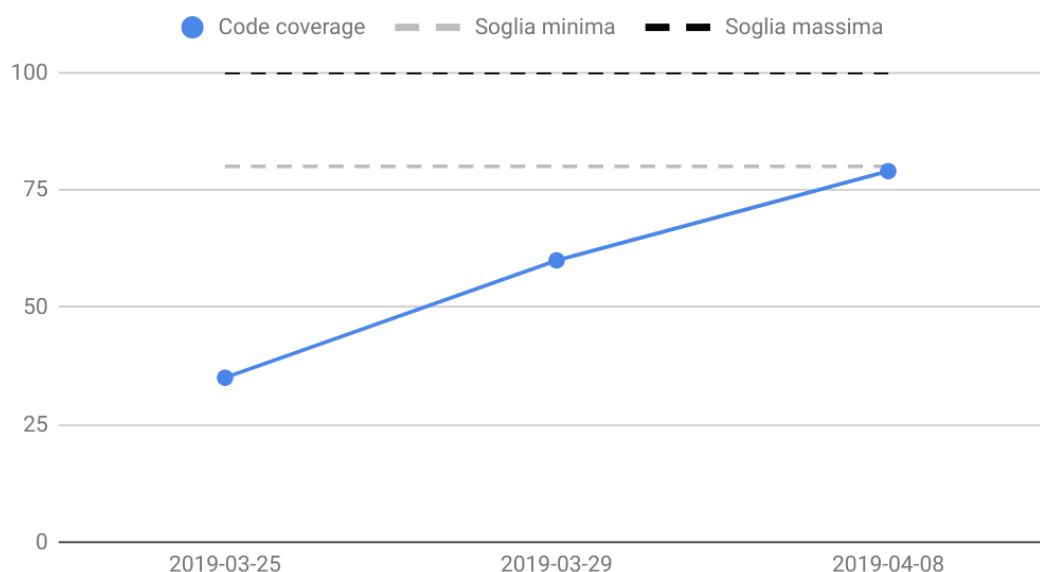


Figura 28: Diagramma con valori misurati tramite MPS017 Code coverage

- **Obiettivo:** QPS006 Massima code coverage.
- **Valore desiderato:** 80%-100%.
- **Descrizione:** viene mostrata la percentuale di codice testato rilevato da SonarQube.
- **Valutazione:** 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 Revisione di Qualifica, raggiungendo appena la soglia minima. Prevediamo di raggiungere un valore intorno al 90%.



#### A.4.2.3.6 MPS019 Test di integrazione

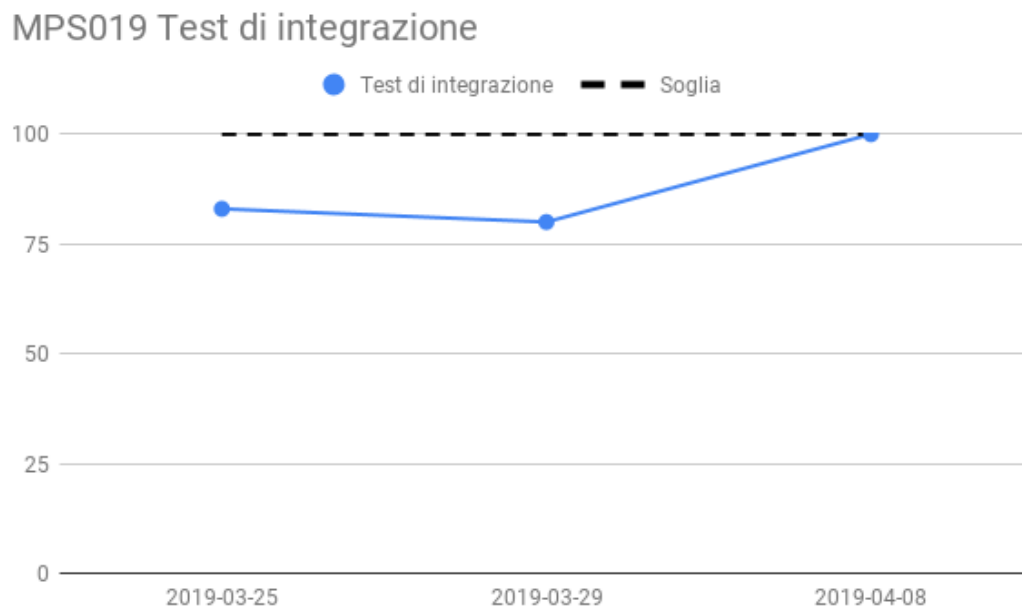


Figura 29: Diagramma con valori misurati tramite MPS019 Test di integrazione

- **Obiettivo:** QPS008 Superamento test di integrazione.
- **Valore desiderato:** 100%.
- **Descrizione:** viene mostrata la percentuale di codice coperto da test di integrazione.
- **Valutazione:** soddisfacente.
- **Considerazioni:** studiando il diagramma si nota che inizialmente ci siamo concentrati più sullo sviluppo individuale delle parti del software e successivamente sulle relazioni che esistono tra queste.

#### A.4.2.3.7 MPS020 Test di unità

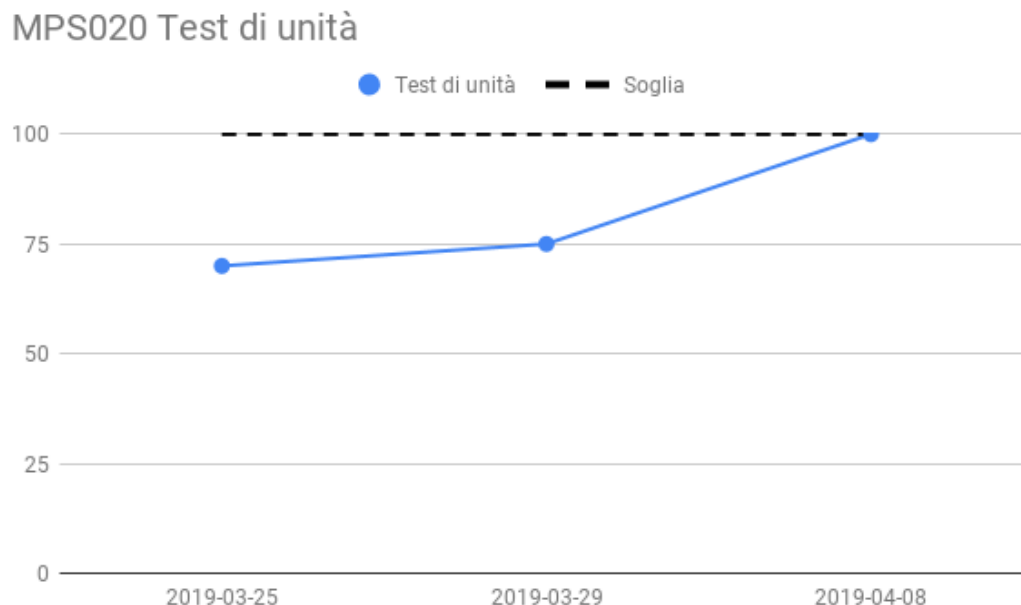


Figura 30: Diagramma con valori misurati tramite MPS020 Test di unità

- **Obiettivo:** QPS009 Superamento test di unità.
- **Valore desiderato:** 100%.
- **Descrizione:** viene mostrata la percentuale di codice coperto da test di unità.
- **Valutazione:** soddisfacente.
- **Considerazioni:** guardando le tre verifiche si nota che nel periodo antecedente alla Product Baseline il valore raccolto nella fase di verifica è molto insoddisfacente. Avvicinandosi alla PB e alla consegna della Revisione di Qualifica il valore è salito e siamo riusciti a raggiungere il valore desiderato.

### A.5 Quarto periodo (RA)

Periodo individuato come Revisione di accettazione, la quale dura all'incirca un mese dell'intera durata del progetto. L'obiettivo di questo periodo è quello di validare e collaudare il prodotto.

#### A.5.1 Riassunto delle attività di verifica

In questo periodo di tempo abbiamo attuato verifiche sui documenti per quanto riguarda i manuali e sul codice riguardante i test di sistema e di accettazione. Non avendo colloqui intermedi ci aspettiamo un andamento più lineare rispetto ai due periodi precedenti.

#### A.5.2 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 2019-04-12) 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 §A.1.
- **Considerazioni:** nostro commento in relazione ai risultati ottenuti dalla misurazione.

#### A.5.2.1 Documenti

##### A.5.2.1.1 MPD001 Indice di Gulpease

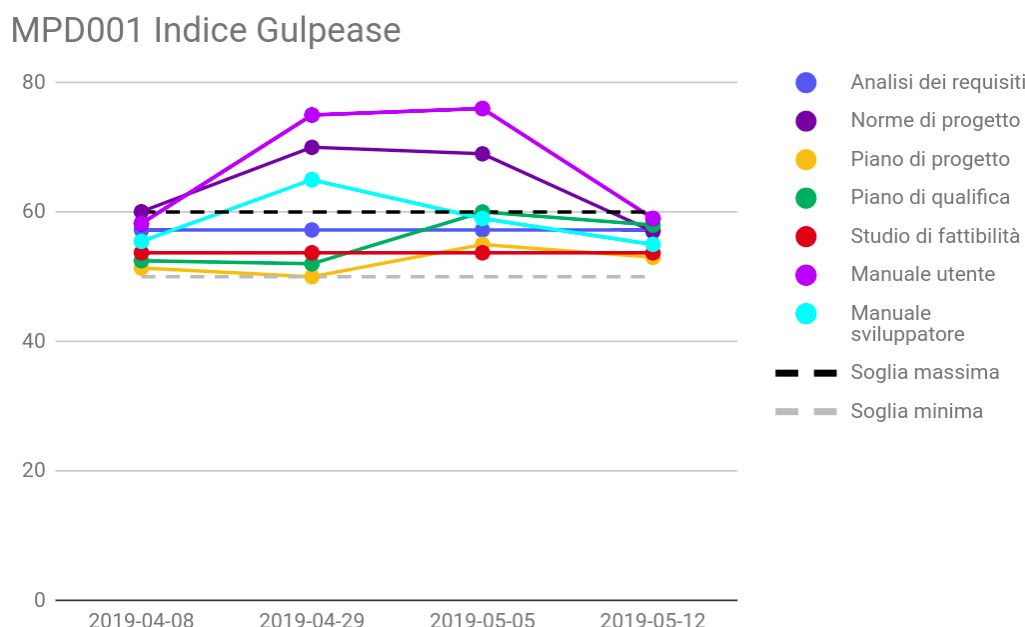


Figura 31: Diagramma con valori misurati tramite MPD001 Indice di Gulpease

- **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 rispettano i valori desiderati. Studiando il diagramma si nota che, a differenza del precedente periodo di Revisione di Qualifica, i valori ottenuti dalle verifiche non variano di molto durante la stesura. Questo è una conseguenza del fatto che i documenti non hanno subito radicali modifiche.

### A.5.2.1.2 MPD002 Correttezza ortografica

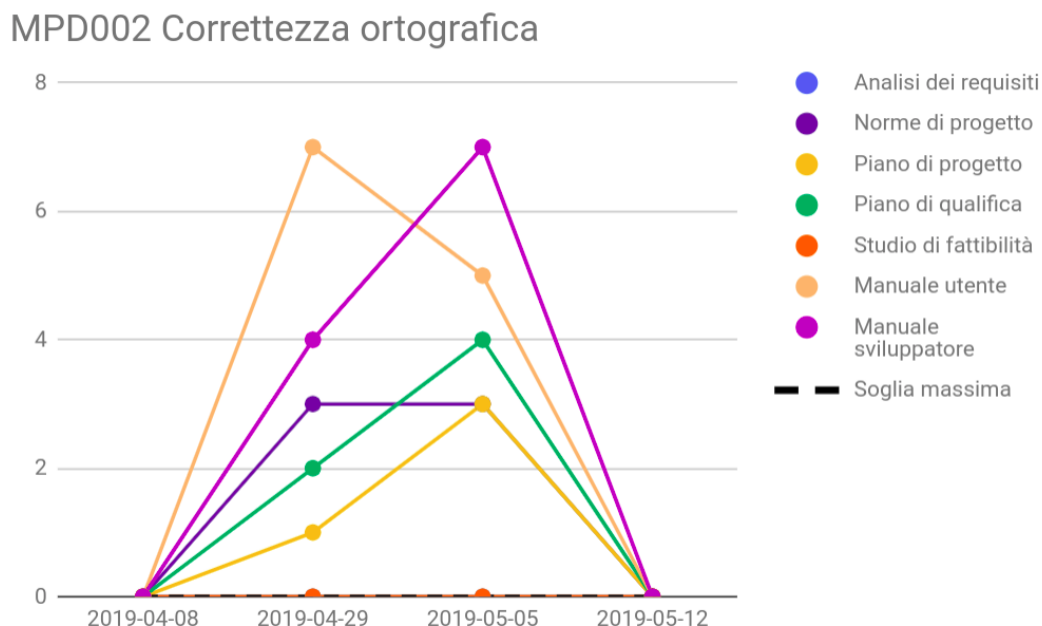


Figura 32: Diagramma con valori misurati tramite MPD002 Correttezza ortografica

- **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:** studiando il grafico si nota che i documenti con più errori sono i manuali, essendo documenti ancora poco maturi però proseguendo con le varie verifiche notiamo una costante diminuzione degli errori grammaticali.

## A.5.2.2 Processi

### A.5.2.2.1 MPR001 Varianza della pianificazione

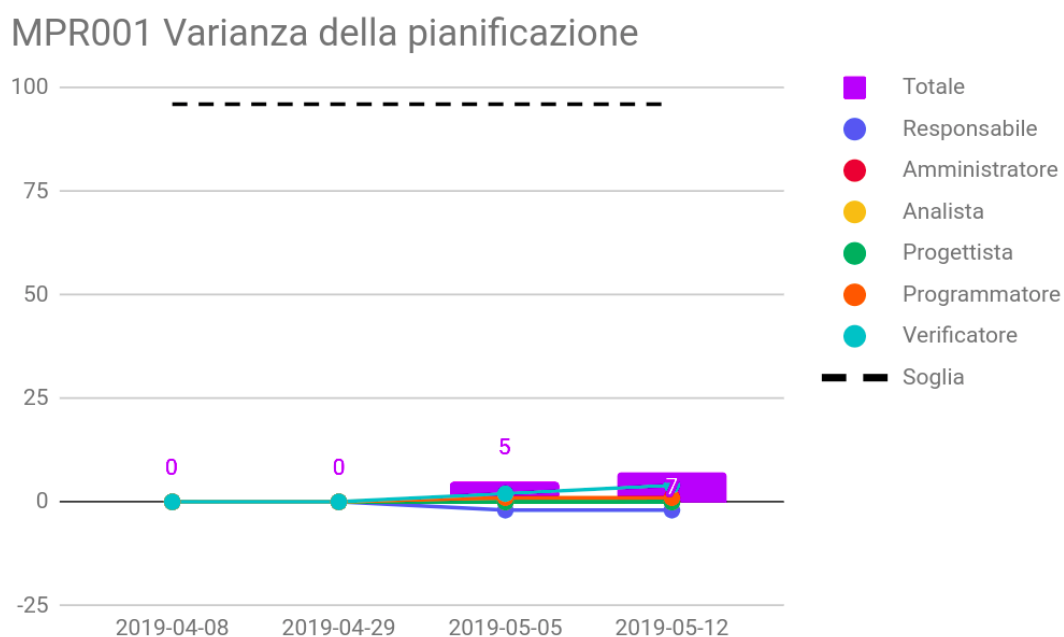


Figura 33: Diagramma con valori misurati tramite MPR001 Varianza della pianificazione

- **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:** soddisfacente.
- **Considerazioni:** in questo periodo abbiamo dovuto ridurre le ore al Responsabile per darne al Verificatore e al Programmatore, avendo bisogno di alcune ore per fare delle piccole correzioni al codice.

### A.5.2.2.2 MPR002 Varianza dei costi

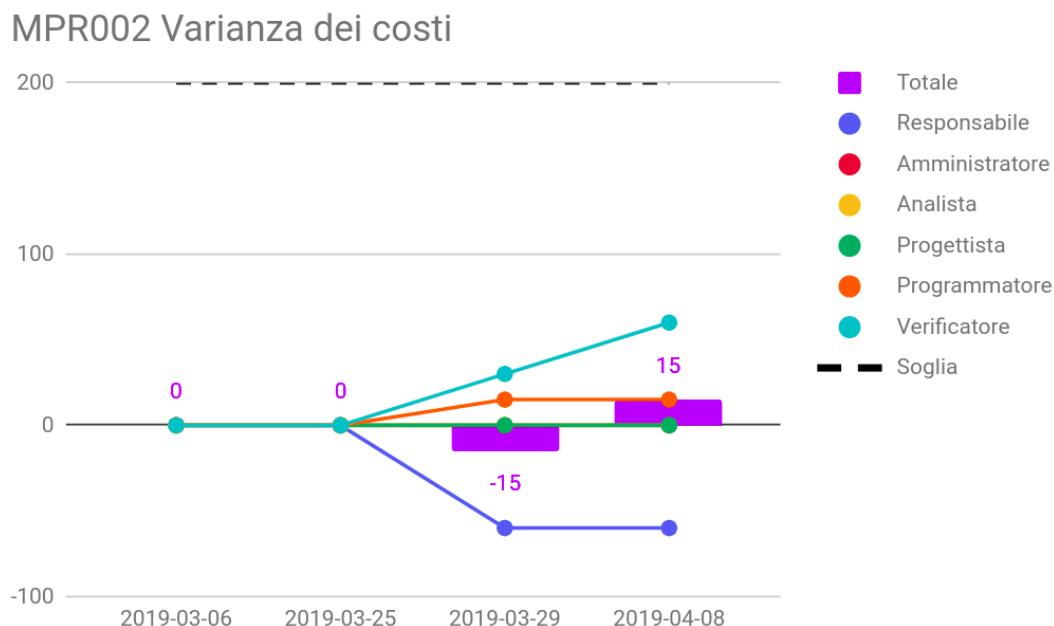


Figura 34: Diagramma con valori misurati tramite MPR002 Varianza dei costi

- **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 Responsabile a favore di quelle per il Verificatore e per il Programmatore non ha portato nessun cambiamento del preventivo. Un dato che risulta essere soddisfacente.

### A.5.2.2.3 MPR003 Aderenza agli standard

MPR003 Aderenza agli standard (del 2019-04-29)

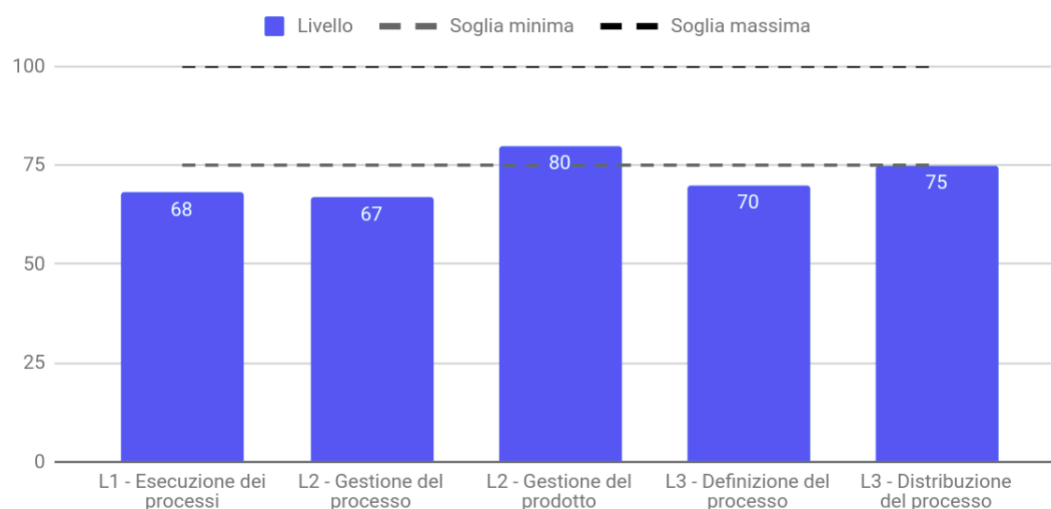


Figura 35: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (1)

MPR003 Aderenza agli standard (del 2019-05-05)

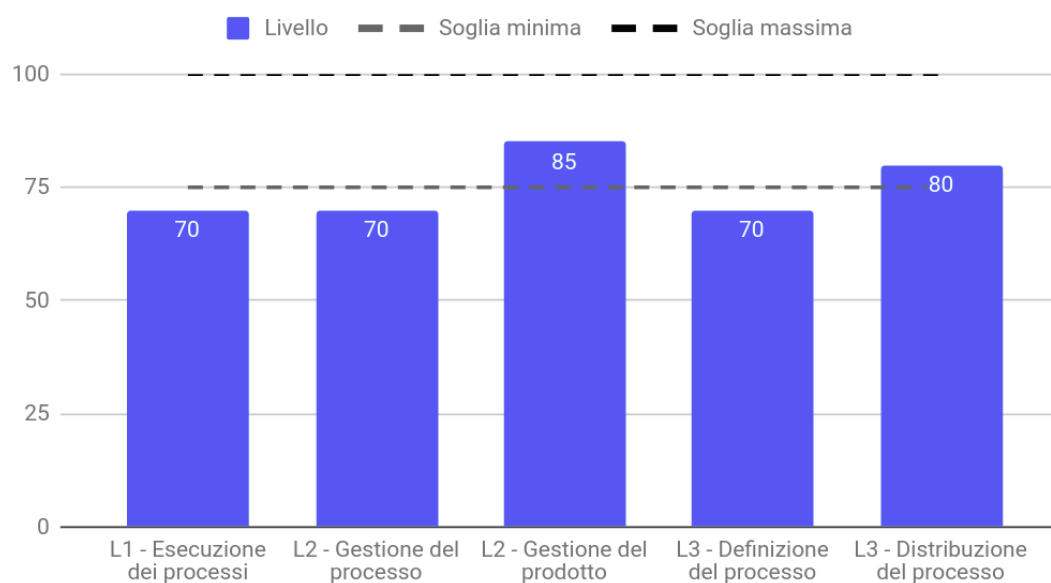


Figura 36: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (2)

## MPR003 Aderenza agli standard (del 2019-05-12)

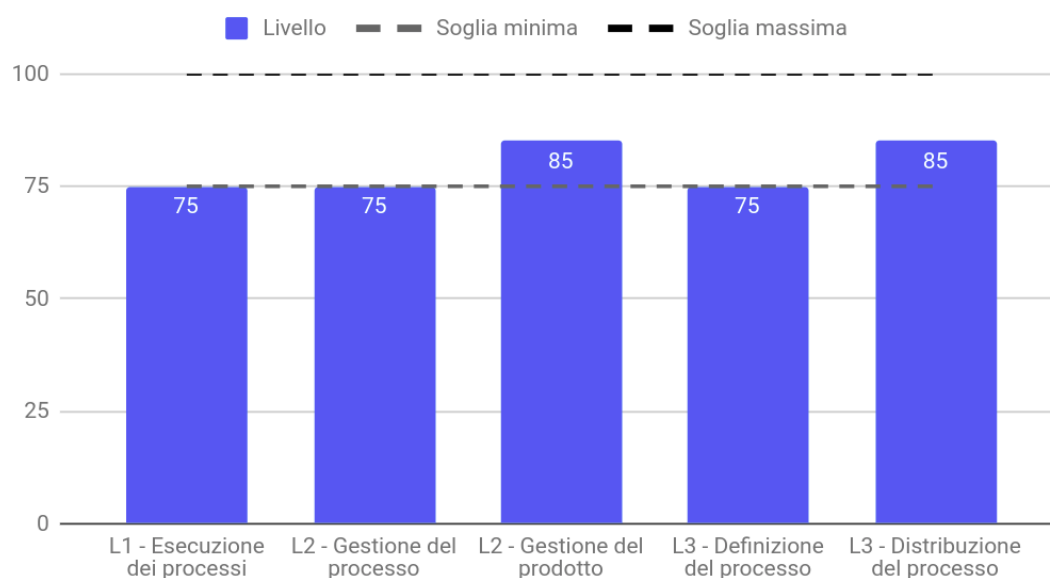


Figura 37: Diagramma con valori misurati tramite MPR003 Aderenza agli standard (3)

- **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:** soddisfacente.
- **Considerazioni:** il miglioramento per quanto riguarda l'aderenza agli standard, dopo un duro lavoro, è finalmente sufficiente per gli obiettivi prefissati.



## A.5.2.2.4 MPR004 Frequenza di commit nella repository

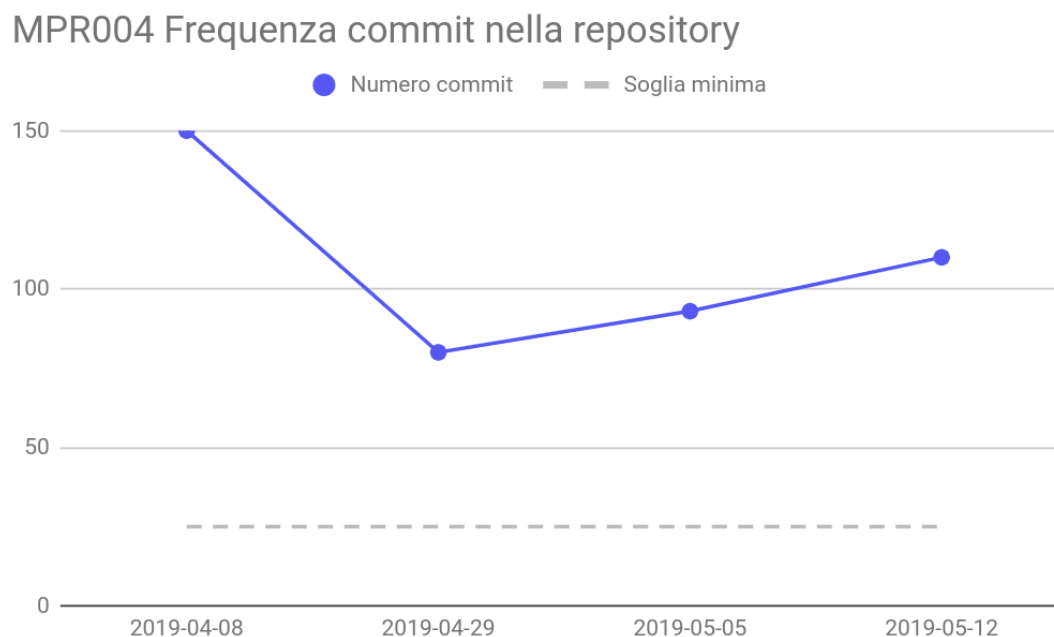


Figura 38: Diagramma con valori misurati tramite MPR004 Frequenza di commit nella repository

- **Obiettivo:** QPR004 Rispetto del numero minimo di commit nella repository.
- **Valore desiderato:** 25.
- **Descrizione:** è mostrato il numero di commit effettuati tra una verifica e la successiva.
- **Valutazione:** soddisfacente.
- **Considerazioni:** studiando il diagramma notiamo che il numero di commit nel primo periodo è calato, sintomo dei vari colloqui affrontati dai membri del gruppo. Dal secondo periodo il gruppo ha ripreso a lavorare con impegno e costanza.

#### A.5.2.2.5 MPR010 Moduli codificati prima della progettazione

##### MPR010 Moduli codificati prima della progettazione

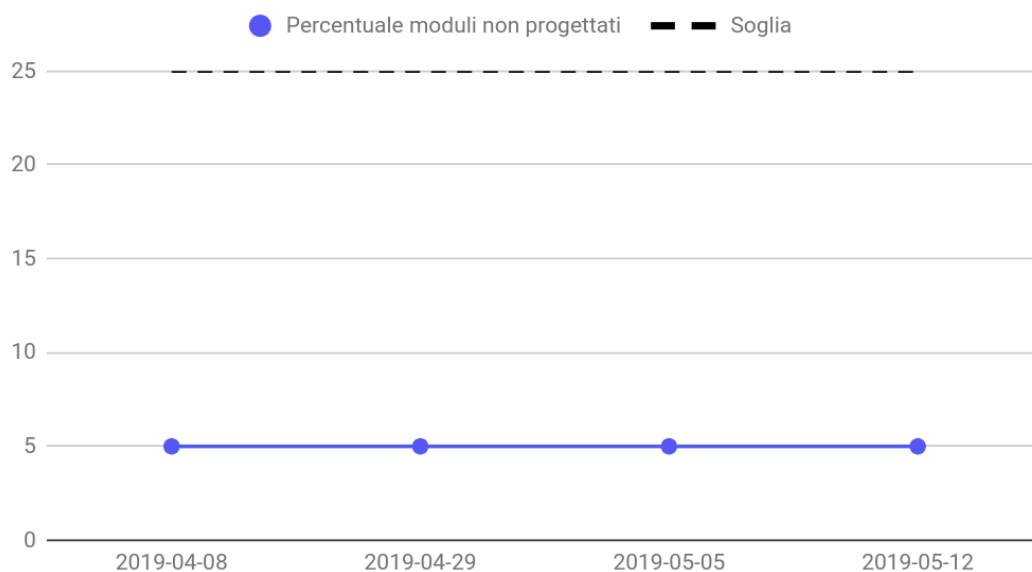


Figura 39: Diagramma con valori misurati tramite MPR010 Moduli codificati prima della progettazione

- **Obiettivo:** QPR011 Progettare prima di codificare.
- **Valore desiderato:** 25%.
- **Descrizione:** è mostrato il numero di moduli codificati prima dell'effettiva progettazione.
- **Valutazione:** soddisfacente.
- **Considerazioni:** i valori sono rimasti invariati perché la fase di progettazione è terminata lo scorso periodo. Inoltre non sono stati codificati ulteriori moduli non previsti dalla progettazione.

## A.5.2.2.6 MPR011 Numero di design pattern applicati

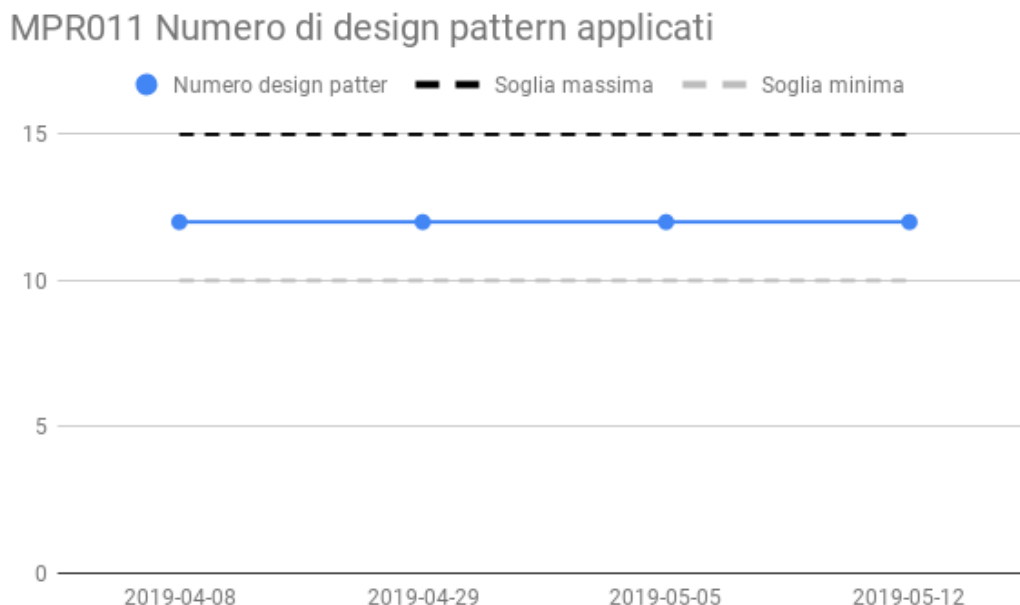


Figura 40: Diagramma con valori misurati tramite MPR011 Numero di design pattern applicati

- **Obiettivo:** QPR012 Usare un numero limitato di design pattern.
- **Valore desiderato:** 5-6.
- **Descrizione:** è mostrato il numero di design pattern applicati tra una verifica e la successiva.
- **Valutazione:** soddisfacente.
- **Considerazioni:** i valori sono rimasti invariati perché la fase di progettazione è terminata lo scorso periodo e non sono state implementate modifiche strutturali.

#### A.5.2.2.7 MPR012 Moduli non testati

MPR012 Moduli non testati

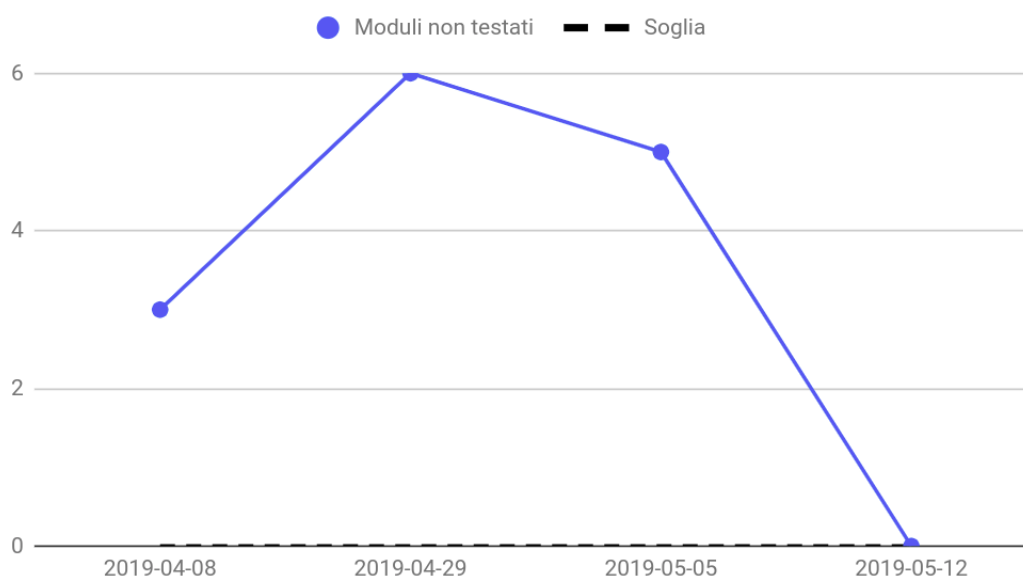


Figura 41: Diagramma con valori misurati tramite MPR012 Moduli non testati

- **Obiettivo:** QPR013 I moduli possiedono suite di test.
- **Valore desiderato:** 0.
- **Descrizione:** viene mostrato il numero di moduli non testati.
- **Valutazione:** soddisfacente.
- **Considerazioni:** come ci eravamo promessi nel periodo di Revisione di Progettazione siamo riusciti a testare tutti i moduli raggiungendo il valore desiderato.

#### A.5.2.2.8 MPR013 Percentuale metriche rispettate

##### MPR013 Percentuale metriche rispettate

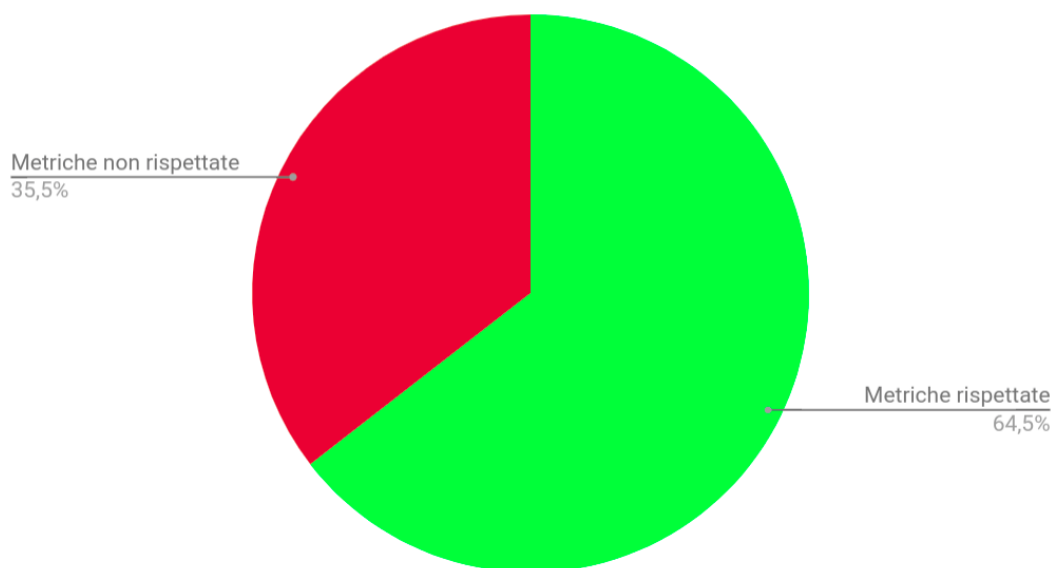


Figura 42: Diagramma con valori misurati tramite MPR013 Percentuale metriche rispettate

- **Obiettivo:** QPR016 Possedere un prodotto sviluppato correttamente.
- **Valore desiderato:** 60%.
- **Descrizione:** viene mostrata la percentuale delle metriche rispettate e non rispettate nel corso della Revisione di Progettazione, Revisione di Qualifica e Revisione di Accettazione nelle loro fasi di verifica.
- **Valutazione:** soddisfacente.
- **Considerazioni:** come ci aspettavamo il diagramma non ci restituisce un valore alto, questo è dovuto al fatto che analizza i risultati dei primi periodi quando non eravamo in grado di soddisfare sempre i valori desiderati. Questo diagramma ha lo scopo di fare un vero e proprio resoconto delle attività di verifica.

#### A.5.2.3 Software

Non tutte le metriche elencate in §3.3.2 sono riportate. Questo per non rendere il documento troppo verboso. I risultati non riportati appartengono alle metriche meno significative che vanno da MPS006 a MPS016. Tale scelta è stata fatta perché nella fase di codifica tutte le norme di codifica venivano rispettate durante la stesura del codice, perciò inserire diagrammi che riportano solo circa un 100% di rispetto della norma ci è sembrato superfluo.

#### A.5.2.3.1 MPS001 Presenza di bug



Figura 43: Diagramma con valori misurati tramite MPS001 Presenza di bug

- **Obiettivo:** QPS001 Assenza di bug.
- **Valore desiderato:** 0.
- **Descrizione:** viene mostrato il numero totale di bug rilevati da SonarQube.
- **Valutazione:** soddisfacente.
- **Considerazioni:** anche dopo la conclusione quasi definitiva del prodotto software, abbiamo riscontrato alcuni bug dovuti anche alle ultime funzionalità marginali che abbiamo aggiunto.

#### A.5.2.3.2 MPS002 Presenza di vulnerabilità



Figura 44: Diagramma con valori misurati tramite MPS002 Presenza di vulnerabilità

- **Obiettivo:** QPS002 Assenza di vulnerabilità.
- **Valore desiderato:** 0.
- **Descrizione:** viene mostrato il numero totale di vulnerabilità rilevate da SonarQube.
- **Valutazione:** soddisfacente.
- **Considerazioni:** durante le fasi di test si presentavano temporanee vulnerabilità nel codice.

### A.5.2.3.3 MPS003 Presenza di code smell

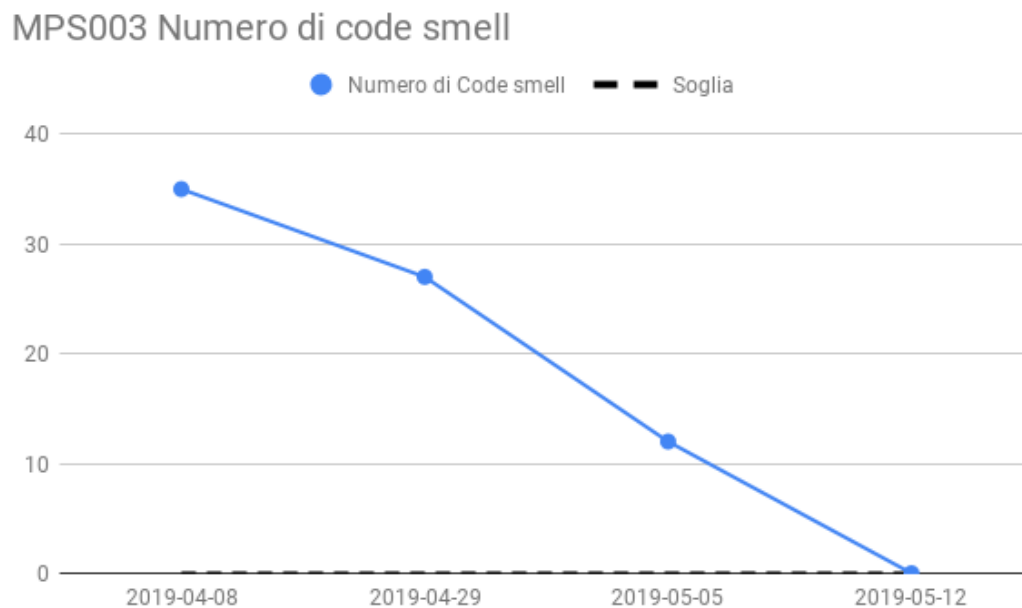


Figura 45: Diagramma con valori misurati tramite MPS003 Presenza di code smell

- **Obiettivo:** QPS003 Assenza di code smell.
- **Valore desiderato:** 0.
- **Descrizione:** viene mostrato il numero totale di code smell rilevati da SonarQube.
- **Valutazione:** poco soddisfacente.
- **Considerazioni:** come indicato alla revisione precedente, ci siamo concentrati sul sistemare le ultime sregolatezze del codice, tra cui anche i code smell.



#### A.5.2.3.4 MPS004 Duplicazione del codice

##### MPS004 Duplicazione del codice

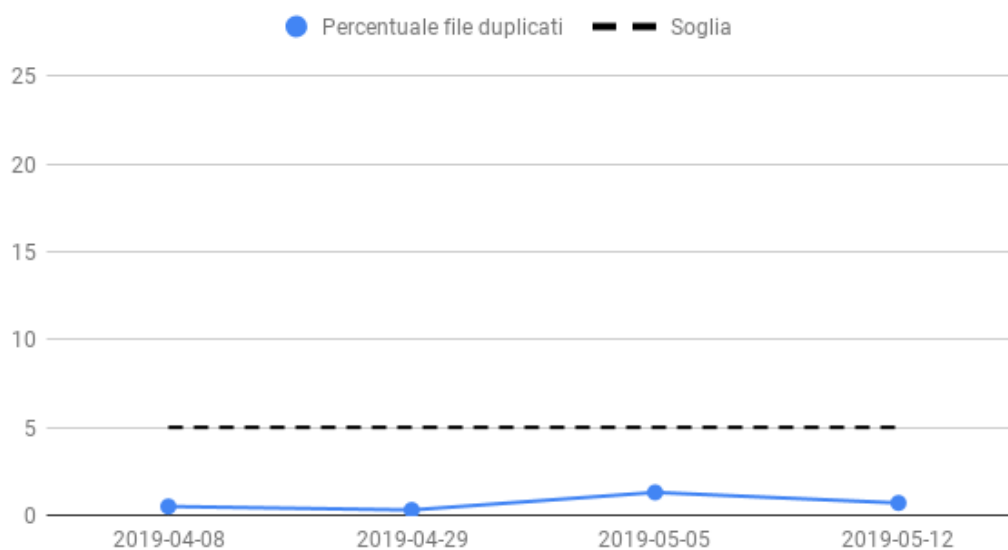


Figura 46: Diagramma con valori misurati tramite MPS004 Duplicazione del codice

- **Obiettivo:** QPS004 Minima duplicazione del codice.
- **Valore desiderato:** 0 - 5%.
- **Descrizione:** viene mostrata la percentuale di codice duplicato rilevato da SonarQube.
- **Valutazione:** soddisfacente.
- **Considerazioni:** alla termine del progetto non siamo riusciti ad azzerare la duplicazione di codice, anche se non era uno dei nostri obiettivi principi per la consegna dei prodotti.

## A.5.2.3.5 MPS017 Code coverage

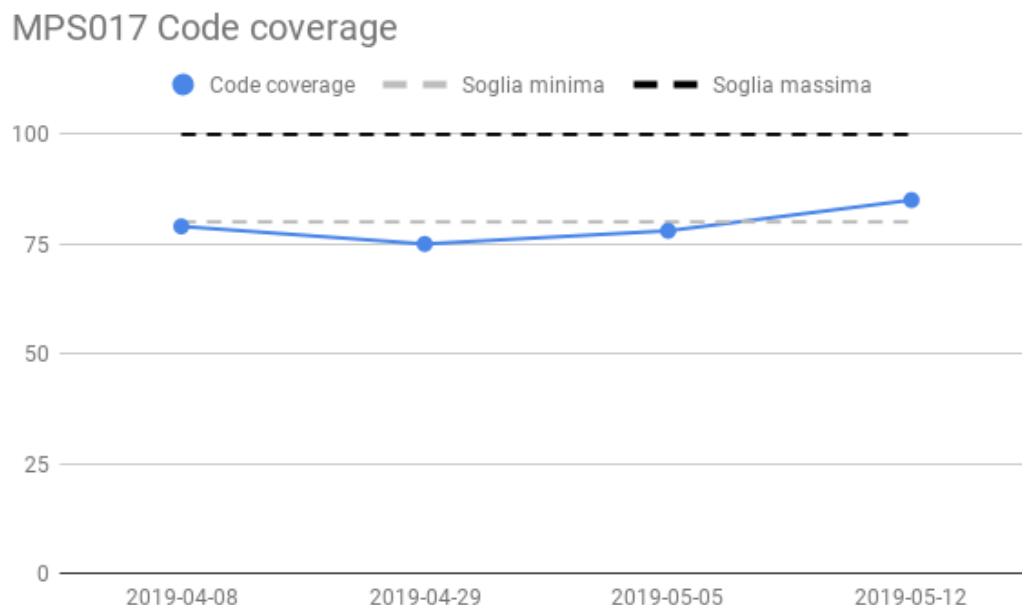


Figura 47: Diagramma con valori misurati tramite MPS017 Code coverage

- **Obiettivo:** QPS006 Massima code coverage.
- **Valore desiderato:** 80%-100%.
- **Descrizione:** viene mostrata la percentuale di codice testato rilevato da SonarQube.
- **Valutazione:** soddisfacente.
- **Considerazioni:** all'ultima verifica il risultato ha raggiunto la soglia minima, purtroppo senza toccare la soglia massima. In particolare abbiamo avuto difficoltà a verificare la parte del progetto scritta in `JAVASCRIPT_G`.

## A.5.2.3.6 MPS018 Test di sistema

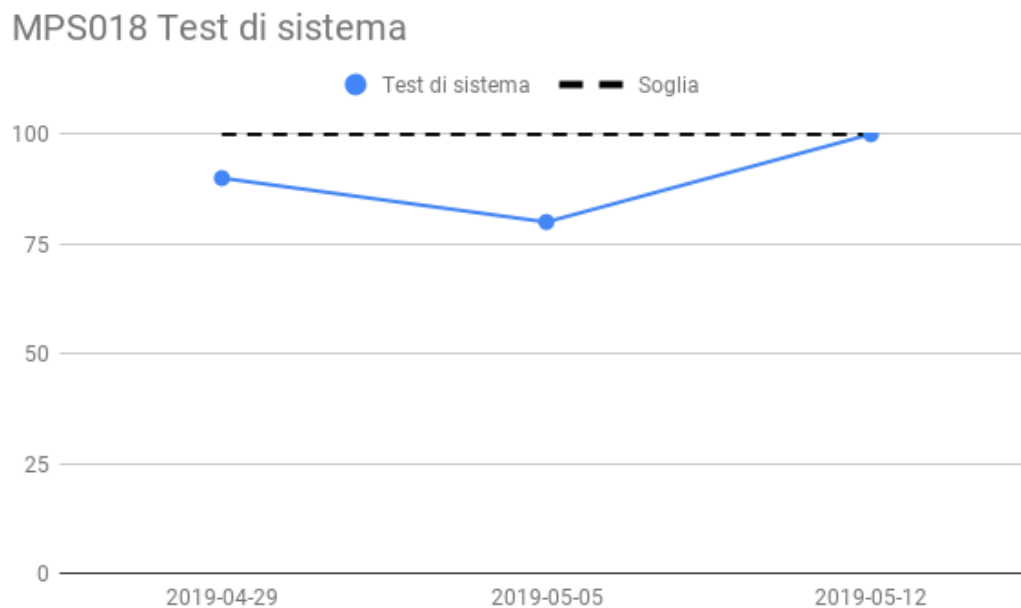


Figura 48: Diagramma con valori misurati tramite MPS018 Test di sistema

- **Obiettivo:** QPS007 Superamento test di sistema.
- **Valore desiderato:** 100%.
- **Descrizione:** viene mostrata la percentuale di codice coperto da test di sistema.
- **Valutazione:** soddisfacente.
- **Considerazioni:** fin da subito non tutti i test di sistema non sono stati superati, dato che Davide Zanetti di Imola Informatica ci ha chiesto, durante un colloquio, di modificare alcune pagine del Gestore Personale.

## A.5.2.3.7 MPS019 Test di integrazione

## MPS019 Test di integrazione

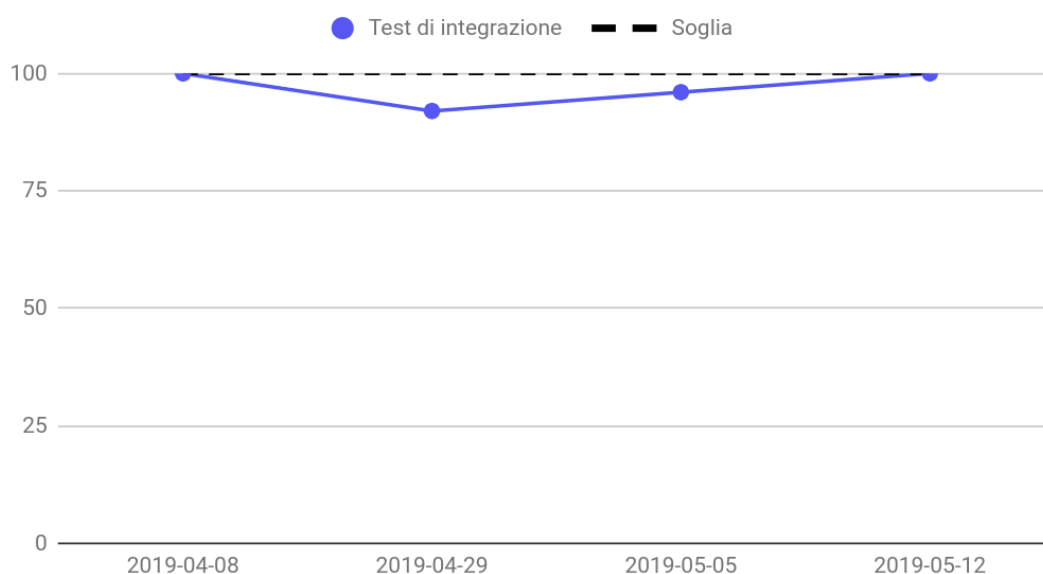


Figura 49: Diagramma con valori misurati tramite MPS019 Test di integrazione

- **Obiettivo:** QPS008 Superamento test di integrazione.
- **Valore desiderato:** 100%.
- **Descrizione:** viene mostrata la percentuale di codice coperto da test di integrazione.
- **Valutazione:** soddisfacente.
- **Considerazioni:** provando ad aggiungere nuove funzionalità si sono aggiunti ulteriori test di sistema che non sono stati superati fin da subito.

## A.5.2.3.8 MPS020 Test di unità

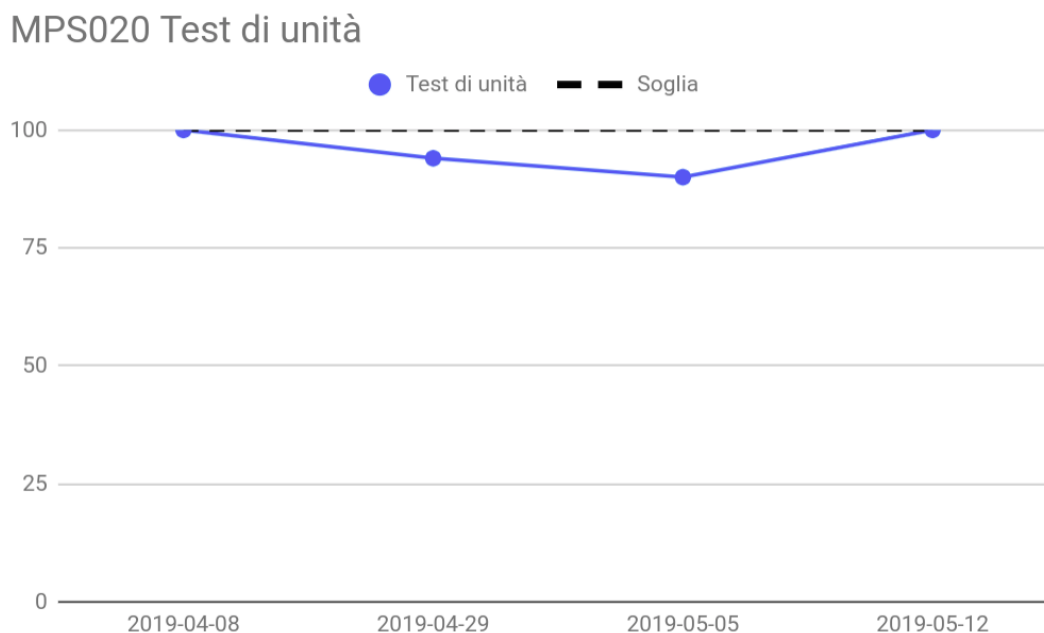


Figura 50: Diagramma con valori misurati tramite MPS020 Test di unità

- **Obiettivo:** QPS009 Superamento test di unità.
- **Valore desiderato:** 100%.
- **Descrizione:** viene mostrata la percentuale di codice coperto da test di unità.
- **Valutazione:** soddisfacente.
- **Considerazioni:** come detto per i test di integrazione, con l'introduzione delle ultime funzionalità sono stati introdotti nuovi test che non venivano superati subito.

## B 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 §A.

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

### B.1 Primo periodo (RR)

#### B.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 v4.0.0<sub>D</sub>*.
- **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<sub>G</sub> risolvevano la stessa issue.
- **Soluzione proposta:** l'Amministratore si impegna a creare issue più precise e circoscritte, evitando di dimenticarsi di inserire un assegnatario.

#### B.1.2 Valutazione dei ruoli

##### B.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<sub>G</sub> di GitHub per assegnare i compiti in maniera incrementale e omogenea.

##### B.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 *NormeDiProgetto v4.0.0<sub>D</sub>* 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.

##### B.1.2.3 Analista

- **Problema riscontrato:** la principale difficoltà è stata la stesura dell'*AnalisiDeiRequisiti v3.0.0<sub>D</sub>* 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<sub>G</sub> 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.

#### B.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.

#### B.1.3 Valutazione sugli strumenti

##### B.1.3.1 L<sup>A</sup>T<sub>E</sub>X

- **Problema riscontrato:** la necessità iniziale di avere dei TEMPLATE<sub>G</sub> su cui poter lavorare è stato uno dei problemi iniziali che ha necessitato grande attenzione in quanto non tutti i membri sapevano usare L<sup>A</sup>T<sub>E</sub>X 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.

##### B.1.3.2 Git

- **Problema riscontrato:** una difficoltà riscontrata, anche se raramente, è stata quella dei conflitti durante i commit sulla REPOSITORY<sub>G</sub> 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<sub>G</sub> da un'altro di noi.

#### B.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 *NormeDiProgetto v4.0.0<sub>D</sub>*, 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.

#### B.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<sup>A</sup>T<sub>E</sub>X:** 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<sub>G</sub>, 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<sub>G</sub>.

## B.2 Secondo periodo (RP)

### B.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.

### B.2.2 Valutazione dei ruoli

#### B.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.



#### B.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.

#### B.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.

#### B.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 stringente, il numero o le date per le fasi di verifica obbligatorie che non possono essere prorogate.

### B.2.3 Valutazione sugli strumenti

#### B.2.3.1 Docker

- **problema riscontrato:** conoscere una nuova tecnologia richiede tempo, e nel nostro caso con DOCKER<sub>G</sub> 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.

### B.3 Terzo periodo (RQ)

#### B.3.1 Valutazioni sull'organizzazione

- **Problema riscontrato:** la scarsa conoscenza dei design pattern ha rallentato inizialmente il lavoro pianificato.
- **Soluzione proposta:** maggior studio individuale dei design pattern, soprattutto sul valore aggiunto con la loro applicazione.

#### B.3.2 Valutazione dei ruoli

##### B.3.2.1 Analista

- **Problema riscontrato:** è stato necessario modificare alcuni requisiti e aggiungere alcuni diagrammi dei casi d'uso, attività non previste dalla pianificazione.

- **Soluzione proposta:** abbiamo aggiunto delle ore necessarie alla correzione dei requisiti e dei casi d'uso.

### B.3.2.2 Progettista

- **Problema riscontrato:** l'eccessivo zelo del gruppo nell'applicare i design pattern al progetto ha portato a una complessità eccessiva nell'architettura del progetto.
- **Soluzione proposta:** a seguito del riscontro della PB, abbiamo ritenuto necessario rivedere la progettazione di alcune componenti del sistema in modo da semplificare l'architettura.

### B.3.2.3 Programmatore

- **Problema riscontrato:** è stato pianificato un monte ore troppo elevato per la codifica delle varie componenti del sistema.
- **Soluzione proposta:** riduzione delle ore consuntivate.

### B.3.2.4 Verificatore

- **Problema riscontrato:** a causa di impegni di alcuni membri del gruppo, le attività di verifica della progettazione sono state posticipate, portando a un ritardo nello svolgimento di altre attività.
- **Soluzione proposta:** rispettare le scadenze date dalla pianificazione.

## B.3.3 Valutazione sugli strumenti

### B.3.3.1 Rancher

- **Problema riscontrato:** come per Docker, si tratta di una nuova tecnologia e richiede tempo per imparare a usarla.
- **Soluzione proposta:** concentrarsi sugli aspetti inerenti al progetto, in modo da ridurre i tempi richiesti per l'uso dei vari componenti.

## B.4 Quarto periodo (RA)

### B.4.1 Valutazioni conclusive sull'intero progetto

Osservando a ritroso tutto il nostro percorso effettuato all'interno di questo progetto, è visibile l'incrementale acquisizione di competenze, conoscenze e abilità; i tre metri di misura principali per riconoscere le qualifiche ottenute da una persona durante un percorso formativo, come questo progetto.

Grazie ad un'auto analisi dei ruoli svolti all'interno del progetto è stato possibile migliorare nel tempo le loro funzioni. All'inizio del progetto, era stato fornito cosa ogni ruolo dovesse svolgere, ma non era subito chiaro come invece dovesse essere svolto. Tali lacune sono state poi riempite grazie alla pratica.

Inserire metriche sempre più granulari è stato utile per poter analizzare aspetti del progetto all'apparenza futili, ma che si sono rivelate importanti per capire dove spendere più risorse nelle fasi di verifica.

In conclusione è stato utile tracciare i risultati delle varie metriche per avere una visione generale, ma anche specifica, dell'andamento del progetto. Ancora di più utile è stata l'autovalutazione dei vari membri del gruppo attraverso suggerimenti reciproci i quali ci hanno aiutato a restare un gruppo unito e con lo stesso livello di conoscenze, evitando disparità.